UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

DOCTORADO EN CIENCIAS APLICADAS Y TECNOLOGÍA

TESIS

Agile Data Science - Analytics Methodology (AgileDSAM) –a Scrum-aligned Development Methodology for Big Data Software Systems in Small Business

PRESENTA

MITC. Gerardo Salazar Salazar

TUTOR

Dr. José Manuel Mora Tavarez

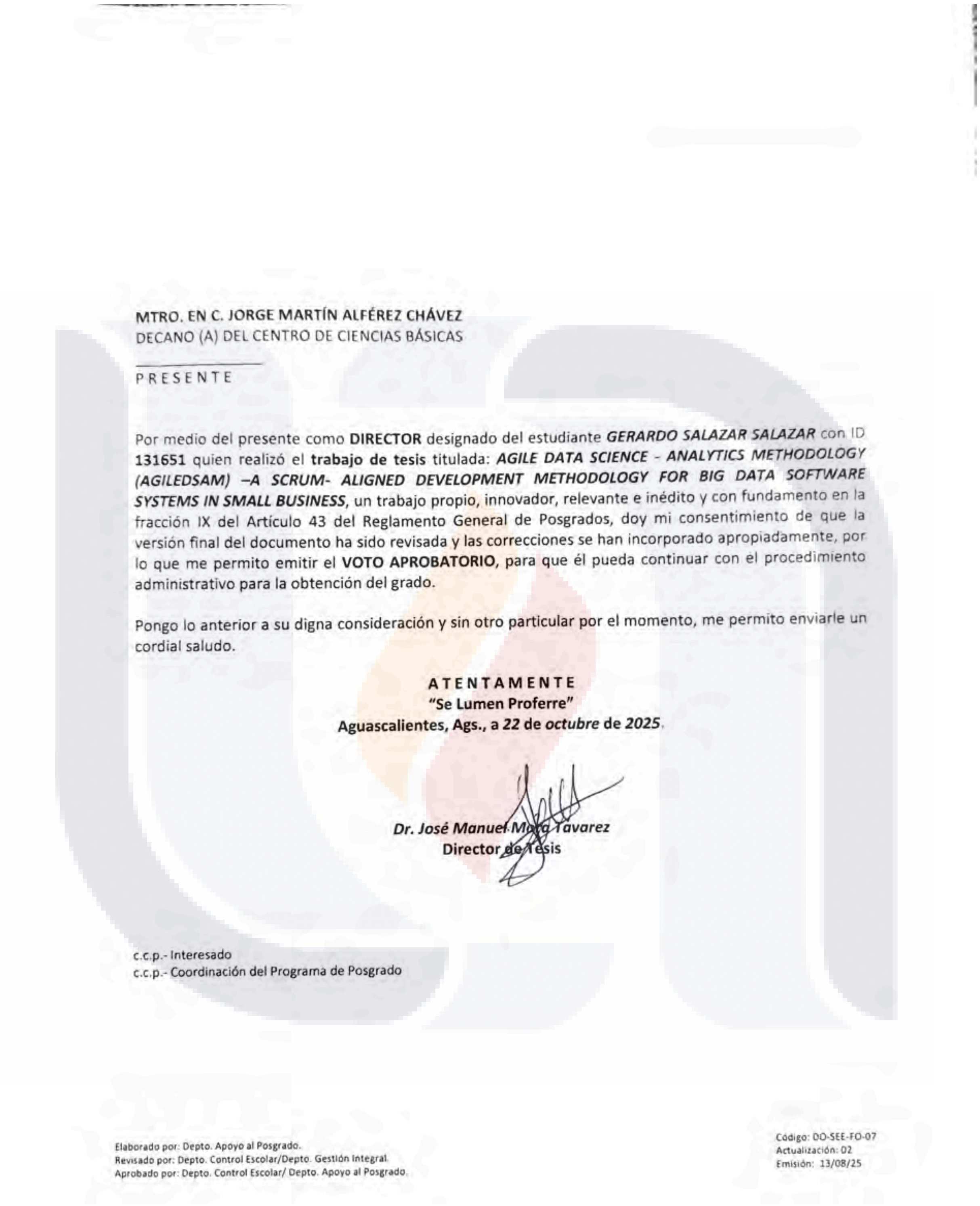
CO-TUTOR

Dr. Héctor Alejandro Durán Limón

COMITÉ TUTORAL

Dr. Francisco Javier Álvarez Rodríguez

Cd. Universitaria, Aguascalientes, Ags. Noviembre 2025



MTRO. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como **DIRECTOR** designado del estudiante ***GERARDO SALAZAR SALAZAR*** con ID **131651** quien realizó el **trabajo de tesis** titulada: ***AGILE DATA SCIENCE - ANALYTICS METHODOLOGY (AGILEDSAM) –A SCRUM- ALIGNED DEVELOPMENT METHODOLOGY FOR BIG DATA SOFTWARE SYSTEMS IN SMALL BUSINESS***, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a *22* de *octubre* de *2025*.

Dr. José Manuel Mora Tavarez
Director de Tesis

c.c.p.- Interesado
c.c.p.- Coordinación del Programa de Posgrado

CARTA DE VOTO APROBATORIO

MTRO. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
____
PRESENTE

Por medio del presente como **CODIRECTOR** designado del estudiante ***GERARDO SALAZAR SALAZAR*** con ID **131651** quien realizó el **trabajo de tesis** titulada: ***AGILE DATA SCIENCE - ANALYTICS METHODOLOGY (AGILEDSAM) –A SCRUM- ALIGNED DEVELOPMENT METHODOLOGY FOR BIG DATA SOFTWARE SYSTEMS IN SMALL BUSINESS***, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a *22* de *octubre* de *2025*.

Dr. Héctor Alejandro Durán Limón
Codirector de Tesis

c.c.p.- Interesado
c.c.p.- Coordinación del Programa de Posgrado

[illegible]

MTRO. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como **ASESOR** designado del estudiante ***GERARDO SALAZAR SALAZAR*** con ID **131651** quien realizó el **trabajo de tesis** titulada: ***AGILE DATA SCIENCE - ANALYTICS METHODOLOGY (AGILEDSAM) –A SCRUM- ALIGNED DEVELOPMENT METHODOLOGY FOR BIG DATA SOFTWARE SYSTEMS IN SMALL BUSINESS***, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a *22* de *octubre* de *2025*.

Dr. Francisco Javier Álvarez Rodríguez
Asesor de Tesis

c.c.p.- Interesado
c.c.p.- Coordinación del Programa de Posgrado

universidad autónoma de aguascalientes

DICTAMEN DE LIBERACIÓN ACADÉMICA PARA INICIAR LOS TRÁMITES DEL EXAMEN DE GRADO

POSGRADOS uaa

Fecha de dictaminación (dd/mm/aaaa): 29/10/2025

NOMBRE: Gerardo Salazar Salazar **ID** 131651

PROGRAMA: Doctorado en Ciencias Aplicadas y Tecnología **LGAC (del posgrado):** Tecnologías de Ingeniería de Software y Objetos de Aprendizaje

MODALIDAD DEL PROYECTO DE GRADO: Tesis Tradicional (x) *Tesis por artículos científicos ( ) **Tesis por Patente ( ) Trabajo Práctico ( )

TITULO: Agile Data Science - Analytics Methodology (AgileDSAM) –a Scrum- aligned Development Methodology for Big Data Software Systems in Small Business

IMPACTO SOCIAL (señalar el impacto logrado): Proporcionar una metodología de desarrollo agil y gratuito para proyectos de ciencia de datos, en pequeñas y medianas empresas de México.

INDICAR SEGÚN CORRESPONDA: **SI, NO, NA (No Aplica)**

	Elementos para la revisión académica del trabajo de tesis o trabajo práctico:
SI	El trabajo es congruente con las LGAC del programa de posgrado
SI	La problemática fue abordada desde un enfoque multidisciplinario
SI	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	Los resultados presentados en el trabajo son de gran relevancia científica, tecnologica o profesional según el área
SI	El trabajo demuestra más de una aportación original al conocimiento de su área
NO	Las aportaciones responden a los problemas prioritarios del país
SI	Generó transferencia del conocimiento o tecnológica
SI	Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
	El egresado cumple con lo siguiente:
SI	Cumple con lo señalado por el Reglamento General de Posgrados
SI	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, predoctoral, etc.)
SI	Cuenta con los votos aprobatorios del comité tutorial
N.A	Cuenta con la carta de satisfacción del Usuario (En caso de que corresponda)
SI	Coincide con el título y objetivo registrado
SI	Tiene congruencia con cuerpos académicos
SI	Tiene el CVU de la SECIHTI actualizado
SI	Tiene el o los artículos aceptados o publicados y cumple con los requisitos institucionales (en caso de que proceda)
	**En caso de Tesis por artículos científicos publicados (completar solo si la tesis fue por artículos)*
N.A	Aceptación o Publicación de los artículos en revistas indexadas de alto impacto según el nivel del programa
N.A	El (la) estudiante es el primer autor(a)
N.A	El (la) autor(a) de correspondencia es el Director (a) del Núcleo Académico
N.A	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
	***En caso de Tesis por Patente*
N.A	Cuenta con la evidencia de solicitud de patente en el Departamento de Investigación (anexarla al presente formato)

Con base en estos criterios, se autoriza continuar con los trámites de titulación y programación del examen de grado:

Sí ___X___
No ______

FIRMAS

Elaboró:

*NOMBRE Y FIRMA DEL(LA) CONSEJERO(A) SEGÚN LA LGAC DE ADSCRIPCION: P.P. Dr. José Antonio Guerrero Díaz de León

** En caso de conflicto de intereses, firmará un revisor miembro del NA de la LGAC correspondiente distinto al director o miembro del comité tutorial, asignado por el Decano.*

NOMBRE Y FIRMA DEL COORDINADOR DE POSGRADO: Dr. Francisco Javier Álvarez Rodríguez

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: Dr. Alejandro Padilla Díaz

Autorizó:

NOMBRE Y FIRMA DEL DECANO: Mtro. en C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 24 fracción V del Reglamento General de Posgrado, que a la letra señala entre las funciones del Consejo Académico: Proponer criterios y mecanismos de selección, permanencia, egreso y titulación de estudiantes para asegurar la eficiencia terminal y la titulación y el Art. 28 fracción D, atender, asesorar y dar el seguimiento del estudiantado desde su ingreso hasta su titulación.

Elaborado por: D. Apoyo al Posg.
Revisado por: D. Control Escolar/D. Gestión de Calidad.
Aprobado por: D. Control Escolar/ D. Apoyo al Posg.

Código: DO-SEE-FO-15
Actualización: 02
Emisión: 12/08/25

Review of Agile SDLC for Big Data Analytics Systems in the Context of Small Organizations Using Scrum-XP

Gerardo Salazar-Salazar
Electronic System Department
Autonomous University of
Aguascalientes, Mexico
gerardo.salazar@edu.uaa.mx

Manuel Mora
Information Systems Department
Autonomous University of
Aguascalientes, Mexico
jose.mora@edu.uaa.mx

Hector Duran-Limon
Information Systems Department
University of Guadalajara, Mexico
hduran@cucea.udg.mx

Francisco Alvarez-Rodriguez
Computer Science Department, Autonomous University of
Aguascalientes, Mexico
francisco.alvarez@edu.uaa.mx

Angel Munoz-Zavala
Statistics Department, Autonomous University of
Aguascalientes, Mexico
eduardo.munoz@edu.uaa.mx

Abstract: *Software development using agile System Development Life Cycles (SDLC), such as Scrum and XP, has gained important acceptance for small businesses. Agile approaches eliminate barriers to required organizational, technical, and economic resources usually necessary when rigorous software development approaches, through heavyweight methodologies (e.g., Rational Unified Process (RUP)) or heavyweight international standards (e.g., ISO/IEC 12207) are used. However, despite their high popularity in small businesses, their utilization is scarce in the emergent domain of Big Data Analytics Systems (BDAS). Consequently, small businesses interested in deploying BDAS lack systematic academic guidance regarding agile SDLC for BDAS. This research, thus, addresses this research gap, and reports an updated comparative study of three of the main proposed SDLCs for BDAS (Cross-Industry Standard Process for Data Mining CRISP-DM), Two mains were Microsoft Team Data Science Process (TDSP), and Domino Data Science Lifecycle (DDSL)) in the current BDAS development literature, against a Scrum and Extreme Programming (Scrum-XP) SDLC. For this aim, a Pro Forma of a generic Scrum-XP SDLC is used to examine the conceptual structure, i.e., roles, phases-activities, roles, and work products-of these two SDLCs. Hence, this comparative study provides theoretical and practical insights on agile SDLC for BDAS adequate for small businesses and calls for further conceptual and empirical research to advance toward an agile SDLC for BDAS supported by academia and used in practice*.

Keywords: *Big data analytics systems, agile system development life cycle, Scrum-XP, CRISP-DM, TDSP and DDSL, small business*.

Received June 6, 2024; accepted October 9, 2024
https://doi.org/10.34028/iajit/21/6/12

1. Introduction

The agile Software Development Paradigm (SDP) emerged in the Software Engineering discipline about 20 years ago [32], as an alternative SDP to the dominant rigorous SDP [34] also known as plan-driven or heavyweight SDP -. Core literature on agile SDP [1, 23, 26, 32, 34] indicates that this paradigm was an overall response to address software development projects highly dynamic given changing user and system requirements, using new technological advances, and the business competitive pressures for shorting delivery timeframe from years to months. Additionally, there was also identified a strong disappointment with the current rigorous SDP because end-users and developers considered it a documentation-based bureaucracy that could be unnecessary for small software development projects [1, 32]. Consequently, formed an Agile Alliance consortium with several relevant practitioners [9] and declared the well-known Agile Manifesto that stands for one overall aim, four agile values, and twelve agile principles [9]. Table 1 reports these aims, values, and the twelve principles grouped in the categories of agile outcome, agile team, agile project, and agile design principles from [9, 56].

Nowadays, this agile SDP has permeated strongly in both small, medium, and large organizations [33, 34, 80] and co-exists with the rigorous SDP [7, 12, 48]. Several agile Software Development Life Cycles (SDLC) have been proposed [1, 34], but the most used and known at present days [22] are Scrum [74] and Extreme Programming (best known as XP) [8]. An SDLC refers to “the software processes used to specify and transform software requirements into a deliverable software product,” [14]. An SDLC is usually represented as a software development process model [14] of phases-activities, roles, and work products proposed to increase the likelihood of delivering software on the expected

October 20, 2025

Dear Gerardo Salazar Salazar, Manuel Mora, Hector A. Duran-limon, Francisco Alvarez-rodriguez, Angel MuÃ±oz-zavala

I am pleased to inform you that your manuscript titled as "A Comparative Review of the Main Heavyweight and Agile Sdlc Development Life Cycles for Bi Data Analytics Systems (bdas): 2000-2023 Period" (Manuscript Number: IAJIT-2025-03-356 was accepted for publication in the International Arab Journal of Information Technology. You could check your possible publication date at your author page.

You may login to your author account page, and visit accepted articles section in order to get offical/formal acceptance letter as PDF.

I would like to remind that you could send your future manuscripts to International Arab Journal of Information Technology.

Sincerely yours,
Prof. Mohammad Hassan,
Editor-in-Chief
IAJIT
Zarqa University
Zarqa, Jordan
Tel: +(962)-5-3821100
WhatsApp: +(962)-780011551

A COMPARATIVE REVIEW OF THE MAIN HEAVYWEIGHT AND AGILE SDLC DEVELOPMENT LIFE CYCLES FOR BI DATA ANALYTICS SYSTEMS (BDAS): 2000-2023 PERIOD

line 1: 1st Given Name Surname
dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 1st Given Name Surname
dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 1st Given Name Surname
dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 1st Given Name Surname
dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Countryline 5: email address or ORCID

Abstract: *Big Data Analytics Systems (BDAS) are relevant software systems for business descriptive, predictive, or prescriptive purposes. BDAS emerged due to the concurrent availability of Analytics techniques and affordable sources of massive business internal and external data. Nowadays, BDAS is implemented in diverse domains such as Marketing, Healthcare, Finance, Manufacturing, Logistics, Education, and Tourism. However, despite the accelerated technological progress on BDAS algorithms and platforms, their development has been conducted mainly using ad-hoc practical guidelines or old rigor-oriented heavyweight development life cycles. Nowadays, the business competitive environment demands agile BDAS development life cycles, and in the last years, the first ones have been proposed. However, studies contrasting plan-driven – i.e. heavyweight – vs agile development life cycles for BDAS are still scarce in the literature. In this research, we address this knowledge gap and present a comparative review between major heavyweight SDLCs (CRISP-DM, KDD, SEMMA, and BDPL) using a generic Scrum-XP lifecycle as a theoretical agile development lifecycle. The main SDLCs considered agile (TDSP, ASUM, DDS). The previous SDLCs were selected through selective systematic literature, using the research method (SSLR) for the period 2000-2023. Where the main objective is the comparison of its conceptual structure, that is, roles, phases-activities, and products. This comparative review provides theoretical and practical insights to discriminate both approaches for BDAS development and requires further conceptual and empirical research.*

Keywords: *Big Data Analytics Systems (BDAS); agile and heavyweight BDAS SDLC; Scrum-XP development workflow; KDD, SEMMA, CRISP-DM, and BDPL; TDSP, ASUM, and DDS.*

Received ……; accepted …
https://doi.org/

1. INTRODUCTION

Nowadays, data exploitation has become a fundamental aspect for companies aiming to generate competitive advantages and improve decision-making processes. The increasing number of Big Data technologies has driven organizations and businesses to implement Big Data Analytics Systems (BDAS) that enhance their commercial value [1]. In recent decades, data diversity has increased significantly in terms of origin, format, and modalities, allowing companies to leverage various techniques for data exploitation, such as machine learning, data management, visualization, causal inference, and other related fields [2]. The adoption of BDAS technology can significantly improve production efficiency and decision-making processes within organizations and businesses [3,4].

There are various success stories of large corporations effectively leveraging BDAS projects, enabling them to gain competitive advantages and dominate their competitors in areas such as social networks, search engines, e-commerce, and video streaming services [5].

Outlook

The Editor-in-Chief has placed your submission on hold - see the message (IJIKM, PID 12097)

Desde Informing Science Institute <notifications@informingscience.org>
Fecha Mié 21/05/2025 01:14 PM
Para GERARDO SALAZAR SALAZAR <gerardo.salazar@edu.uaa.mx>

Do not reply to this email. To contact ISI click here.

Article: DESIGN AND USABILITY EVALUATION OF AGILEDSA: A SCRUM-XP ALIGNED SDLC FOR BIG DATA ANALYTICS SYSTEMS IN SMALL BUSINESS

Dear Gerardo Salazar-Salazar,

Thank you for your recent submission PID-12097 *"DESIGN AND USABILITY EVALUATION OF AGILEDSA: A SCRUM-XP ALIGNED SDLC FOR BIG DATA ANALYTICS SYSTEMS IN SMALL BUSINESS"* to Interdisciplinary Journal of Information, Knowledge, and Management (IJIKM).

I have had an opportunity to read through your paper before its acceptance and submission for review. Your paper is of interest to our readership, and it makes a valued contribution.

I would be pleased to send the paper for review by an ad hoc set of external reviewers. **But first, I would like you complete an initial round of major revision to address some issues, mainly in content management**. My suggestions and amendments are intended to assist you in improving your paper so that it has the best chance of receiving a positive outcome from our board of reviewers.

Specifically,

1. **Although the paper cites many references throughout, it needs a dedicated “Literature Review” section to critically analyze in great depth previous studies directly on the problem – SDLC for BDAS.**
2. **The paper also needs a “Background” section where you can gather all the introductory paragraphs of basic concepts/models together, as preparation to presenting your SDLC for big data analytics systems, AgileDSA.**
3. **Then, “The AgileDSA Model”, where you would focus on (1) documenting the process and methodology of developing**

DEDICATION

To my parents,

For being the unwavering foundation at every step, I have taken. For your unconditional love, your example in life, and your constant sacrifices. For teaching me, through your daily efforts, that dreams are achieved through hard work, humility, and perseverance. This achievement would not have been possible without you.

With all my love and gratitude, this thesis belongs to you as much as it does to me.

To Paola,

For being by my side every step of the way, for your unconditional love, your unwavering support, and your infinite patience.

Thank you for believing in me even in the moments when I doubted myself.

A mis padres,

Por ser el pilar firme en cada paso que he dado. Por su amor incondicional, su ejemplo de vida y su constante sacrificio, Por enseñarme, con su esfuerzo diario, que los sueños se alcanzan con trabajo, humildad y perseverancia. Este logro no habría sido posible sin ustedes.

Con todo mi amor y gratitud, esta tesis les pertenece tanto como a mí.

A Paola,

Por estar a mi lado en cada paso de este camino, por tu amor incondicional, tu apoyo constante y tu paciencia infinita.

Gracias por creer en mí incluso en los momentos en que yo dudaba.

ACKNOWLEDGEMENTS

Upon completing this significant stage, I would like to express my deepest gratitude to those who made this achievement possible, both academically and personally.

First and foremost, I am profoundly grateful to my parents, whose love, example, and constant sacrifice have been the foundation of everything I have accomplished. Thank you for teaching me, through actions more than words, the value of hard work, responsibility, and perseverance.

To my wife, Paola, my tireless companion, thank you for your unconditional support, your patience in the most challenging moments, and for believing in me even when I doubted myself. Your love, understanding, and strength were the driving force that kept me going.

To my best friend, David, thank you for walking with me through every stage of this doctoral journey. Sharing this path with you was invaluable. Your support and sense of humor provided refuge during the most difficult days and served as a constant source of motivation.

I also extend my sincere thanks to my thesis advisor, Dr. José Manuel Mora Tavarez, for his expert guidance and for accompanying me with intellectual rigor and generosity throughout this process, as well as to the committee members for their valuable contributions.

Additionally, I would like to express my deepest appreciation to the University of Seville and Dr. José Luis Roldán Salgueiro for welcoming me during my academic exchange as part of my doctoral training. This experience not only enriched my research but also broadened my educational and cultural perspective, significantly strengthening my personal and professional development.

Finally, I would like to thank the Autonomous University of Aguascalientes for the support that made this research possible. I am grateful to the National Council of Humanities, Sciences, and Technologies (CONAHCyT) for the support of my studies during the past 4 years.

Thank you to all who, in one way or another, walked alongside me on this journey.

CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

ABSTRACT

Currently, data utilization has become an essential component for organizations seeking to gain competitive advantages and optimize their decision-making processes.

The rise of Big Data-related technologies has prompted many companies to implement Big Data Analytics Systems (BDAS). In recent decades, there has been a significant increase in data diversity—in terms of origin, format, and modality—which enables the use of a wide range of techniques for analysis, such as machine learning, data management, data visualization, and causal inference, among others.

Several successful cases of major corporations that have implemented BDAS projects highlight the need for organizations to understand how to effectively manage these types of initiatives. In this context, the adoption of well-structured methodologies becomes fundamental for the efficient development of BDAS projects. It is common for the software development process to face challenges, particularly due to changing requirements, which further emphasizes the need for a solid methodological framework.

A distinctive feature of BDAS is its ability to process and analyze large volumes of data in very short time frames, which entails high technological and methodological demands. For this reason, this research focused on the design and development of a methodology tailored to BDAS projects, with the goal of supporting small and medium-sized enterprises in generating value using data science.

The proposed methodology is based on widely recognized agile frameworks, such as SCRUM and XP, as well as on methods specifically developed for BDAS projects. The results obtained through the developed Electronic Process Guide (EPG) revealed favorable metrics in aspects such as agility, usefulness, ease of use, compatibility, value, and attitude, even exceeding the initial expectations for the proposed methodology.

RESUMEN

En la actualidad, el aprovechamiento de los datos se ha vuelto un componente esencial para las organizaciones que buscan obtener ventajas competitivas y optimizar sus procesos de toma de decisiones. El auge de tecnologías relacionadas con Big Data ha motivado a muchas empresas a implementar Sistemas de Análisis de Big Data (BDAS). En las últimas décadas, se ha observado un incremento notable en la diversidad de los datos, tanto en su origen como en su formato y modalidad, lo que permite emplear una amplia gama de técnicas para su análisis, tales como el aprendizaje automático, la gestión de datos, la visualización de información, la inferencia causal, entre otras.

Diversos casos exitosos de grandes compañías que han implementado proyectos BDAS evidencian la necesidad de que las organizaciones comprendan cómo gestionar eficazmente este tipo de iniciativas. En este contexto, la adopción de metodologías bien estructuradas se vuelve fundamental para el desarrollo eficiente de proyectos BDAS. Es común que el proceso de desarrollo de software enfrente dificultades, particularmente por la variabilidad de los requerimientos, lo cual subraya aún más la necesidad de contar con un marco metodológico sólido.

Una característica distintiva de los sistemas BDAS es su capacidad para procesar y analizar grandes volúmenes de datos en tiempos muy reducidos, lo que implica una elevada demanda tanto tecnológica como metodológica. Por ello, esta investigación se orientó al diseño y desarrollo de una metodología adaptada a proyectos BDAS, con el objetivo de apoyar a pequeñas y medianas empresas en la generación de valor mediante el uso de la ciencia de datos.

La propuesta metodológica se basa en marcos ágiles ampliamente reconocidos, como SCRUM y XP, así como en metodologías especialmente desarrolladas para proyectos BDAS. Los resultados obtenidos mediante la Guía Electrónica de Procesos (EPG) desarrollada revelaron métricas favorables en aspectos como agilidad, utilidad, facilidad de uso, compatibilidad, valor y actitud, superando incluso las expectativas planteadas para la metodología propuesta.

CONTRIBUTIONS

Research stays in April 2024 at the University of Seville (US) with Dr. José Luis Roldán Salgueiro, Associate Professor at the Faculty of Economics and Business Sciences, University of Seville (US Senior Editor of The DATA BASE for Advances in Information Systems. Guest Editor of the European Journal of Information Systems (EJIS).

“A Selective Comparative Review of CRISP-DM and TDSP Development Methodologies for Big Data Analytics Systems” (Springer Book). Gerardo Salazar-Salazar, Manuel Mora, Hector A. Duran-Limon, and Francisco Javier Álvarez Rodríguez. ISSN 2569-7072 ISSN 2569-7080 (electronic) Transactions on Computational Science and Computational Intelligence ISBN 978-3-031-40955-4 ISBN 978-3-031-40956-1 (eBook) https://doi.org/10.1007/978-3-031-40956-1

“Review of Agile SDLC for Big Data Analytics Systems in the Context of Small Organizations Using Scrum-XP” (JCR). Gerardo Salazar-Salazar, Manuel Mora, Hector A. Duran-Limon, Francisco Javier Álvarez Rodríguez and Angel Munoz-Zavala. The International Arab Journal of Information Technology. 10.34028/iajit/21/6/12

“DESIGN AND USABILITY EVALUATION OF AGILEDSA: A SCRUM-XP ALIGNED SDLC FOR BIG DATA ANALYTICS SYSTEMS IN SMALL BUSINESS” (SCOPUS). Gerardo Salazar-Salazar, Manuel Mora, Hector A. Duran-Limon, Francisco Javier Álvarez Rodríguez and Angel Munoz-Zavala.

“A COMPARATIVE REVIEW OF THE MAIN HEAVYWEIGHT AND AGILE SDLC DEVELOPMENT LIFE CYCLES FOR BI DATA ANALYTICS SYSTEMS (BDAS): 2000-2023 PERIOD” (JCR). Gerardo Salazar-Salazar, Manuel Mora, Hector A. Duran-Limon, Francisco Javier Álvarez Rodríguez and Angel Munoz-Zavala.

1. INTRODUCTION

1.1 CONTEXT OF THE RESEARCH PROBLEM

Nowadays, many organizations in Mexico are undergoing digital transformation processes, which require the development of useful, secure, and valuable software applications that must be available within short periods, generating high-quality services that meet the needs of both organizations and their clients.

The development of these applications requires the use of agile development methodologies that allow for the rapid and continuous delivery of functional software (usually within periods of 4 to 8 weeks instead of 4 to 8 months).

This has led the Agile Software Development (ASD) paradigm to gain significant attention in software engineering, largely due to its flexible approach to managing requirements volatility and emphasis on wide collaboration between clients and developers (Abrahamsson et al., 2002). This provides us with the main benefits of rapid response to change, allowing for client intervention in the process, breaking the project or product into intervals, eliminating unnecessary tasks, among others. The above makes it easier for organizations to adjust to the project's schedule and budget, generating products with great flexibility and quality.

However, the use of agile methodologies in Data Science has been applied and studied with moderation, as agile methodologies have a greater focus on software development platforms. On the contrary, the project lifecycles of Data Science are currently in the same situation as software development before the introduction of agile methodologies, with problems in delivery times, early generation of value, and risk reduction of failure (Grady et al., 2017).

The growing production and collection of data involved in Data Science projects generate the need for a framework that allows for efficient data processing.

In this research, we believe that applying the agile approach to the development of Data Science projects can generate benefits in the usefulness, security, and quality of the project, while maintaining the established schedule and planned budgets for the project.

1.2 MOTIVATION AND RELEVANCE OF THE RESEARCH PROBLEM

Several global business studies report that the use of agile development methodologies is a common practice in organizations of all types (large, medium, and small companies). Organizations are developing more systems with new forms of organization and work with a human-centered purpose, and the roles and responsibilities of individuals are changing as agility implies a new mentality (Leybourn, 2013; Oestereich & Schröder, 2017). This means that more intelligent solutions are expected in the future. Similarly, the market for platforms focused on Data Science has opened the possibility of growth in the 2020s.

With these two technological trends and the current need for multiple web-based software systems and the development of intelligent systems, commercial organizations require agile software development methodologies that can produce useful, easy-to-use, secure, and valuable software (i.e., adapted to the product quality). It is also necessary to use these agile software development methodologies to help organizations meet the project schedule and budget.

1.3 FORMULATION OF THE RESEARCH PROBLEM

1.3.1 RESEARCH PROBLEM

Consequently, based on the previous research context described, we can identify the research problem directly as the lack of development methodologies for Data Science Projects that are considered by the software developers as agile, easy to use, useful, compatible, and valuable.

1.3.2 RESEARCH QUESTIONS AND HYPOTHESES

- **RQ.1** What is the state of the art – contributions and limitations- on agile and non-agile development methodologies for Big Data-Data Science-Analytics Software Systems?
- **H0.1** There is no need for an agile development methodology for Big Data-Data Science-Analytics Software Systems.

- **RQ.2** What is the state of the art – capabilities and limitations – of open-source development platforms for Big Data-Data Science-Analytics Software Systems?
- **H0.2** There are no available open-source development platforms for Big Data-Data Science-Analytics Software Systems that can be satisfactorily evaluated in the technical, end-user, and organizational dimensions.

- **RQ.3** What elements of Agile Development and Big Data-Data Science-Analytics Development Methodologies can be used to elaborate an Agile Development Methodology for Big Data-Data Science-Analytics Software Systems that can be evaluated as theoretically valid from a Panel of Experts?
- **H0.3** There are no elements of Agile Development and Big Data-Data Science-Analytics Development Methodologies that can be used to elaborate an Agile Development Methodology for Big Data-Data Science-Analytics Software Systems that can be evaluated as theoretically valid from a Panel of Experts.

- **RQ.4** Can the new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems be documented in an Electronic Process Guide (EPG), and be evaluated as agile, useful, easy to use, compatible, and valuable from a pilot group of Big Data-Data Science-Analytics academics and practitioners?
- **H0.4.1** The new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems cannot be documented in an Electronic Process Guide (EPG).
- **H0.4.2** The new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems is not considered agile, useful, easy to use, compatible, and valuable from a pilot group of Big Data-Data Science-Analytics academics and practitioners.

1.3.3 GENERAL AND SPECIFIC RESEARCH OBJECTIVES

To design conceptually a Development Methodology for Big Data-Data Science-Analytics Software Systems, and document it in an Electronic Process Guide, that is evaluated as agile, useful, easy to use, compatible, and valuable for a pilot group of Big Data-Data Science-Analytics academics and practitioners.

1.3.4 CONTRIBUTIONS AND DELIVERABLES OF THE RESEARCH

In this research proposal, it is expected to produce the following products:

1. For the Software Engineering Theory
   - 1 research paper for an indexed journal with a theoretical analysis on “The State of the Art on Open-Source Data Science – Data Analytics Development Platforms”.
   - 1 research paper for an indexed journal with a theoretical analysis on “The State of the Art on Development Methodologies for Data Science – Data Analytics Projects”.
   - 1 submitted research paper for an indexed journal with the theoretical analysis and empirical evaluation of the AgileDSA Methodology – an agile Methodology for Big Data-Data Science-Analytics Software Systems in Small Business.
2. For the Software Engineering Practice
   - 1 new AgileDSA Methodology – an agile Methodology for Big Data-Data Science-Analytics Software Systems in Small Business, available in a web-based, free-cost access EPG (Electronic Process Guideline).
   - 1 new PhD graduate in the Software Engineering area.

1.4 GENERAL DESCRIPTION OF THE RESEARCH METHODOLOGY

In this research, we propose to use a Design Science Research approach (Vom Brocke et al., 2020; Peffers et al., 2007). "Design Science Research (DSR) is a problem-solving paradigm that seeks to improve the scientific and technological knowledge base through the creation of innovative artifacts that solve problems and improve the environment in which they are instantiated. The results of DSR include both newly designed artifacts, represented by constructions, and/or models, and/or methods, and/or instantiations, as well as design knowledge (DK)."

1.4.1 OVERVIEW OF THE RESEARCH METHODOLOGY

The specific DSR methodology used is the Design Science Research Methodology proposed by Peffers et al. (2007). It has six activities described below:

- **Activity 1: Problem identification and motivation.** "*Define the specific research problem and justify the value of a solution. Justifying the value of a solution accomplishes two things: it motivates the researcher and the audience of the research to pursue the solution and to accept the results, and it helps to understand the reasoning associated with the researcher's understanding of the problem".*
- **Activity 2.1: Define the objectives for a solution***. "Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. The objectives can be quantitative, such as terms in which a desirable solution would be better than current ones, or qualitative, such as a description of how a new artifact is expected to support solutions to problems not hitherto addressed".*
- **Activity 2.2: Review the State of the Art.** Review the state of the art on the main element to be designed and identify the main contributions and limitations.
- **Activity 3: Design and development.** *Create the artifact. Such artifacts are potentially constructing, models, methods, or instantiations (each defined broadly). Conceptually, a design research artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture, and then creating the actual artifact.*
- **Activity 4: Demonstration.** *"Demonstrate the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity".*
- **Activity 5: Evaluation.** *"Observe and measure how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from the use of the artifact in the demonstration. At the end of this activity, the researchers can decide whether to*

iterate back to activity 3 to try to improve the effectiveness of the artifact or to continue to communicate and leave further improvement to subsequent projects". The specific Evaluation methods to be used will be: 1) Evaluation Conceptual from a Panel of Experts; 2) Evaluation from a Proof of Concept, and 3) Empirical survey-based evaluation from a pilot sample of Software Engineering professionals.

- **Activity 6: Communication.** *"Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate".*

1.4.2 TIMELINE – SEMESTERS, ACTIVITIES AND DELIVERABLES

Table 1.1 Timeline, semesters, activities, and deliverable.

Phases	2021	2022	2023	2024
Activities 1 and 2.1 a) Background and history of the problem. b) Problematic situation. c) Type and purpose of research. d) Relevance. e) Objectives, questions, and hypotheses/research propositions.	X			
Activity 2.2 Review the State of the Art a) Theories bases. b) Studies related. c) Contributions and limitations of related studies.	X	X		
Activity 3 Design and Development of Artifact a) Application or creative-deductive relational conceptual design model.		X	X	
Activities 4 and 5 – Demonstration and Evaluation a) Validation of content by a panel of experts. b) Validation by logical argument. c) Validation for proof of concept of the artifact.			X	X
Activities 6 – Communication a) Write and submit research paper 1. b) Write and submit research paper 2. c) Write and submit research paper 3.		X	X	X

2. RESEARCH METHODOLOGY

2.1 MAIN ACTIVITIES

The scientific research process can be carried out using several methods (Ackoff, 1962). In the case of this thesis, we will use a combination of three methods that allow us to better manage the development of the methodology of this thesis, as well as various alternative approaches, development, and evaluation. The three research methods that we will use in this thesis will be combined to obtain the maximum performance of each and therefore a better result in the development of our methodology.

The first research method that we rely on is the conceptual method (Mora, 2009). The second research method that we will use is the DSRM (Peffers et al., 2007), and finally, we will use the 3 DSR cycles research method (Hevner, 2007).

Concept-based research was used when the designed objects were evaluated in the final stage of this thesis, since, in general, there are no physical laws to apply to the designed objects in this thesis, and it is also difficult to apply mathematical models or methods to evaluate the designed objects. The conceptual method is considered the main source of generating new theories, models, or conceptual frameworks. In the field of information systems, this method is considered an important part of the possible repertoire of research methods. This method consists of four phases: Phase I, Formulation of the Research Problem; Phase II, Analysis of Related Works; Phase III, Application or Design of the Conceptual Model; and finally, Phase IV, Validation of the Applied or Designed Conceptual Model (Mora, 2009).

These phases can be observed in Table 2.1 Conceptual-based Design Research Phases.

Table 2.1 Conceptual-based Design Research Phases (Mora et al., 2012).

Conceptual-based Design Research Phases
Phase I. Formulation of Research Problem • Background and history of the problem. • Problematic situation. • Type and purpose of research. • Relevance. • Objectives, questions, and hypotheses / research propositions.
Phase II. Analysis of Related Work. • Theories bases. • Studies related. • Contributions and limitations of related studies. • Selection/design of general conceptual framework.
Phase III. Conceptual Design of Artifact. • Application of creative-deductive relational conceptual design model.
Phase IV. Validation of Designed Artifact. • Validation of Content by a Panel of Experts. • Validation by Logical Argumentation. • Validation by Proof of Concept of Designed Artifact. • Empirical Validation by a Pilot Survey or Case Study or Experimental Study.

At the same time, the conceptual research method was merged with the DSRM method, which allowed us to better document the development of the methodology during the development of this thesis. The objective of a DSRM process is to improve the production, presentation, and evaluation of research.

Figure 2.1 Design Science Research Methodology DSRM) The Process Model shows the 6 activities that make up the DSRM research method as a nominal sequence. The figure also shows a brief description in general terms of what the method proposes in each of these 6 activities. This method is used to generate artifacts in information systems that solve an instance of a problem.

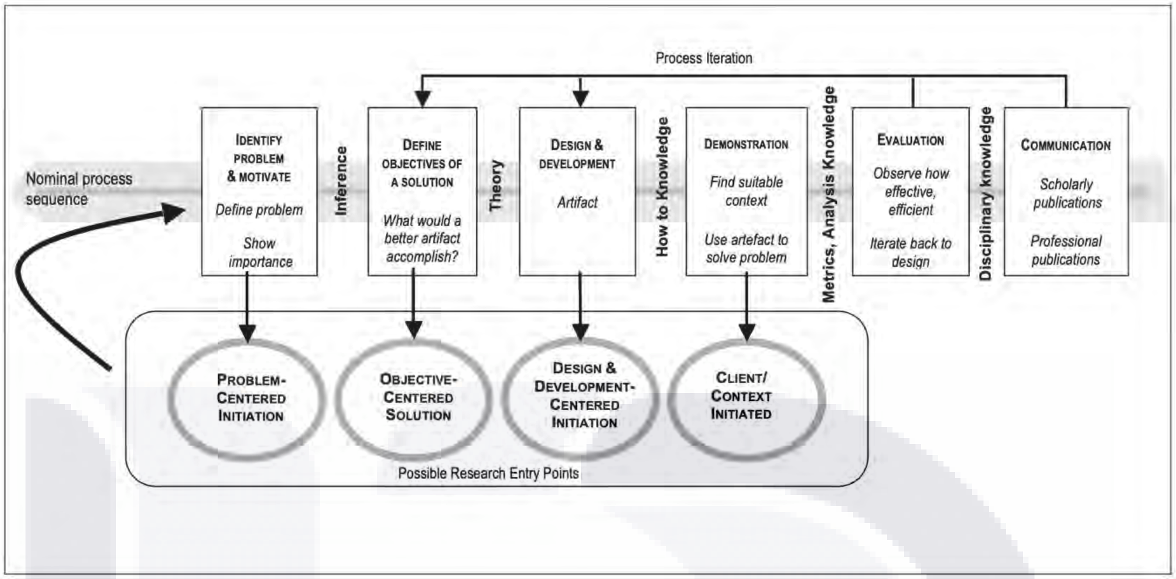

Process Iteration
Nominal process sequence
IDENTIFY PROBLEM & MOTIVATE
Define problem
Show importance
Inference
DEFINE OBJECTIVES OF A SOLUTION
What would a better artifact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to Knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, Analysis Knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM-CENTERED INITIATION
OBJECTIVE-CENTERED SOLUTION
DESIGN & DEVELOPMENT-CENTERED INITIATION
CLIENT/ CONTEXT INITIATED
Possible Research Entry Points

Figure 2.1 Design Science Research Methodology (DSRM) Process Model (Peffers et al., 2007).

Finally, the implementation of Design Science Research (DSR) aims to improve our understanding of information systems through the creation of technological artifacts. These created artifacts embody the solution to a problem (Hevner et al., 2004).

This process is represented in Figure 2.2, Design Science Research Cycles, which shows the function of each of the cycles represented in the two main research approaches proposed by Hevner. The relevance cycle links the contextual environment with the design science activities with the scientific knowledge base. The design cycle iterates between core activities of artifact and process design construction, artifact and process evaluation, and research design (Hevner, 2007).

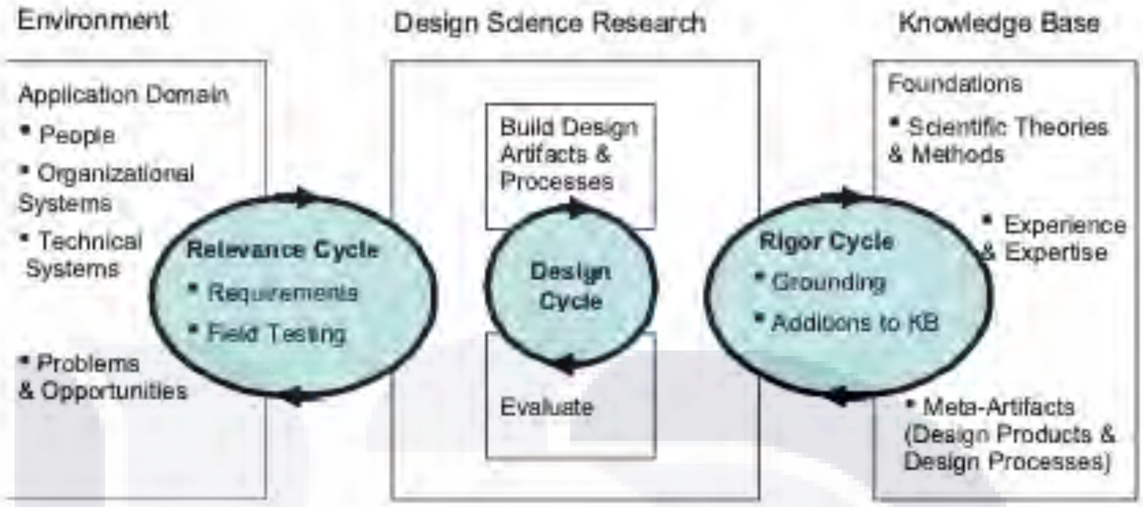


Figure 2.2 Design Science Research Cycles (Hevner, 2007).

This Ph.D. research uses the Design Science Research Methodology (DSRM) (Peffers et al., 2007) complemented with additional specific research methods. These methods are: Selective Systematic Literature Review method (Cooper, 1988), Conceptual Design (Mora et al., 2009), Heuristic Design with Means-Ends Analysis (Newell & Simon, 1972; Mora et al., 2023), Conceptual Verification by Panel of Experts (Hevner et al., 2004; Beecham et al., 2005), Empirical Validation with Statistical Analysis (Wohlin et al., 2012; Chin, 2009), and Guide for Scientific Reports in Software Engineering (Shaw, 2003). Table 2.2 summarizes steps, purpose, complementary research methods, and expected outcomes.

Table 2.2 Design Science Research Methodology (DSRM) with complementary research methods.

Step	Purpose	Complementary research methods	Outcomes
1) Design problem identification and motivation.	To state the expected overall research goal that delimits the scope of the research, the research questions that focus on the knowledge gaps of interest, and the motivations to pursue the research design. (For these aims is required to conduct a Review of the State of the Art on the specific problem.).	• Conceptual Literature Review (CLR), or • Systematic Literature Review (SLR), or • Selective Systematic Literature Review (SSLR).	• Research overall goal statement. • Research questions. • Research motivation statements. • Review of the State of the Art.
2) Definition of the design objectives and restrictions for the expected artifact.	To define the specific design objectives (i.e. expected qualities in the designed artifact), design restrictions (i.e. the limitations on time, cost and resources utilized to design the artifact), design approach (i.e. analytics, axiomatic or heuristic), design theoretical sources (i.e. the design materials), and design components (i.e. the specific design building-blocks).	• Conceptual Design.	• Design objectives. • Design restrictions. • Design approach. • Design theoretical sources. • Design components.
3) Design and development of the artifact.	To design and implement the expected artifact guided-controlled by the design objectives and restrictions, and using the agreed design approach, design theoretical sources and design components.	• Conceptual Design.	• Conceptual designed artifact. • Implemented designed artifact.
4) Demonstration of the artifact (Proof of Concept).	To demonstrate the designed and implemented artifact and conduct initial verification.	• Conceptual Verification by Panel of Experts.	• Conceptual Verification.
5) Evaluation of the artifact.	To conduct empirical evaluation of the designed and implemented artifact.	• Empirical Validation and Statistical Analysis by a Pilot Sample of Evaluators.	• Empirical Validation with Statistical Analysis.
6) Communication of research results.	To generate a structured scientific report (i.e. Thesis, Technical Report, Chapter, Conference Proceeding document, or Journal article) of results and communicate them in academic outlets.	• Guidelines for Scientific Reports in Software Engineering.	• Structured Scientific Report.

2.2 OBJECT AND SUBJECTS OF STUDY

The development of this thesis is based on current agile development methodologies such as Scrum and XP, as well as Analytics/Data Science project development methodologies, and finally Agile Analytics/Data Science development methodologies. The validation of the developed methodology was evaluated with a pilot sample of software professionals and academics interested in agile development methods for Analytics/Data Science projects, through a usability perception measurement instrument where ease of use, usefulness, compatibility, and how valuable the methodology is were evaluated. The instrument is commonly used in scientific literature (Moore & Benbasat, 1991).

2.3 MATERIALS AND EQUIPMENT

- Research articles, chapters, and conference presentations related to Agile development methods and Data Science.
- Official documents and literature associated with Agile development methods, Data Science, Analytics, software engineering, and small Data Sciences.
- Laptop computer equipment.
- VM server in the LabDC-2004 laboratory.
- Open-Source development environments/platforms for the development of Analytics or Data Sciences projects (R + Python for R + Weka for R + Shiny + Radiant and web libraries such as Weka + Shiny + Radiant).

2.4 RESEARCH EVALUATION METHODS

According to Hevner et al. (2004), the validation techniques are the following:

- Observational: Through a case study or a field study, or a survey study.
- Analytics: Through statistical analysis or dynamic analysis, or optimization.
- Experimental: Through a controlled experiment or simulation.
- Testing: Through functional testing or structural testing.
- Descriptive: Through information, arguments, of demonstration cases.

Peffers et al. (2007) mention in the DSRM methodology that when applying the generated artifact in a specific case, results will be generated that can be evaluated with relevant metrics to be compared with the objectives defined from the beginning. The authors also mention that if the evaluation is conclusive, that is, it generates relevant conclusions about the artifact, the next step is to communicate the artifact to the relevant entities. Otherwise, if the artifact is not conclusive, it will be necessary to rethink the objectives or the elaboration of the artifact to obtain conclusive results (Peffers et al., 2007).

The Survey research method is a way to collect data and information from a group of individuals or a specific population. It involves using standardized or structured questionnaires to gather data from a representative sample.

Surveys can be conducted through various means, such as face-to-face interviews, telephone interviews, paper-based surveys, or online surveys. The main objective is to gather information about people's opinions, attitudes, behaviors, or other relevant characteristics.

The surveys that will be applied in this thesis can be seen in Appendix 7, with the three baskets that will be applied to experts in the Big Data sector.

2.5 RESTRICTIONS AND LIMITATIONS

Due to the complexity and limited use of methodologies for Analytics / Data Science project development in a small and medium-sized business environment in Mexico that utilizes Big Data, designing and developing a methodology specifically for this sector is a largely complex task. Therefore, this thesis will have the following limitations:

- The periods available for the development of the methodology are 3 to 4 years.
- Development costs, only the budget for the doctoral study is available.
- The scope of the projects for this methodology is developed for micro and small projects with participants of 5 to 10 people, with periods of 3 to 6 months, and with limited budgets.

3. THEORETICAL BACKGROUND

3.1 FOUNDATIONS OF SOFTWARE ENGINEERING

Software Engineering is a branch of computer science that arises from the need to control the development of complex software systems, facilitating understanding, communication, and human coordination of software projects, which allows for to improvement of reliability and quality of software products more efficiently.

A software product or artifact can be defined as *“a stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making”(Pressman, 2015).*

One of the first and most important definitions of **Software Engineering** that is still valid today is the one proposed by Fritz Bauer in which he describes **Software Engineering** as: *“[Software Engineering is] the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines” (McClure, 1968).*

Another relevant definition is the one proposed by the IEEE in which it defines **Software Engineering** as: “*The application of a systematic, disciplined and quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software”* (ISO/IEC/IEEE 24765:2017, 2017).

Considering the definitions of **Software Engineering**, we can conclude that this transcends beyond the programming of a software product; the software engineer encompasses all the management of managing a software project. Going through different stages, based on different processes, methodologies, and standards, allows us to improve the identification of needs, design, quality, efficiency, and organization of software products (Bourque et al., 2014).

Within **Software Engineering**, different process models have been proposed; these aim to give an order and structure to software development, facilitating development for software engineers. One of the most recognized guides today is the SWEBOK (Software Engineering Body of Knowledge) guide (Bourque et al., 2014), which is a document created by the Software Engineering Coordinating Committee and promoted by the IEEE Computer Society. It is defined as a guide to present knowledge in **Software Engineering**.

SWEBOK defines 15 knowledge areas known as (KAs) which are the following: Software Requirements, Software Design, Software Construction, Software Testing, Software Maintenance, Software Configuration Management, Software Engineering Management, **Software Engineering Process**, Software Engineering Models and Methods, Software Quality, Software Engineering Professional Practice, Software Engineering Economics, Computing Foundations, Mathematical Foundations, and Engineering Foundations.

In this Doctoral Thesis, we will focus on the area of knowledge of the **Software Engineering Process**, which consists of activities for management in the creation of software, including the collection of requirements, analysis, design, coding, testing, and maintenance.

SWEBOK defines the **Software Engineering Process** as: *“A Software Engineering Process consists of a set of interrelated activities that transform one or more inputs into outputs while consuming resources to achieve the transformation”* (Bourque et al., 2014). In turn, the **Software Engineering Process** is defined by Fuggetta as: “*It is a set of people, organizational structures, rules, policies, activities and procedures, software components, methodologies and tools used or created specifically to conceptualize, develop, offer a service, innovate and extend a software product*" (Oktaba & Ibargüengoita, 1998).

Figure 3.1 (Breakdown of Topics for the **Software Engineering Process** KA) shows the different phases of the **Software Engineering Process**, according to SWEBOK.

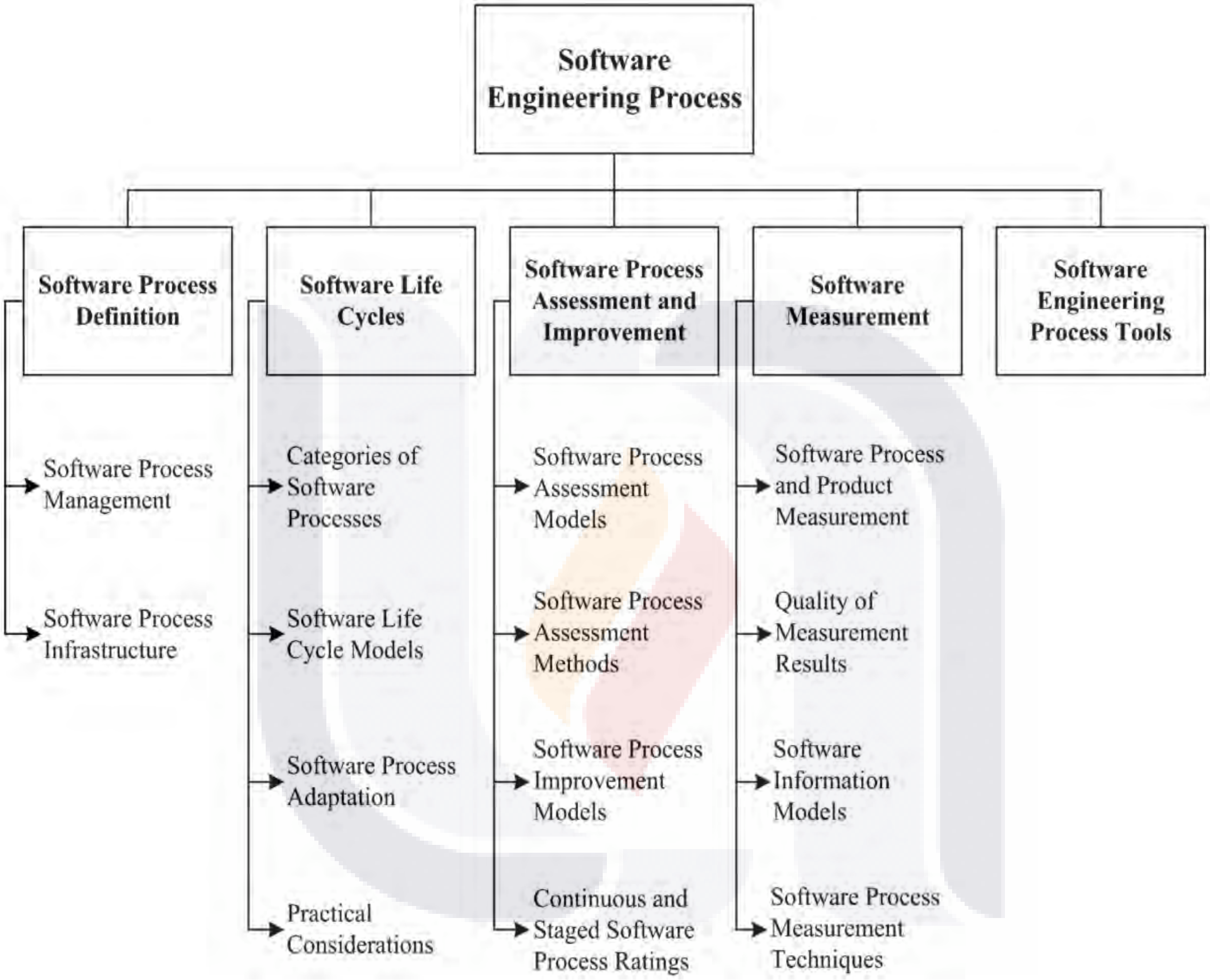

Software Engineering Process
Software Process Definition
Software Process Management
Software Process Infrastructure
Software Life Cycles
Categories of Software Processes
Software Life Cycle Models
Software Process Adaptation
Practical Considerations
Software Process Assessment and Improvement
Software Process Assessment Models
Software Process Assessment Methods
Software Process Improvement Models
Continuous and Staged Software Process Ratings
Software Measurement
Software Process and Product Measurement
Quality of Measurement Results
Software Information Models
Software Process Measurement Techniques
Software Engineering Process Tools

Figure 3.1 Breakdown of Topics for the Software Engineering Process KA (Bourque et al., 2014).

In the specific case of software development, we will focus solely on the **Software Life Cycle**, which is responsible for transforming customer requirements into software products or artifacts, providing the implementation, maintenance, support, and evaluation of a software product.

A clear example of the **Software Engineering Process** is the one developed by Oktaba & Ibargüengoita (1998), where the **Software Engineering Process** is made up of phases, activities, artifacts, roles, and agents. The phases are the highest level of a process, and these, in turn, contain activities. The activities are a fundamental piece since it is the execution of useful work for the generation of output artifacts. In turn, a vital part of the **Software Engineering Processes** are the roles, which allow us to carry out the activities; these can be assigned to a human being or an automated tool (Oktaba & Ibargüengoita, 1998). The **Software Life Cycle** defines the temporal and logical relationships between each phase, activities, roles, and artifacts, as some of the output artifacts may become the inputs to other activities or processes.

The processes and activities of the various parts of software development and **Software Life Cycles** are classified as follows:

- **Primary processes:** These include processes for the development, operation, and maintenance of the software product.
- **Support processes:** They are applied intermittently or continuously throughout the life cycle of the software product.
- **Organizational processes:** These are processes that provide support to software engineering, such as training, process analysis, and infrastructure administration, among others.
- **Cross-project processes:** These processes consider the reuse of processes and contemplate the line of software products of the organization.

Figure 3.2 (Class diagram software process) below shows a UML diagram of the relationship between a **Software Engineering Process**, phases, activities, artifacts, and roles of the Oktaba and Ibargüengoita (1998) model.

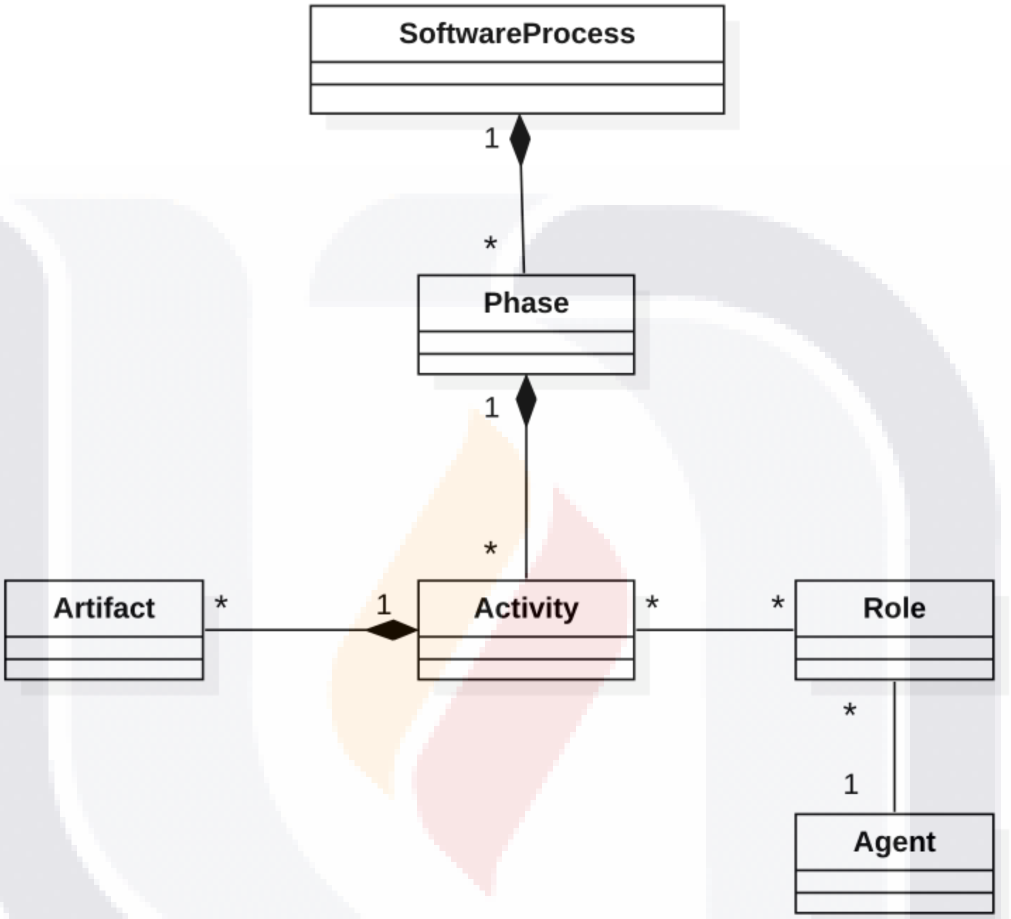


Figure 3.2 Class diagram software process (Oktaba & Ibargüengoita, 1998).

Because software development is so changeable, this has allowed the development of a wide variety of **Software Life Cycles**, some examples of these are the Waterfall model, the Spiral Model, the iterative and incremental model, among others. Agile models have recently been created that involve fewer processes but persist in maintaining the same quality.

These models may contain the following phases, but not all models need to contain these phases or have the same name:

- **Analysis Phase:** Includes activities that allow documentation of software system requirements.

- **Design Phase:** Carries out the design of how the requested requirements will be met and the model that will enable the implementation of software development.

- **Code and Test Phase:** In this phase, the design and the previously developed model are implemented, and it is where the different programming technologies are implemented.

- **Installation Phase:** This is the final phase where the software system is delivered to the customer and implemented in the real environment.

There are additional factors to consider when implementing the **Software Life Cycle**, which include required compliance with standards, directives, and policies, customer requirements, the impact of the software product, maturity, and competencies of the organization.

The life cycle models contemplate that software development must adapt to meet the needs or requirements of the client, clients, and their environments to help determine the necessary adaptations in the phases of the software processes.

The above indicates the importance and relevance of Software Engineering in software development since it is a fundamental part to guarantee costs and software development schedules. Software Engineering aims to help improve the quality and efficiency of software, facilitating development for software engineers and their clients. The great diversity of life cycle models suggests, then, that none of the life cycle models is sufficient to cover all needs and guarantee success in the development of software-intensive systems. This has generated the creation and evolution of different models, allowing software developers to adapt to new technologies, customer demand, and organizational environments. As can be seen in Figure 3.3, where the evolution of software models is shown, this figure indicates a clear trend towards agile development models, which allow us greater flexibility, maintaining the same quality, and with a shorter development time. (Rodríguez et al., 2009).

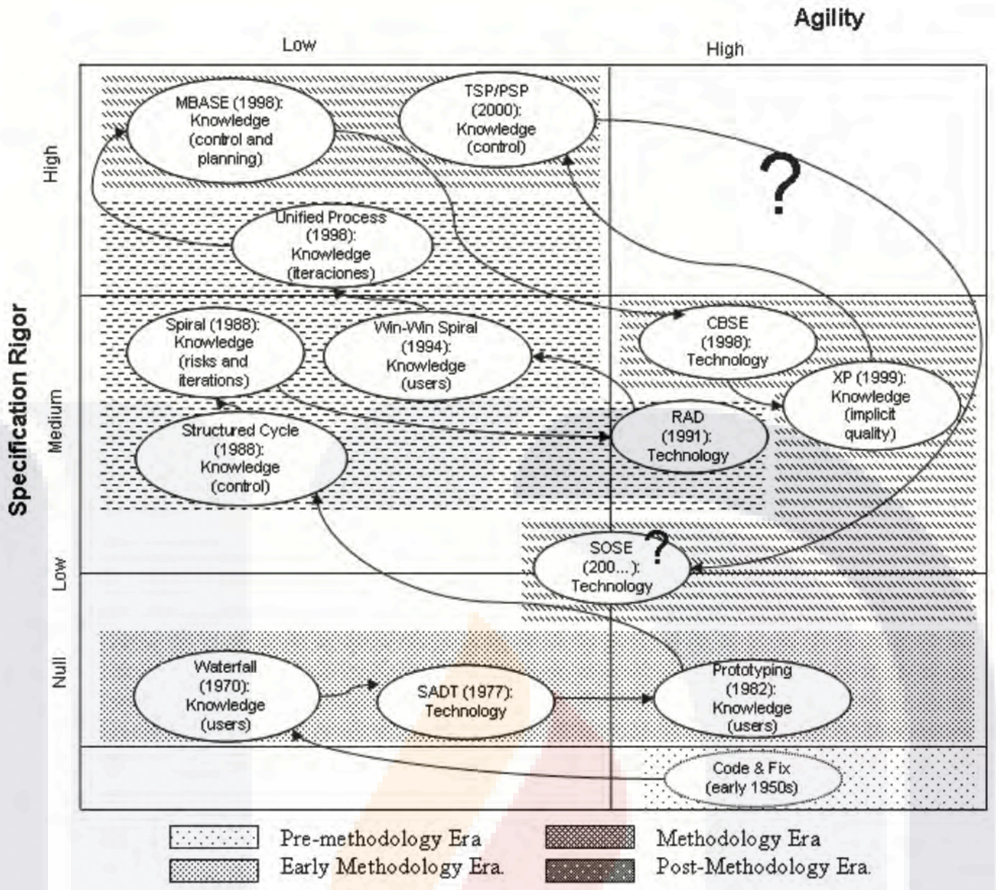

Agility
Low
High
Specification Rigor
High
Medium
Low
Null
MBASE (1998): Knowledge (control and planning)
TSP/PSP (2000): Knowledge (control)
?
Unified Process (1998): Knowledge (iteraciones)
Spiral (1988): Knowledge (risks and iterations)
Win-Win Spiral (1994): Knowledge (users)
CBSE (1998): Technology
XP (1999): Knowledge (implicit quality)
RAD (1991): Technology
Structured Cycle (1986): Knowledge (control)
SOSE (200...): Technology
?
Waterfall (1970): Knowledge (users)
SADT (1977): Technology
Prototyping (1982): Knowledge (users)
Code & Fix (early 1950s)
Pre-methodology Era
Methodology Era
Early Methodology Era.
Post-Methodology Era.

Figure 3.3 Map of PM-SDLC´S evolution (Rodríguez et al., 2009).

3.1.1 ON SOFTWARE ENGINEERING

According to Qumer and Henderson-Sellers, agility can be defined as the ability to accommodate changes (expected or not) in a dynamic environment, be simple, inexpensive, and have quality in a short iteration strategy, applying previous knowledge and generating new knowledge (Qumer et al., 2006).

Agile software development methods arise out of the need for accelerated software product development, as users and organizations demanded more high-quality software products with fast and agile software development processes. This was especially reflected in the case of the volatile Internet software industry and the emerging mobile application environment (Abrahamsson et al., 2003).

However, when the concept of agility for software development emerged, some did not trust its implementation, due to the simplicity and speed of agile approaches, this generated a large amount of literature and debates, since some defended the traditional models of software development, while others saw agility as a new paradigm in software engineering.

Agile development models promised higher customer satisfaction, lower defect rates, faster development times, and a solution to the changing requirements of the organizational environment. While traditional models promised predictability, stability, and high security (Boehm & Turner, 2003).

These characteristics allow agile models to better adapt to small, changing projects that require less stability, where the priority of the clients is the early delivery of the project. In turn, the Traditional Models are better adapted to large projects, where much broader planning is required, with more critical and less changing processes to guarantee the safety and stability of the project.

Both agile and plan-based approaches have a base of project characteristics where each works best and where the other will struggle (Boehm, 2002). The key differences between the two approaches are shown in Table 3.1, Agile and plan-driven methods home grounds.

Table 3.1 Agile and plan-driven method grounds (Boehm & Turner, 2003).

Characteristics	Agile	Plan-Driven
Application		
Primary Goals	Rapid value; responding to change	Predictability, stability, high assurance.
Size	Smaller teams and projects	Larger teams and projects
Enviroment	Turbulent; high change; project-focused	Stable; low-change; project/organization focused
Management		
Customer Relations	Dedicated on-site customers; focused on prioritized increments.	As-needed customer interactions; focused on contract provisions
Plannig and Control	Internalized plans; qualitative control.	Documented plans, quantitative control.

Communications	Tacit interpersonal knowledge.	Explicit documented knowledge.
Technical		
Requirements	Prioritized informal stories and test cases; undergoing unforeseeable change.	Formalized project, capability, interface, quality, foreseeable evolution requirements.
Development	Simple design; short increments; refactoring assumed inexpensive.	Extensive design; longer increments; refactoring assumed expensive.
Test	Executable test cases define requirements, testing.	Documented test plans and procedures.
Personnel		
Customer	Dedicated, knowledgeable, collocated, collaborative, representative, and empowered.	Access to knowledgeable, collaborative, representative, and empowered customers.
Developers	Agile, knowledgeable, collocated, and collaborative.	Plan-oriented; adequate skills; access to external knowledge.

"The handling of unstable requirements, the delivery of software that works in short periods, with high quality and under budget are the main characteristics of agile methods compared to traditional ones" (Jyothi & Rao, 2011).

The traditional approaches rely on a linear or incremental life cycle. These methods are plan-driven and are characterized by a requirement/design/build approach to development (Boehm & Turner, 2004). In these projects, the requirements are specified, and little change is expected; this indicates that the environment is predictable, and planning tools can be used. These approaches are resistant to change and focus on the fulfillment of planning as a measure of success (Wysocki, 2009).

On the other hand, agile methods are created to respond to the dynamic aspects of the environment; they are based on an iterative and adaptable life cycle and were designed to adopt changes in a better way. These methods use the technical knowledge of the work team members rather than the heavy documentation of traditional methods. All the above provide flexibility and adaptability (Wysocki, 2009).

Figure 3.4 Traditional and agile life cycles show the life cycle of both methods, in which we can see the reflected.

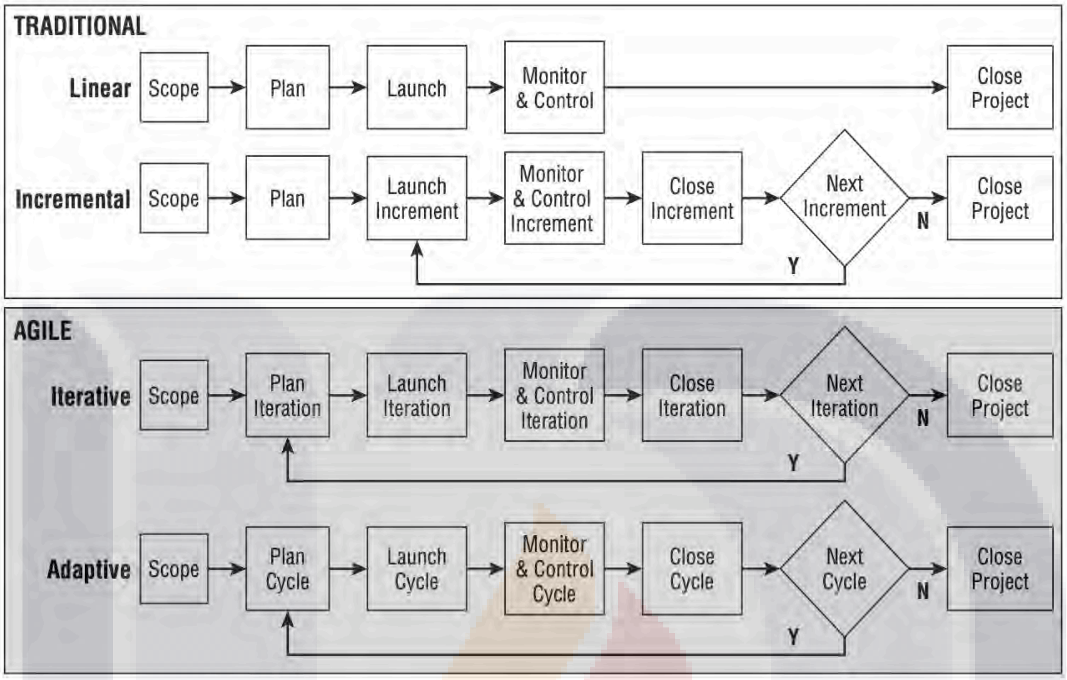


Figure 3.4 Traditional and agile life cycles (Wysocki, 2009).

Both approaches cover the set of conditions in which one approach, or the other, is more likely to be successful. Barry Boehm and Richard Turner determined that 5 critical factors describe the environment of a project and help determine which approach is better in which situations. Table 3.2 The five critical agility and plan-driven factors are described these 5 factors.

Table 3.2 The five critical agility and plan-driven factors (Boehm & Turner, 2003).

Factor	Agility discriminators	Plan-driven discriminators
Size	Well matched to small products and teams; reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams; hard to tailor down to small projects.
Criticality	Untested on safety-critical products; potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products; hard to tailor down efficiently to low-criticality products.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments but present a source of potentially expensive rework for highly stable environments.	Detailed plans and “big design up front” excellent for highly stable environment, but a source of expensive rework for highly dynamic environments.
Personnel	Require continuous presence of a critical mass of scarce Cockburn Level 2 or 3 experts: risky to use nonagile Level 1B people.	Need a critical mass of scarce Cockburn Level 2 and 3 experts during project definition but can work with fewer later in the project—unless the environment is highly dynamic. Can usually accommodate some Level 1B people.
Culture	Thrive in a culture where people feel comfortable and empowered by having many degrees of freedom; thrive on chaos.	Thrive in a culture where people feel comfortable and empowered by having their roles defined by clear policies and procedures; thrive on order.

Boehm and Turner's model is based on an agility-oriented risk assessment and traditional models; the risk associated with an inappropriate choice of the project methodology is reduced by evaluating project factors to determine how well it fits with the methodologies. Agile or traditional methodologies. These 5 factors are graphically shown in the form of a radar in Figure 3.5 Dimensions affecting method selection, which will help us determine which is the best profile for our project or, failing that, a balance can be obtained between both methods.

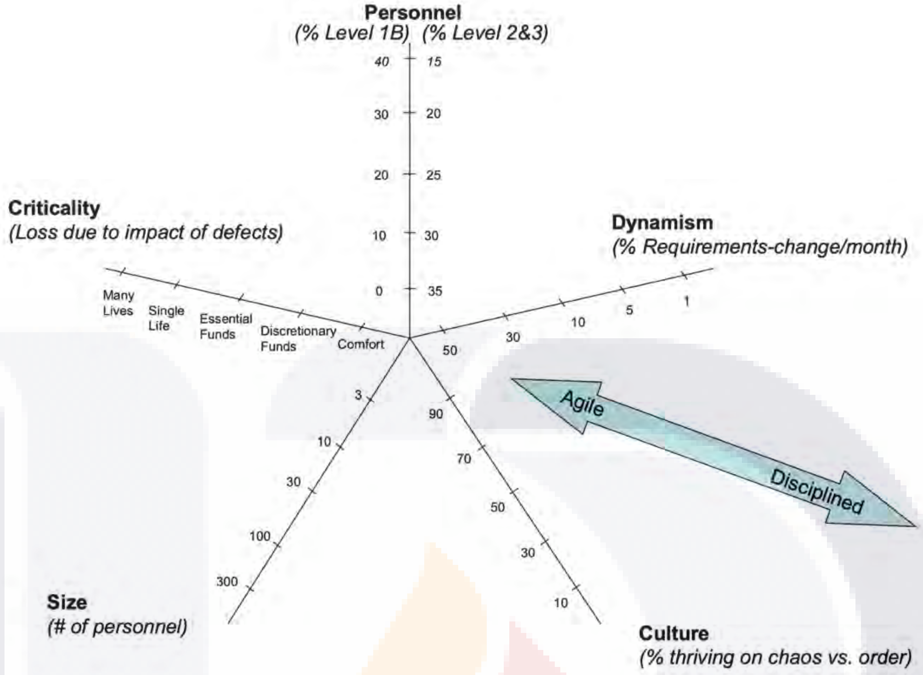


Figure 3.5 Dimensions affecting method selection (Boehm & Turner, 2003).

Of the five axes that we have in Figure 3.5, size, and criticality, the closer to the center of the graph, the better the use of an agile methodology, while if these values are further away from the center of the graph, implement a traditional methodology it is the best choice for the project.

The cultural axis reflects the reality where agile methods are most successful in a culture that "thrives on chaos", while traditional methods are best in an environment where there is a culture that "thrives on order" (Boehm & Turner, 2003).

The axis of dynamism refers to how the project behaves with high and low exchange rates; agile methodologies prefer high exchange rates while traditional methodologies prefer low exchange rates (Boehm & Turner, 2003).

The staff scale refers to the extended skills rating scale of the Cockburn method, where different levels establish the skills of the project developers and, in turn, place a relative framework of the complexity of the project (Boehm & Turner, 2003). This is interpreted so that traditional methods can work well with any skill level, be it high or low, while agile methods require a richer combination of levels with developers at a higher level.

Table 3.3 Levels of Software Method Understanding and Use (After Cockburn) (Boehm & Turner, 2003).

Level	Characteristics
3	Able to revise a method (break its rules) to fit an unprecedented new situation.
2	Able to tailor a method to fit a precedented new situation.
1A	With training, able to perform discretionary method steps (e.g., sizing stories to fit increments, composing patterns, compound refactoring, complex COTS integration). With experience can become Level 2.
1B	With training, able to perform procedural method steps (e.g. coding a simple method, simple refactoring, following coding standards and CM procedures, running tests). With experience can master some Level 1A skills.
-1	May have technical skills, but unable or unwilling to collaborate or follow shared methods.

Table 3.3 Levels of Software Method Understanding and Use (After Cockburn) shows the different levels handled by the method and how to classify which is the correct level for each developer.

It was not until 2001 that agile software development (ASD) was officially presented to the software engineering community through a set of four fundamental values and twelve principles, established in the "Agile Manifesto" (Fowler & Highsmith, 2001). This manifesto establishes 4 main bases for agile software development, which are the following:

- Value people and their interactions more than processes and tools.
- Value functional software over comprehensive documentation.
- Value collaboration with the client more than contractual negotiation.
- Value the response to change more than following a plan.

The creation of these principles gave agile software development the impetus it needed to expand rapidly. The fundamentals and principles of the manifesto allowed the development of methods with a focus towards the real world, where the response to change became a factor of success (Campanelli & Parreiras, 2015). “Since its inception approximately two decades ago, ASD has rapidly become a mainstream software development model in use today” (Stavru, 2014) causing a dramatic impact on current software development, leading to the development of numerous manifestations, methodologies, frameworks, processes, and standards that comply with the fundamentals and principles of the agile manifesto.

Figure 3.6 Evolutionary map of agile software development methods shows the intellectual origins of how these methods began to emerge, in other words, these previous studies have influenced existing agile methods, in the figure agile methods existed before the agile manifesto and how it affected the creation or even the change or adaptation of new and existing methods.

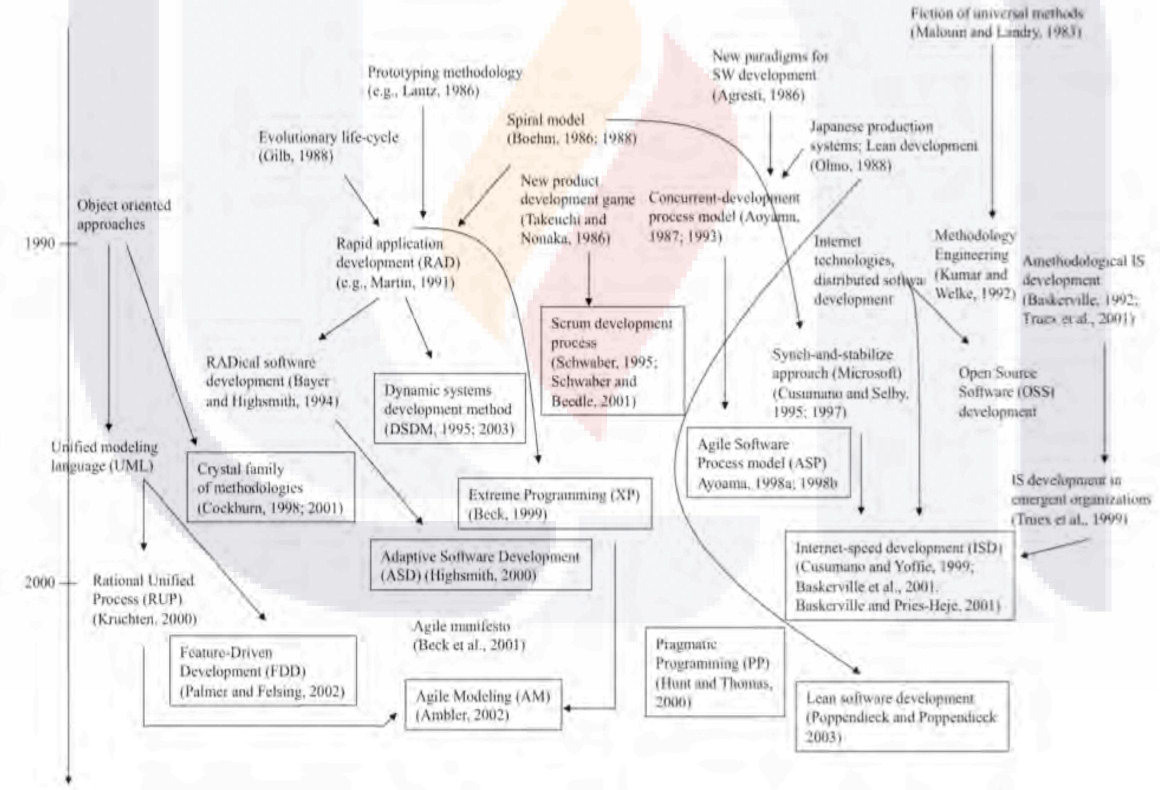

Fiction of universal methods (Malouin and Landry, 1983)
New paradigms for SW development (Agresti, 1986)
Prototyping methodology (e.g., Lantz, 1986)
Spiral model (Boehm, 1986; 1988)
Japanese production systems; Lean development (Olmo, 1988)
Evolutionary life-cycle (Gilb, 1988)
New product development game (Takeuchi and Nonaka, 1986)
Concurrent-development process model (Aoyama, 1987; 1993)
Object oriented approaches
1990
Rapid application development (RAD) (e.g., Martin, 1991)
Internet technologies, distributed softwa development
Methodology Engineering (Kumar and Welke, 1992)
Amethodological IS development (Baskerville, 1992; Truex et al., 2001)
Scrum development process (Schwaber, 1995; Schwaber and Beedle, 2001)
Synch-and-stabilize approach (Microsoft) (Cusumano and Selby, 1995; 1997)
Open Source Software (OSS) development
RADical software development (Bayer and Highsmith, 1994)
Dynamic systems development method (DSDM, 1995; 2003)
Unified modeling language (UML)
Crystal family of methodologies (Cockburn, 1998; 2001)
Agile Software Process model (ASP) Ayoama, 1998a; 1998b
Extreme Programming (XP) (Beck, 1999)
IS development in emergent organizations (Truex et al., 1999)
Adaptive Software Development (ASD) (Highsmith, 2000)
Internet-speed development (ISD) (Cusumano and Yoffie, 1999; Baskerville et al., 2001. Baskerville and Pries-Heje, 2001)
2000
Rational Unified Process (RUP) (Kruchten, 2000)
Agile manifesto (Beck et al., 2001)
Feature-Driven Development (FDD) (Palmer and Felsing, 2002)
Pragmatic Programming (PP) (Hunt and Thomas, 2000)
Agile Modeling (AM) (Ambler, 2002)
Lean software development (Poppendieck and Poppendieck 2003)

Figure 3.6 Evolutionary map of agile software development methods (Abrahamson et al., 2010).

These characteristics of both methodologies allow us to determine that, for the specific case of this thesis, we will use an agile methodology, due to its flexibility, its adaptation to the changing requirements of the environment, and its faster development times. Given that in Data Science projects, these factors are very important, and, at present, there are very few agile methodologies for the development of this type of project. In turn, this type of methodology is better adapted to small organizations, which handle a smaller amount of data.

In the same way, based on the characteristics of each of the methods, we can obtain the main terms for the concepts of traditional methodologies and agile methodologies. Figure 3.7, Main terms traditional methodology, and Figure 3.8, Main terms agile methodology, show us these terms represented in a word cloud.

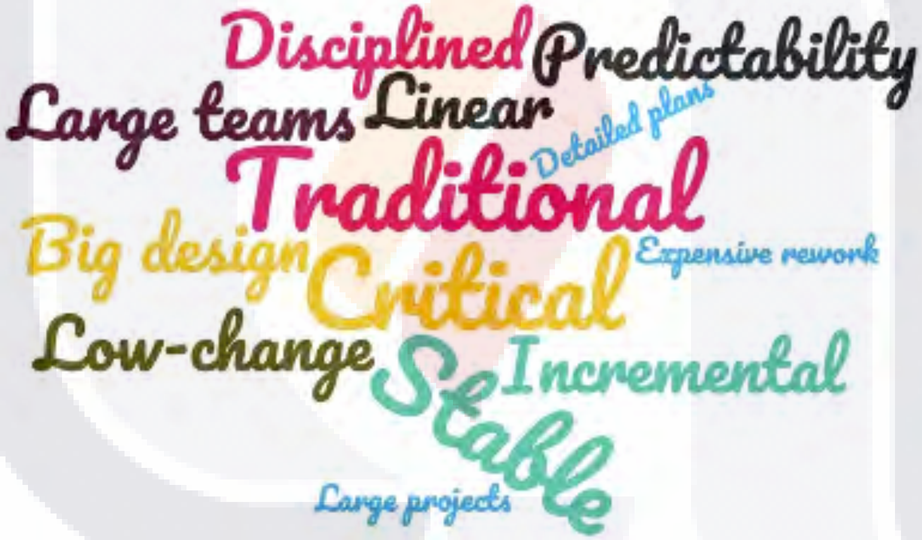


Figure 3.7 Mains terms traditional methodology.

Figure 3.8 Mains terms agile methodology.

3.1.2 ON AGILE DEVELOPMENT PARADIGM

Two of the most widely used **Agile Methodologies** today are Scrum and Extreme Programming (XP); both methodologies are based on the agile manifesto, however, they advocate a significantly different set of agile practices. Scrum is an agile method that primarily focuses on managing project team tasks through practices such as daily meetings, iteration planning, and delivery in short sprints. In contrast, XP is an agile method that advocates practices that focus on quality and software engineering techniques (pair programming, unit tests, etc.) (F. Tripp & Armstrong, 2018).

In this thesis, we will focus on the agile Scrum methodology, which, together with its variants, is the agile methodology most used by 2020 by organizations as shown in Figure 3.9 **Agile Methodologies** Used, being one of the most documented, one of the easiest to implement and adapt in organizations.

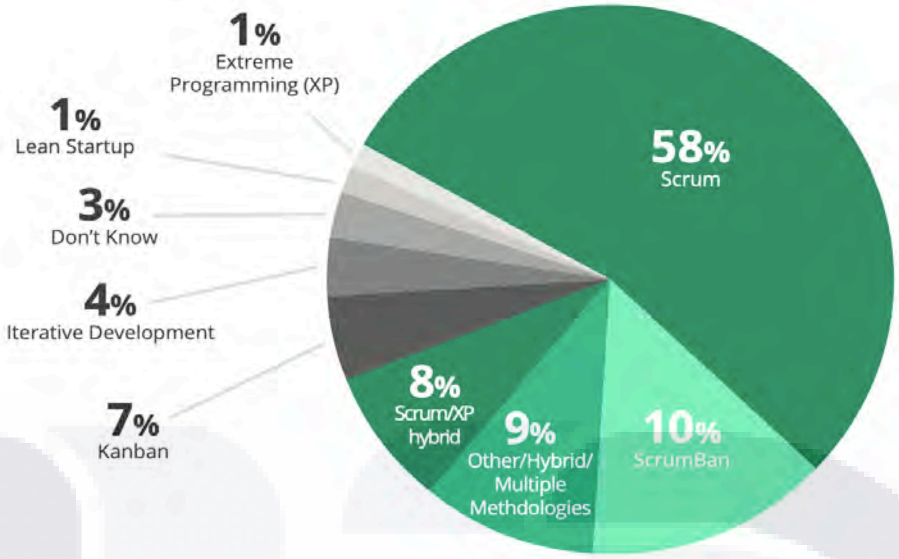


Figure 3.9 Agile Methodologies Used (stateofagile.com, 2020).

Scrum first appeared in 1995, at the Programming, Systems, Languages and Applications conference (OOPSLA). This presentation mainly documents the learning that Ken Schwaber and Jeff Sutherland have obtained over the years applying Scrum. The origin of the term "Scrum" came from the popular sport of Rugby, in which fifteen players from two teams compete against each other. Some of the processes handled by Scrum adopt fundamental Rugby strategies, such as teamwork and constant iteration between team members, which led to an improvement in the iterative and incremental approaches of the time (Sutherland & Schwaber, 2020).

Scrum is defined by the Scrum guide itself as: "A lightweight framework that helps people, teams, and organizations to generate value through adaptive solutions to complex problems" (Sutherland & Schwaber, 2020).

Scrum is a management process that reduces complexity in developing products to meet customer needs. Scrum is based on the experience and collective intelligence of those who make up the team. Instead of giving them detailed instructions for software development, this allows the team to use various processes, techniques, and methods within the same project.

This framework consists of Scrum Teams, their roles, events, artifacts, and associated rules. Each component within the framework serves a specific purpose and is essential to the success of Scrum (Sutherland & Schwaber, 2020).

Scrum is based on an empirical process. Empiricism is based on making decisions based on concrete information obtained from observation that shows the progress of product development, changes in the market, and customer feedback (Sutherland & Schwaber, 2020). Scrum is made up of three fundamental empirical pillars that should be used throughout all software development, and therefore in each of the iterations of the product, these pillars are represented in the following Table 3.4 Empirical pillars of Scrum:

Table 3.4 Empirical pillars of Scrum.

Pillars	Definition
Transparency	*"It establishes that work processes must be visible both to those who do the work and to those who receive it. Artifacts with poor transparency can lead to decisions that decrease project value and increase risk" (Sutherland & Schwaber, 2020).*
Inspection	*"The artifacts and processes that are carried out to achieve the objectives should be inspected frequently for variations or potentially desirable problems" (Sutherland & Schwaber, 2020).*
Adaptation	*"If any of the processes or artifacts deviate from the primary goal, these should be adjusted as soon as possible to minimize further deviation" (Sutherland & Schwaber, 2020).*

As mentioned above, Scrum is made up of roles, events, artifacts, and rules that fulfill a common mission and objective, and this is essential for the success of the project.

Scrum roles

The Scrum Team is the fundamental unit of Scrum, this is a small group of people, generally composed of 10 people or less, since it has been shown that teams communicate better and are more productive, so if a project is required Too large require reorganization into multiple cohesive Scrum teams, each focused on the same product. The Scrum team is responsible for all activities related to the product, from stakeholder collaboration, verification, maintenance, operation, experimentation, research, development, and anything else that may be necessary for development. (Sutherland & Schwaber, 2020).

Scrum Teams must consist of three essential roles to meet the project objectives. Table 3.5 Scrum Roles shows the roles by which the Scrum Team is formed and what is the function of each of them.

Table 3.5 Scrum Roles

Roles	Description
Product Owner	*“He is responsible for maximizing the value of the product resulting from the Scrum Team's work, that is, defining, prioritizing, and communicating the product requirements. He is the only person responsible for managing the Product Backlog, clearly expressing the elements of the Product Backlog, prioritizing user stories to achieve the objectives and missions in the best way” (Sutherland & Schwaber, 2020).*
Scrum Master	*“He is responsible for establishing compliance with the rules and principles of Scrum-based development. The Scrum Master is responsible for the effectiveness of the Scrum Team, helping to eliminate development impediments and improving processes, helping the Scrum Team to improve its practices, within the framework of Scrum. This helps the Product Owner, the Scrum Team and the organization by guiding them on iterations that they have with each other, maximizing the value created between them” (Sutherland & Schwaber, 2020).*
Scrum Team	*“It consists of professionals who carry out the work of delivering a finished product increment that can potentially be put into production at the end of each sprint. The development team follows the user stories established by the Product Owner to meet the delivery of an increment in the established time. The specific skills that developers need are broad and vary by scope of work” (Sutherland & Schwaber, 2020).*

Scrum Events

In Scrum, there are predefined events to create regularity and minimize the need for meetings not defined by Scrum. All events are time-boxed, so they all have a maximum duration. Once the sprint begins, the duration of the events is fixed and cannot be shortened or lengthened.

Each of the Scrum events constitutes a formal opportunity for inspection and adaptation of some aspects. The lack of any of these events results in a reduction in transparency and constitutes a missed opportunity for inspection and adaptation.

Table 3.6 Scrum Events shows the events that Scrum is made up of and the definition of each of these.

Table 3. 6 Scrum Events.

Events	Description
Sprint	*“Defined as the heart of Scrum, it is a block of time of one month or less during which a usable and potentially deployable increment of finished product is created. This event is a container for the rest of the events, this means that the sprint consists of the Sprint Planning, the Daily Scrums, the Sprint Review, and the Sprint Retrospective. Each Sprint has a definition of what will be built, a design and a flexible plan that will guide its construction, the team's work and the resulting product” (Sutherland & Schwaber, 2020).*
Sprint Planning	*“It is all the work that will be done during the Sprint, this plan is created through the collaborative work of the Scrum Team. Planning a Sprint is a maximum of 8 hours in length for a one-month Sprint. This section answers questions such as: What can be delivered in the resulting increase in the Sprint that begins? And how will you get the work necessary to deliver the increase?” (Sutherland & Schwaber, 2020).*
Daily Scrum	*“It is an event that is repeated every day with an approximate duration of 15 minutes, and is aimed at the team's developers, in which the development progress status is communicated and evaluated, improving communication, identifying impediments, promoting streamlining decisions and consequently eliminates the need for other meetings” (Sutherland & Schwaber, 2020).*
Sprint Review	*“This is carried out at the end of each Sprint, to inspect the increase and make corrections for future Sprints. The Scrum*

	Team and stakeholders collaborate on what was done during the Sprint, collaborating to determine the following things that could be done to optimize the value of the product. This is a meeting restricted to a 4-hour block of time for a one-month Sprint" (Sutherland & Schwaber, 2020).
Sprint Retrospective	*"It is an opportunity for the Scrum Team to inspect itself and create a plan for improvements that are addressed during the next Sprint. This takes place after the Sprint Review and before the next Sprint schedule. This is a meeting restricted to a block of three horas for one-month Sprints. Its main function is to create a plan to implement the improvements to which the Scrum Team performs its work" (Sutherland & Schwaber, 2020).*

Scrum artifacts

Scrum artifacts provide transparency and opportunities for inspection and adaptation. Scrum-defined artifacts are specifically defined to promote transparency of information so that everyone has the same understanding of what is taking place through artifacts.

Table 3.7 Scrum Artifacts shows the artifacts that Scrum is made of and the definition of each of them.

Table 3.7 Scrum Artifacts.

Artifacts	Definition
Product Backlog	*"It is a pop-up and ordered list of what is needed for a correct delivery of the product or an improvement of it. In other words, a list of initial requirements for the product being developed. As Scrum is an agile methodology, the Product Backlog may change as products or project requirements evolve. This provides a list of tasks to perform to meet the goal of each of the requirements" (Sutherland & Schwaber, 2020).*
Sprint Backlog	*"It is a plan made by and for the developers, it is a visible and real-time image of the work that the developers plan to do during the Sprint to achieve the goal. This list is written by selecting tasks from the Product Backlog part, organizing enough work for the next sprint, considering the*

	capacity of the Scrum Team and the past performances of the development team" (Sutherland & Schwaber, 2020).
Increment	*"The increment is the sum of all the items in the Product List completed during a Sprint and the value of the increments of all previous Sprints" (Sutherland & Schwaber, 2020).*

The following Figure 3.10 Scrum life cycle represents the Scrum life cycle with all the components that make up Scrum, as described by Ken Schwaber and Jeff Sutherland, who are the creators of the framework.

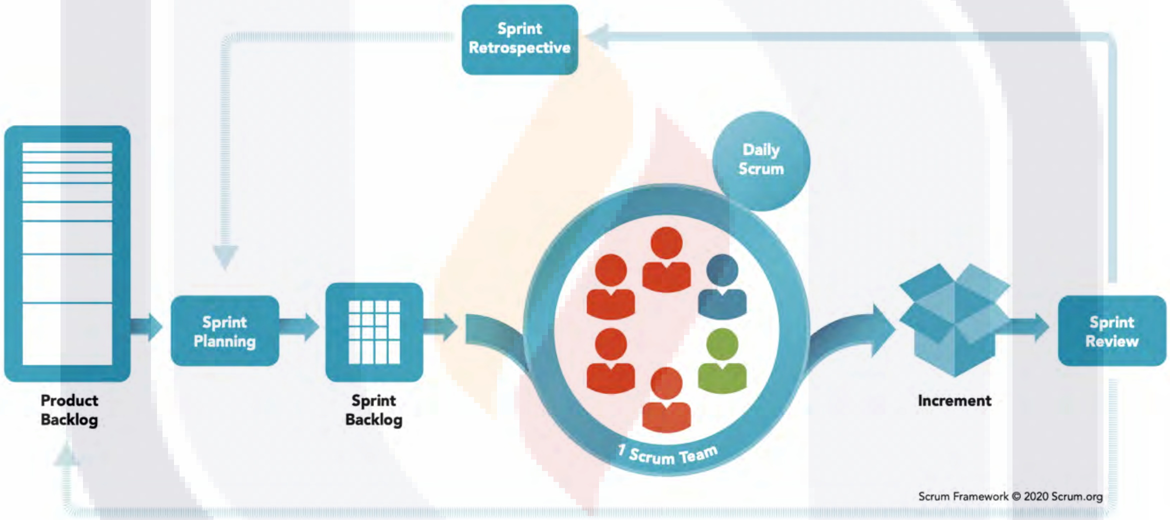


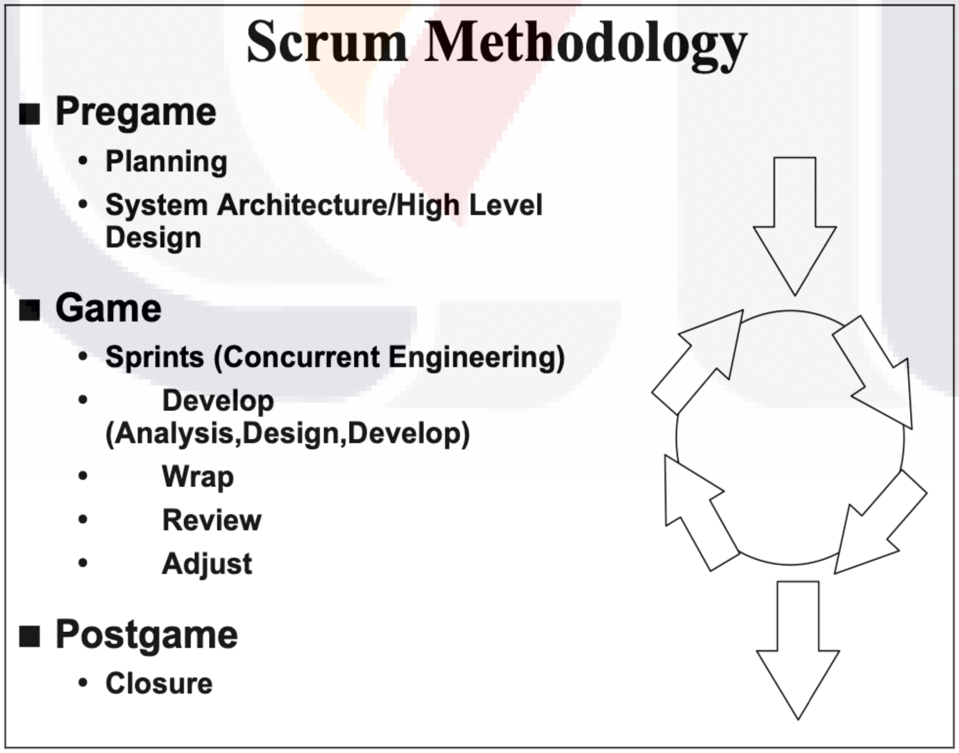
Figure 3.10 Scrum life cycle (Scrum.org, 2020).

Another, more scientific way that the Scrum life cycle can be represented is that proposed by Schwaber in 1997, which consists of three phases: the pregame phase, the game phase, and the postgame phase. These phases encompass all the roles, events, and artifacts that Scrum has, and are seen as a more disciplined way of representing this methodology. The objectives and functions of each of these phases are described in Table 3.8, Scrum Phases.

Table 3.8 Scrum Phases.

Phases	Description
Pre-game	This phase is the one in charge of making a schedule and cost estimate. For the development of a new system, this phase consists of planning and developing the architecture to a high-level design, while if it is an existing system, the analysis is much more limited (Schwaber, 1997).
Game	This phase is the one in charge of making a schedule and cost estimate. For the development of a new system, this phase consists of planning and developing the architecture to a high-level design, while if it is an existing system, the analysis is much more limited (Schwaber, 1997).
Post-game	Finally, the post-game phase prepares for release, including final documentation, pre-release staged testing, and launch (Schwaber, 1997).

The following Figure 3.11 Scrum Methodology shows how the Scrum life cycle was interpreted at the beginning of the methodology; in this figure, we can see the three phases mentioned above and the events that must take place in each of these.



Scrum Methodology

- **Pregame**
  - Planning
  - System Architecture/High Level Design
- **Game**
  - Sprints (Concurrent Engineering)
  - Develop (Analysis,Design,Develop)
  - Wrap
  - Review
  - Adjust
- **Postgame**
  - Closure

Figure 3.11 Scrum Methodology (Schwaber, 1997).

These phases help to establish the Scrum methodology in a more disciplined context, since it shows us how the implementation of the methodology is from the planning, analysis, and design of the architecture, until the closure of the project. This is something that Scrum does not currently contemplate, due to the changes that the methodology has undergone since each work team that uses Scrum can adapt it as it works best for them or best suits their needs. For this thesis, it is essential to show the Scrum methodology as clearly and completely as possible; for this reason, the interpretation by Schwaber is taken as the basis for this work.

We can corroborate what Schwaber mentioned with the XP methodology, which establishes three very similar phases, which consist of different events and activities to be carried out to complete a product launch. Like Scrum, XP is divided into smaller mini projects that result in a functional increase, which is known as a launch. An XP project creates frequent releases (every one to three months) to get early and frequent feedback, gradually building up the sloppy functionality (Dudziak, 1999).

These phases and their XP events are represented in Figure 3.13, Simplified Process Structure XP; this figure shows us a clear similarity with the Scrum methodology and even more with the version proposed by Schwaber. Allowing us to confirm that, seeing Scrum in a more scientific and disciplined way, it is correct to divide this methodology into three phases.

With what is established by Scrum and by XP, we can create Table 3.9 Scrum and XP Phases, which shows us in a clearer way how both methodologies overlap, showing the events and roles that participate in each of the phases established by Schwaber.

Table 3.9 Scrum and XP Phases.

			Roles		
Scrum Phases	**XP**	**Scrum Event**	**Principal**	**Secondary**	**Artifacts**
Pre-game	Exploration	Create Project Vision	Product Owner	Scrum Master	Project Vision Statement
		Develop Epics	Product Owner	Scrum Master, Scrum Team	
		Create User Stories	Product Owner	Scrum Master, Scrum Team	User Stories
	Release Planning	Create Prioritized Product Backlog	Product Owner	Scrum Master, Scrum team	
		Conduct Release Planning	Product Owner	Scrum Master, Scrum team	Product Backlog
Game	Iteration Planning + Implementation + Functional Testing	Create Sprint Backlog (Sprint Planning)	Scrum Team	Product Owner, Scrum Master	Spring Backlog
		Conduct Daily Standup (Daily Scrum)	Scrum Team	Product Owner, Scrum Master	Product & Sprint, Kanban Bord
		Increment Development	Scrum Team	Product Owner, Scrum Master	Increment
		Review Sprint	Scrum Team	Product Owner, Scrum Master	
		Retrospective Sprint	Scrum Team	Product Owner, Scrum Master	Agreed Actionable Improvements
Post-game	Release	Ship Deliverables	Scrum Team	Product Owner, Scrum Master	Final Release

In the same way, a diagram was developed Figure 3.12 Phases and life cycle of Scrum shows us the life cycle of Scrum, divided into the three phases, with the events and activities that are carried out in each phase, in the same way, it shows which are the roles in charge of carrying out each of these events and activities, finally, it is shown how the Scrum and XP methodologies overlap.

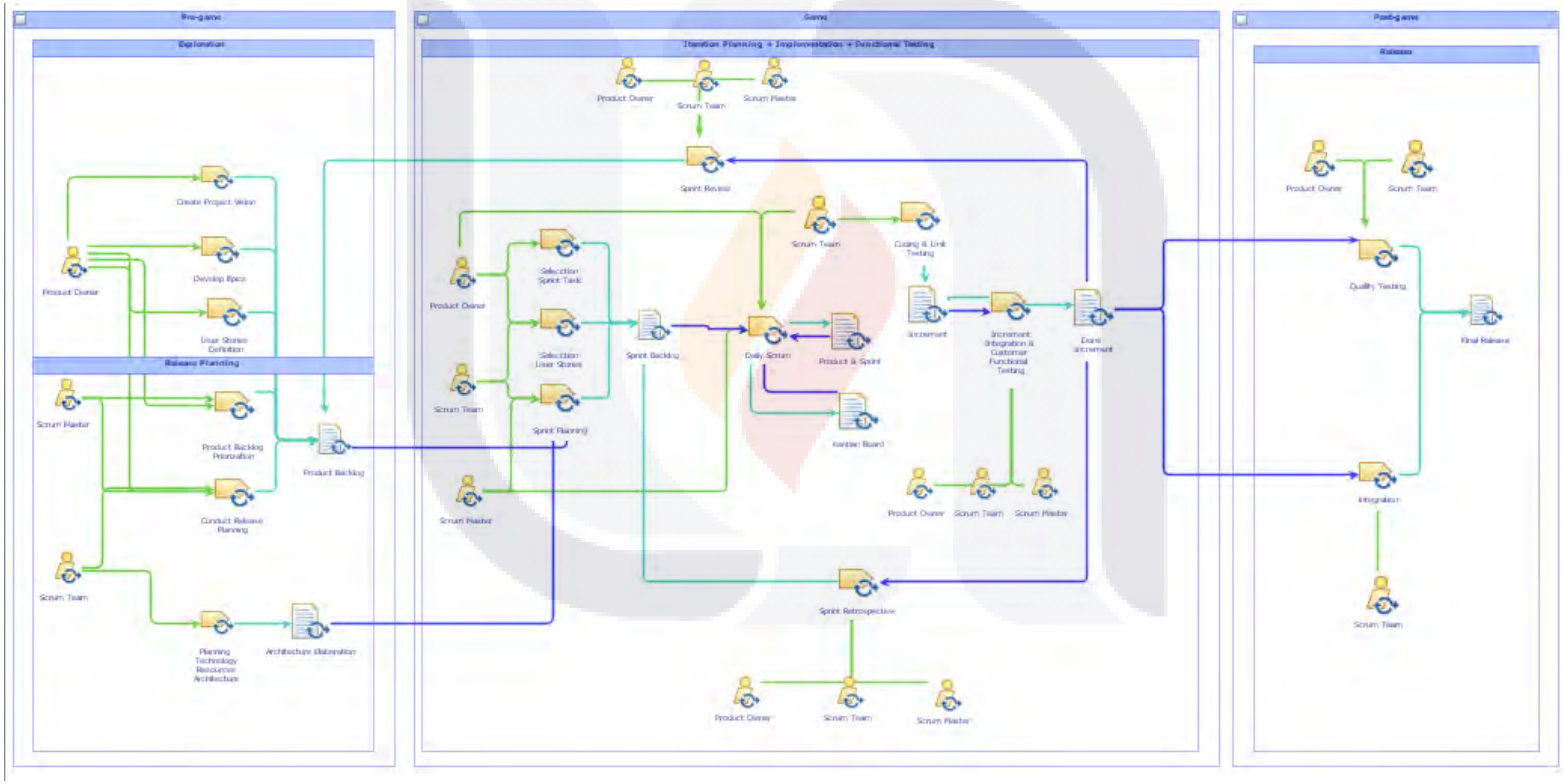


Figure 3.12 Phases and life cycle of Scrum.

With this, we can conclude that Scrum is a framework that can not only be used for software development, since it is a well-defined framework that allows flexibility and adaptability to different projects of different sizes. In turn, it can be concluded that Scrum is more than just Roles, Events, and Artifacts; it is an empirical and incremental framework that uses rules for the development and maintenance of complex products. Its main characteristics are being light, easy to understand, and difficult to master, which allows the strategies to use Scrum to be diverse, and each person or organization can describe how they implement Scrum.

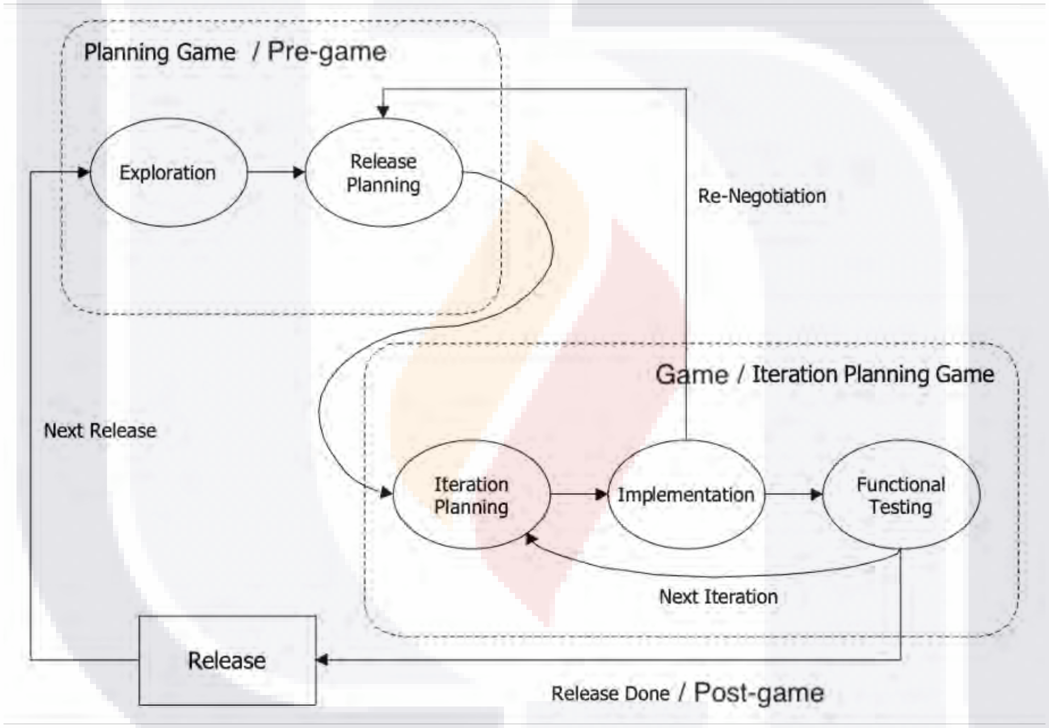


Figure 3.13 Simplified Process Structure XP (Dudziak, 1999).

3.1.3 ON ANALYTICS / DATA SCIENCE SYSTEMS

3.1.3.1 ORIGIN AND CORE DEFINITIONS (ANALYTICS, DATA SCIENCE, DATA SCIENCE / ANALYTICS, BIG DATA IN LARGE BUSINESS, BIG DATA IN SMALL BUSINESS)

In the late 1960s, Analytics began to receive more attention as computers became decision support systems. With the development of Big Data, Data Warehouses, the Cloud, and a variety of software and hardware, Data Analytics has evolved significantly. Data analysis involves the investigation, discovery, and interpretation of patterns within the data.

Due to the growing enthusiasm around Data Science / Data Analytics and its many success stories, more and more organizations find themselves in the need to exploit these technologies, since many companies in the industry offer similar products and use comparable technologies, causing business processes to be among the last points of differentiation (Davenport, 2006). This has generated that organizations that use Data Science / Analytics generate competitive advantages that allow them to better understand the situation of their organizations, the market, and the competition. These companies come to know what their customers want, but they also know what prices those customers will pay, how many items they will buy, and what triggers will make them buy more products. In the same way, they can know when their inventories are running low and can predict problems with demand and supply chains, to achieve low inventory rates and high rates of perfect orders (Davenport, 2006).

Today, due to the enormous amount of data that is being produced at an unprecedented rate, this data is not effectively processed into information, delaying the extraction and production of knowledge. Therefore, our society faces even more challenging problems in transforming data into information and/or knowledge (Song & Zhu, 2016). This led to the creation of two concepts that use this data to generate value in organizations, such as Data Science and Data Analytics.

Since currently making accurate, timely, and better decisions has become essential, but also a matter of survival in the complex and competitive current business context (Demirkan & Delen, 2013).

Analytics

Companies have spent the past forty years or so (Keen & Morton, 1978) building their capabilities for analytics, or the systematic use of statistics and other quantitative methods to enhance decision-making (Davenport & Harris, 2017). The analytics started with a limited number of data sources that came from internal systems and the data that was collected from organizations, for traditional record-keeping and transaction-processing purposes. However, organizations wanted to extract useful information from the data to improve decision making, which was very difficult at the time because data acquisition was expensive and time-consuming (Viswanathan, 2014).

Since today's analytics can require extensive computation (Due to the volume, variety, and speed at which data is created, Big Data), the technical tools and algorithms used for analytics projects take advantage of state-of-the-art, state-of-the-art methods developed in a wide variety of fields including management science, computer science, statistics, data science, and mathematics.

One of the most important definitions is the one mentioned by Davenport & Harris, who defined analytics as ***"By analytics we mean the extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based human analysis, ability to drive decisions and actions".*** In Table 3.10 (Definitions of Analytics), you can see the most important definitions with some of the most important and recognized authors in the field of Data Analytics.

In analytics, we can indicate that data analysis projects can be divided into several phases. The data is evaluated, selected, cleaned, filtered, visualized, and analyzed, to finally be interpreted and evaluated (Runkler, 2020). Figure 3.14 shows us the phases and processes that are carried out in each of these phases to complete the Data Analytics process.

Table 3. 10 Definitions of Analytics.

Autor	Definition
Davenport & Harris, 2007	***“By analytics we mean the extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based human analysis. ability to drive decisions and actions”.***
Denle & Ram, 2018	Analytics (or perhaps more appropriately, data analytics) can simply be defined as “the discovery of meaningful patterns – new and novel information and knowledge – in data.
Delen & Ram, 2018	***“Analytics (or perhaps more appropriately, data analytics) can simply be defined as “the discovery of meaningful patterns – new and novel information and knowledge – in data.” Since we are living in an era of big data, the analytics definitions are mostly focused on that – data that are being created in large volumes, varieties with a high velocity”.***
Chang et al., 2019	***“Is the systematic processing and manipulation of data to uncover patterns, relationships between data, historical trends and attempts at predictions of future behaviors and events”.***
Runkler, 2020	***“Data analytics is defined as the application of computer systems to the analysis of large data sets for the support of decisions. Data analytics is a very interdisciplinary field that has adopted aspects from many other scientific disciplines such as statistics, machine learning, pattern recognition, system theory, operations research, or artificial intelligence”.***
Informs, 2021	***"The scientific process of transforming data into knowledge to make better decisions."***
Stobierski,2021	***“Data analytics refers to the process and practice of analyzing data to answer questions, extract insights, and identify trends. This is done using an array of tools, techniques, and frameworks that vary depending on the type of analysis being conducted”.***

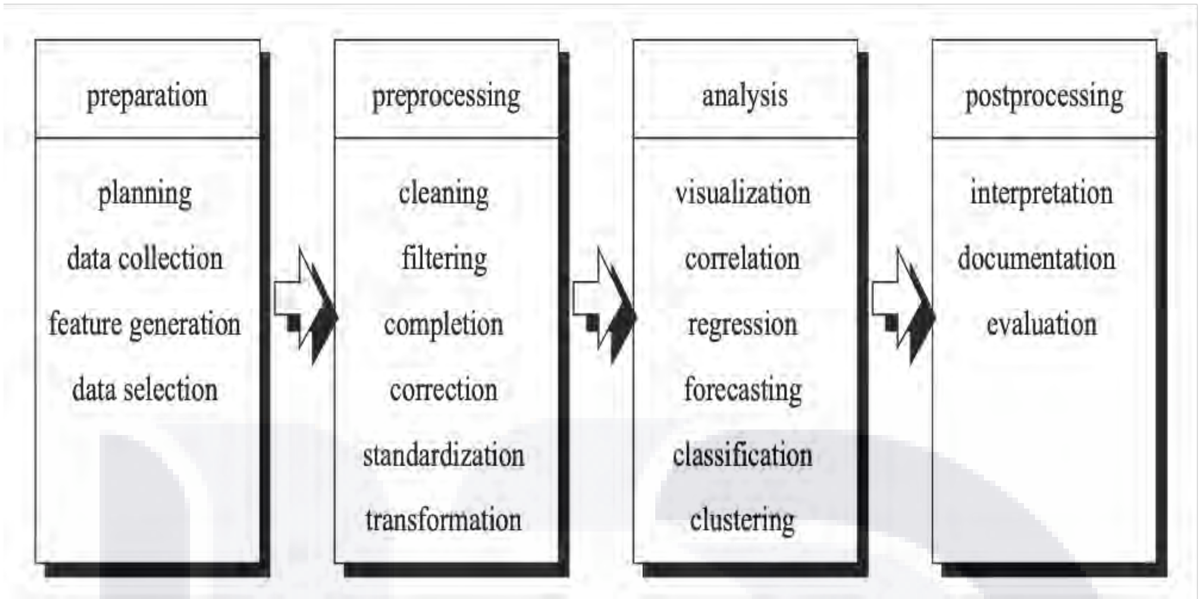


Figure 3.14 Phases of data analysis projects (Runkler, 2020).

In the 1970s, decision support systems (DSS) were the first systems to support decision making. Over time, decision support applications became popular, such as executive information systems, online analytical processing, among others. Then, in the 1990s, Howard Dresner, a Gartner analyst, popularized the term Business Intelligence. A typical definition is that ***"BI is a broad category of applications, technologies, and processes for collecting, storing, accessing, and analyzing data to help business users make better decisions"*** (Watson, 2009).

With this definition, BI can be seen as an umbrella term for all applications that support decision making, and this is how it is interpreted in industry and, increasingly, in academia. BI evolved from DSS, and one could argue that analytics evolved from BI (at least in terms of terminology). BI can also be viewed as ***"data in"*** (to a data mart or warehouse) and ***"data out"*** (analyzing the data that is stored). A second interpretation of analytics is that it is the "pull data" part of BI. The third interpretation is that analytics is the use of ***"rocket science"*** algorithms (e.g., machine learning, neural networks) to analyze data. The progression from DSS to BI and analytics is shown in Figure 3.15 (Watson, 2014).

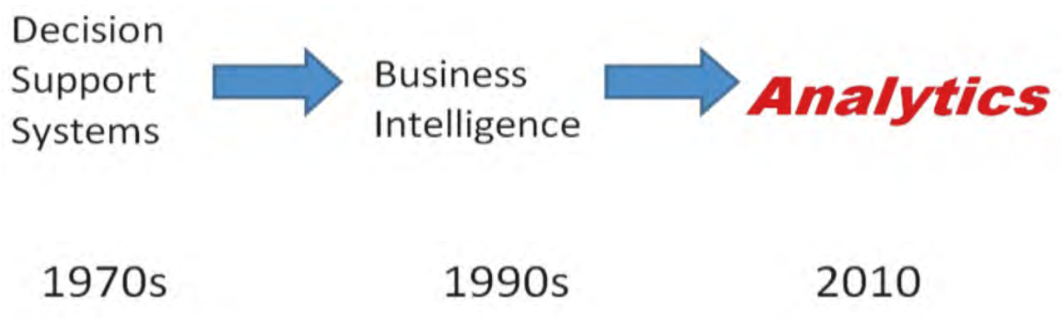


Figure 3.15 DSS & BI & Analytics. (Watson, 2014).

Within Analytics, there are different types of analytics, where it is useful to distinguish between three types of analytics because the differences have implications for the technologies and architectures used for Big Data analytics (Watson, 2014).

Table 3.11 Analysis Types (Watson, 2014).

Type	Definition
Descriptive analytics	They are reports like dashboards, data visualization, they have been widely used for some time and are the core applications of traditional BI. Descriptive analyzes look back and reveal what happened. However, one tendency is to include predictive analytics findings, such as future sales forecasts, in dashboards.
Predictive analytics	Suggest about what will happen in the future. Methods and algorithms for predictive analytics, such as regression analysis, machine learning, and neural networks, have been around for some time. The ability to analyze new data sources, Big Data, creates additional opportunities for insight and is especially important for companies with large amounts of data. Golden Path analysis is an exciting new technique for predictive or analytics. It involves analyzing large amounts of behavioral data (that is, data associated with people's activities or actions) to identify patterns of events or activities that predict customer actions.
Prescriptive analytics	Predict what will happen, prescriptive analysis suggests what to do. Prescriptive analytics can identify optimal solutions, often for scarce resource allocation. It has also been researched in academia for a long time, but now being used more in revenue management it is becoming more common for organizations that have "perishable" assets such as rental cars, hotel rooms, and airplane seats. For example, Harrah's Entertainment, a leader in the use of analytics, has been using revenue management for hotel room rates for many years.

Data Science

The birth of Data Science as a discipline is relatively recent and arose from the need to control the massive volume of data that was emerging with the arrival of Big Data and the evolution of analytics, The data had to be quickly converted into information for analysis. Organizations began to focus more on prescriptive and predictive analytics using machine learning, as well as rapid analytics through visualization. (Larson & Chang, 2016). Big Data is a related field, often thought of as a subset of data science, in the sense that data science applies to large and small data sets and covers the comprehensive process of collecting, analyzing, and communicating data. Analysis results.

Data Science is a body of principles and techniques for applying data analytic methods to data at scale, including volume, velocity, and variety, to accelerate the investigation of phenomena represented by the data, by acquiring data, preparing, and integrating it, possibly integrated with existing data, to discover correlations in the data, with measures of likelihood and within error bounds. Results are interpreted concerning some predefined (theoretical, deductive, top-down) or emergent (fact-based, inductive, bottom-up) specification of the properties of the phenomena being investigated.

Likely, the first appearance of **"Data Science"** as a term in the literature was in the preface to Naur's book ***"Concise Survey of Computer Methods"*** (Naur, 1974) in 1974. In that preface, data science was defined as "*the science of data processing, once established, while the relationship of the data with what they represent is delegated to other fields and sciences."* Another term according to Dhar, data science is defined as ***"data science is the study of the generalizable extraction of knowledge from data"*** (Dhar, 2013). Other definitions that we can find of Data Science are those shown in Table 3.12 (Definitions of Data Science), which are some of the most complete definitions and of the best-known authors in the field of Data Science.

Table 3.12 Definitions of Data Science.

Autor	Definition
Turkey, 1962	***"Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data."***
C. Hayashi, 1998	Data science (DS, by its name in English Data Science) is a concept that not only synthesizes and unifies the field of statistics, data analysis and its related methods, but also seeks to understand the results obtained.
Provost & Fawcett, 2013	***“A set of fundamental principles that support and guide the principled extraction of information and knowledge from data”.***
O'Neil & Schutt, 2013	***“Data science is an emerging discipline that integrates concepts in a variety of fields, including computer science, information systems, software engineering, and statistics”.***
Das et al., 2015	***“Data science is an emerging discipline that combines expertise in a variety of domains, including software development, data management, and statistics. Data science projects generally have the goal of identifying correlations and causal relationships, classifying and predicting events, identifying patterns and anomalies, and inferring probabilities, interests, and feelings”.***
Brodie, 2015	***“Data Science is concerned with analyzing Big Data to extract correlations with estimates of likelihood and error”.***
Bichler et al., 2016	Data science is an interdisciplinary field aiming to turn data into real value. Data may be structured or unstructured, big or small, static or streaming. Value may be provided in the form of predictions, automated decisions, models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results considering ethical, social, legal, and business aspects.
Chang et al., 2019	***“Data science is the methodology for the synthesis of useful knowledge directly from data through a process of discovery or of hypothesis formulation and hypothesis testing”.***

With the previous definitions, it is clear to us that Data Science seeks to extract large amounts of data using the disciplines of mathematics, statistics, and computer science, which will help us identify patterns, increase efficiency, predict behaviors, recognize new market opportunities, reduce costs, generate competitive advantages, among others. Figure 3.16 (Three pillars of data science) shows three pillars of Data Science (Data, Technologies, and People), where Data refers to areas of domains such as relational data, non-relational data, and even data collected from the Internet of Things. Technologies that include concepts such as Data Mining, Deep Learning, Machine Learning, Artificial Intelligence, among others. People who refer to the required personnel, such as computer scientists, statisticians, data scientists, and business analysts (Song & Zhu, 2016).

Among the three pillars, the most important is people. We can buy more computers, storage, and tools to efficiently process Big Data, but human capacity does not scale; Educating people, called data scientists, is key to addressing the challenges of the era of big data (Song & Zhu, 2016).

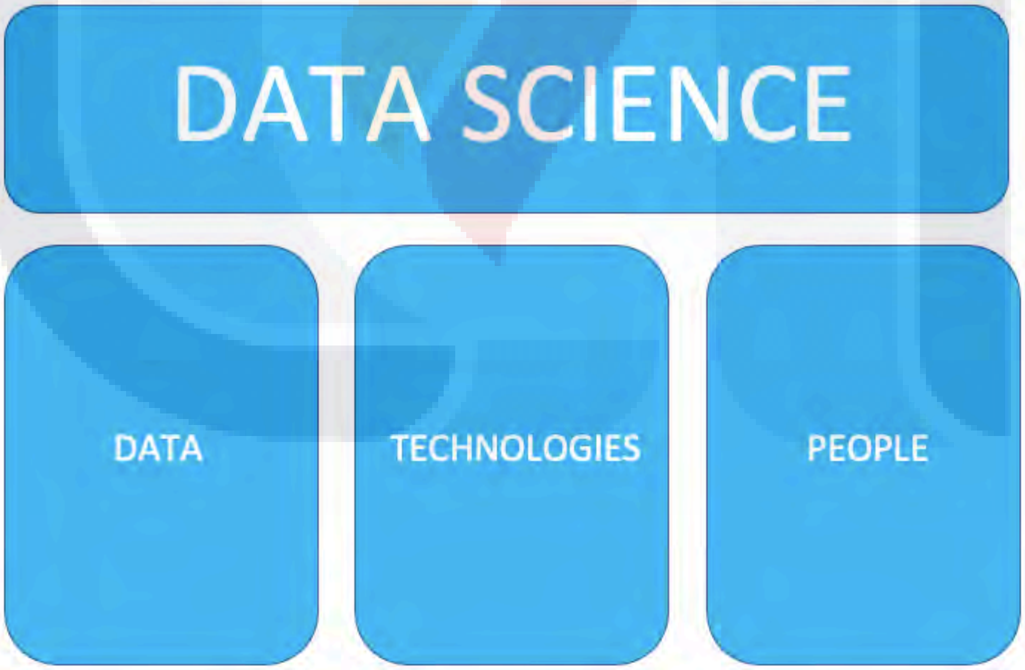


Figure 3.16 Three pillars of data science (Song & Zhu, 2016).

Data Science / Analytics

Considering the above, we can infer that there are few differences between Data Science and Analytics, since both focus on the transformation of data for knowledge, prediction, visual reports, and improvement in decision making, among others. In addition to using the same fundamentals, mathematics, statistics, computer science, and business as its main branches. And we can define Data Science and Analytics as ***"An interdisciplinary field whose objective is to convert data into value, where data is transformed into knowledge to make better decisions, using statistical and quantitative analysis".***

Today, practitioners and academics often use the term "data analysis" or "data science" interchangeably with the older term knowledge discovery (Chen et al., 2012).

Data science and analytics projects generally aim to identify correlations and causal relationships, classify and predict events, identify patterns and anomalies, and infer probabilities, interests, and sentiments.

This is done using a variety of tools, techniques, and frameworks that vary depending on the type of analysis being performed.

This can be seen reflected in Figure 3.17 (Fundamentals of data science and analysis), where it shows us how the three branches come together so that data science and analysis can exist. That is why we will unify both terms in this thesis, referring to them as Data Science / Analytics.

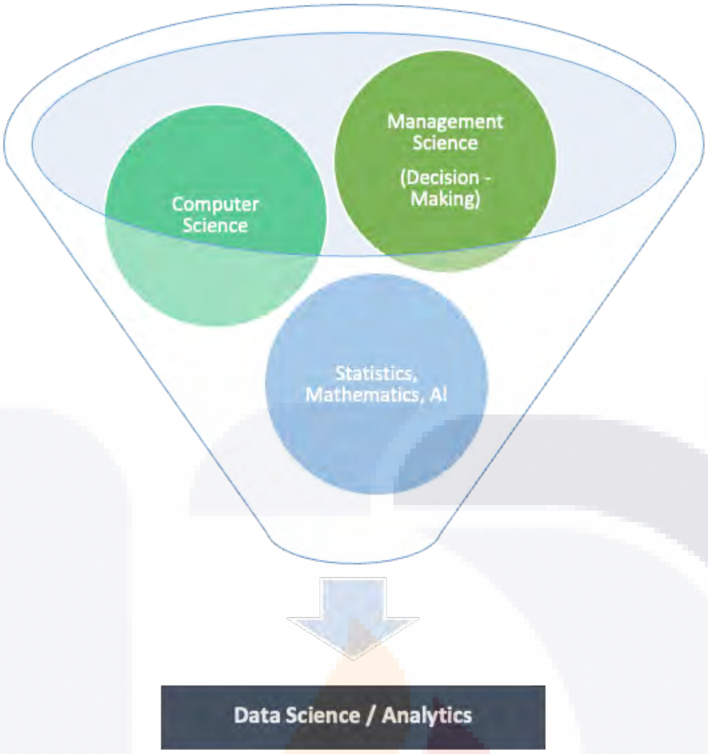


Figure 3.17 Foundations of Data Science and Analytics.

Big Data in Large Business

NASA researchers Michael Cox and David Ellsworth (1997; p. 236) were the first to refer to the term ***'Big Data'*** when they report, ***"Visualization poses an interesting challenge for computer systems of computer systems: the data sets are often quite large, straining the capacity of main memory, local disk, and even remote disk, local disk, and even remote disk. We call this the big data problem".*** They emphasize that even the supercomputers of that time could not process that amount of information, which is why, in the article, they mention a process for handling ***'Big Data'***. Thus, implying that this problem of having information that exceeds the capabilities of computers to handle it traditionally is not a recent problem.

From an evolutionary perspective, Big Data is not new. One of the main reasons for creating data warehouses in the 1990s was to store large amounts of data

(Gandomi & Haider, 2015). Figure 3.18 (Frequency distribution of documents containing the term ***'Big Data'*** in ProQuest Research Library) shows that the term Big Data became mainstream as recently as 2011.

A. Gandomi, M. Haider / International Journal of Information Management 35 (2015) 137–144

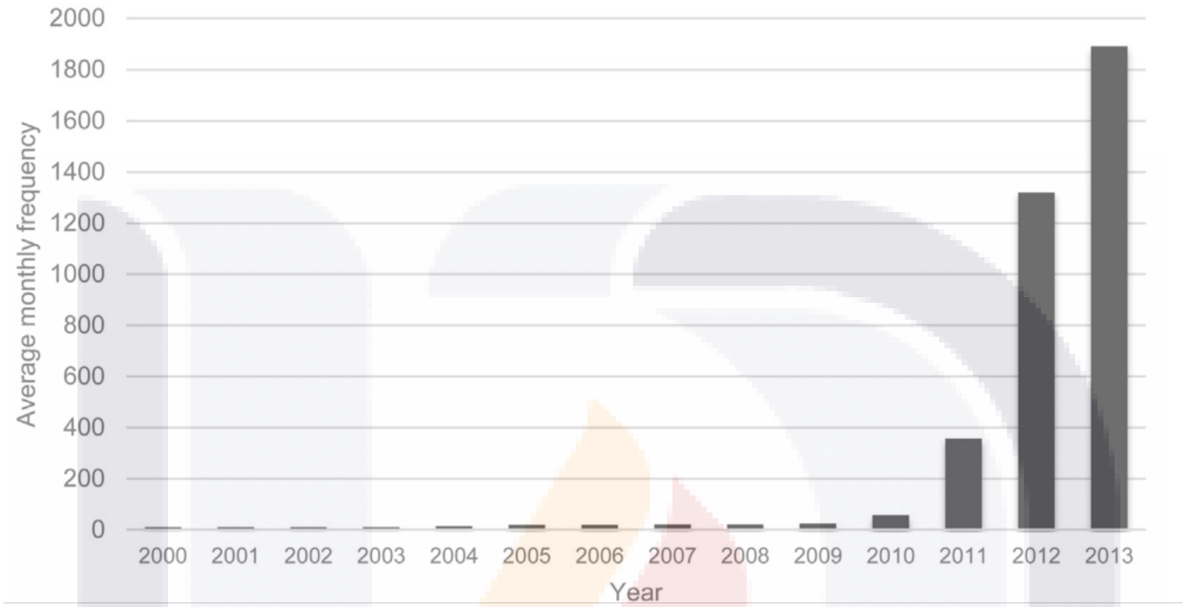


Figure 3.18 Frequency distribution of documents containing the term "Big Data" in ProQuest Research Library (Gandomi & Haider, 2015).

Big Data describes a holistic information management strategy that is formed or constituted by a diversity of new types of data, the management of such data alongside traditional data. Although many of the techniques for processing and analyzing these types of data have been around for some time, it has been the massive generation of data and lower-cost computational models that have fostered wider adoption (Heller & Röthlisberger, 2015).

The different ways to extract information from Big Data can be divided into three types that are:

- **Traditional enterprise data**: Transactional ERP data, including customer information from CRM systems, general ledger data, and web store transactions.
- **Machine-generated /sensor data**: Includes manufacturing sensors, Call Detail Records, equipment logs, weblogs, trading systems data, and smart meters.

- **Social data:** Social media platforms like Facebook, micro-blogging sites like Twitter, include customer feedback streams.

The data, among others, is commonly referred to as ***"Big Data"*** because of its volume, the speed with which it arrives, and the variety of forms it takes. Big Data is creating a new generation of decision support data management because value is created only when data is analyzed and acted upon. One perspective is that big data is more and different types of data than traditional relational database management systems can easily handle. Currently, many data sources are not being leveraged as they should or could be. For example, customer emails, customer service chat, and social media commentary can be processed to better understand customer sentiments. Web browsing data can capture every mouse movement to better understand customer buying behaviors. Radio frequency identification (RFID) tags can be placed on each piece of merchandise to assess the condition and location of each item.

However, considering the emerging nature of Big Data, there are several definitions which are shown in Table 3.13 (Definitions of Big Data), and Figure 3.19 shows the projected growth of Big Data (Watson, 2014).

Table 3.13 Definitions of Big Data.

Autor	Definition
Michael Cox & David Ellsworth, 1997	***“Data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of big data. When data sets do not fit in main memory (in core), or when they do not fit even on local disk, the most common solution is to acquire more resources”.***
Jacobs, 2009	***“Data that is too large to be placed in a relational database and analyzed with the help of a desktop statistics/visualization package— data, perhaps, whose analysis requires massively parallel software running on tens, hundreds, or even thousands of servers”.***
Russom, 2011	***“Description of the voluminous amount of unstructured and semi-structured data a company creates or data that would take too much time and cost too much money to load into a relational database for analysis”.***
Chen et al., 2012	More recently big data and big data analytics have been used to describe the data sets and analytical techniques in applications that are so large (from terabytes to exabytes) and complex (from sensor to social media data) that they require advanced and unique data storage, management, analysis, and visualization technologies.
Davenport et al., 2012	***“Data from everything including click stream data from the Web to genomic and proteomic data from biological research and medicine”.***
Mills et al., 2012	***“Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes”.***
Davoudian & Liu, 2020	***“They are an emerging class of scalable software technologies by which massive amounts of heterogeneous data are collected from multiple sources, managed, analyzed (in batch, in the form of a stream, or hybrid), and served to end users and applications. external. Such systems pose specific challenges in all phases of the software development life cycle and can become very complex due to the evolution of data, technologies, and target value over time”.***

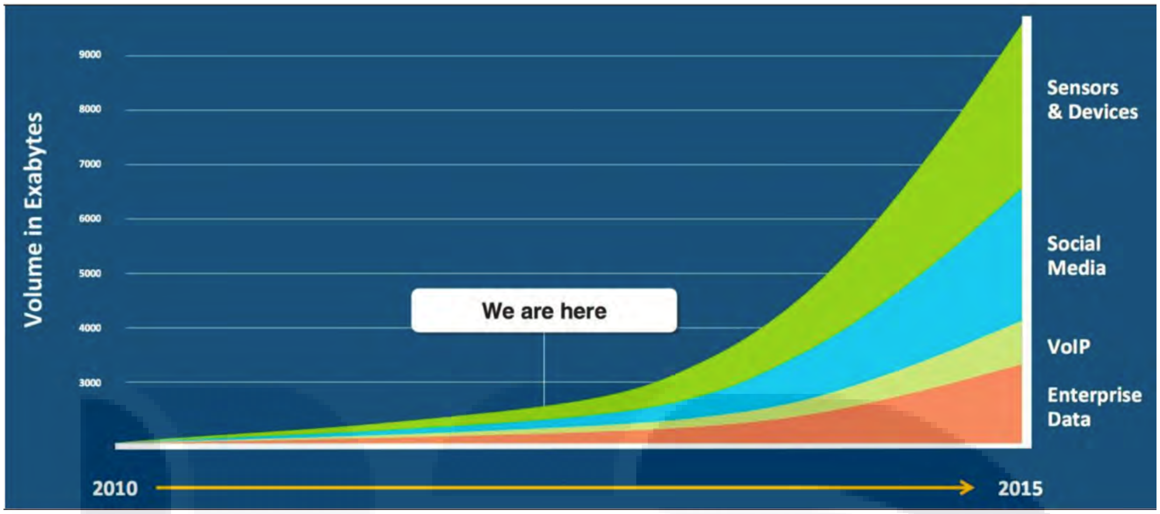


Figure 3.19 The Exponential Growth of Big Data (Palfreyman, 2013).

The current hype can be attributed to the promotional initiatives of certain leading technology companies that invested in building the analytics market niche. Some academics and professionals have considered "Big Data" as data that comes from various channels, including sensors, satellites, social media feeds, photos, videos, and cell phone and GPS signals (Rich, 2012).

Business intelligence and analytics (BI&A) and the related field of big data analytics have become increasingly important to both the academic and business communities over the past few decades. Through BI&A 1.0 initiatives, businesses and organizations across industries began to gain critical insights from structured data collected through various enterprise systems and analyzed by commercial relational database management systems. In recent years, web intelligence, web analytics, web 2.0, and the ability to mine unstructured user-generated content have ushered in a new and exciting era of BI&A 2.0 research, leading to unprecedented intelligence on consumer sentiment, customer needs, and recognizing new business opportunities. Now, in this era of Big Data, even if BI&A 2.0 is still maturing, we stand on the brink of BI&A 3.0, with all the uncertainty that comes with new and potentially revolutionary technologies. (Chen et al., 2012) Figure 3.20 (BI&A Overview: Evolution, Applications, and Emerging Research) shows the evolution of BI&A, applications, and emerging analytics research opportunities.

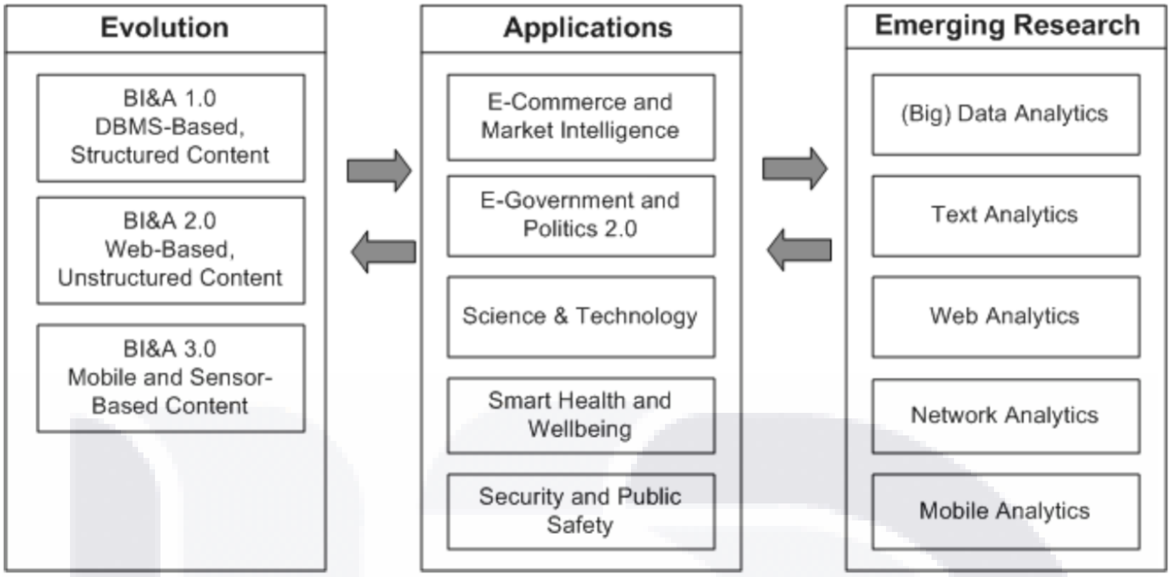


Figure 3.20 BI&A Overview: Evolution, Applications, and Emerging Research (Chen et al., 2012).

The opportunities associated with data and analytics in different organizations have helped generate significant interest in BI&A, which is often referred to as the techniques, technologies, systems, practices, methodologies, and applications that analyze critical business data to help a company better understand its business and marketplace and make timely business decisions. In addition to the underlying data processing and analytical technologies, BI&A includes business-centric practices and methodologies that can be applied to various high-impact applications such as e-commerce, market intelligence, e-government, healthcare, and security (Chen et al., 2012).

One of the most well-known characteristics of macro data is undoubtedly the volume of data that can be stored; However, this is not the only characteristic of Big Data and macro data. For example, Laney (2001) suggested that volume, variety, and speed (or the three Vs) are the three dimensions of data management challenges. The Three Vs have emerged as a common framework to describe Big Data (Chen et al., 2012; Kwon et al., 2014).

However, with time, new characteristics of Big Data were discovered: the 5V: Volume, Variety, Velocity, Veracity, and Value. Table 3.14 (Big Data Features) describes each of these Big Data features, the three initially mentioned, as well as the recently discovered features.

Table 3.14 Big Data Features.

Attributes	Definition
Volume	The most recognized feature of Big Data is the presence of large data sets that allow us to analyze to extract valuable information (Chang et al., 2019). Organizations currently must learn to manage the large volume of data through new processes. Volume in Big Data can be defined as: ***"Large volume of data that either consume huge storage or consist of large number of records" (Russom, 2011).***
Variety	The word ***'Variety'*** denotes the fact that Big Data originates from numerous sources that can be structured, semi-structured, or unstructured (Schroeck et al., 2012). This is another critical attribute of Big Data as data is generated from a wide variety of sources and formats (Russom, 2011).
Velocity	Speed refers to the frequency of data generation and / or the frequency of data delivery (Russom, 2011). The high speed of Big Data can allow analysts to make better decisions, generating commercial value (Gentile, 2012). To utilize the high speed of data, many companies now use sophisticated systems to capture, store, and analyze data to make real-time decisions and retain their competitive advantages (Akter et al., 2016).
Veracity	High data quality is an important Big Data requirement for better predictability in the trading environment (Schroeck et al., 2012). Therefore, verification is necessary to generate authentic and relevant data, and to have the ability to filter incorrect data (Beulke, 2011). This tells us that data verification is essential to the data management process since erroneous data will hinder decision-making or guide analysts down the wrong path. Similarly, incorrect data would have little relevance to add commercial value (Akter et al., 2016).
Value	It is the added value obtained by organizations; value is created only when data is analyzed and acted upon correctly. To do this, we must identify all the data that will help us in the best way to generate value. This can be interpreted as: The extent to which big data generates economically worthy insights and or benefits through extraction and transformation.

Big Data in Small Business (Small Data)

Until recently, the term Small Data was somewhat unknown, but thanks to the rapid growth and impact of Big Data, the term Small Data was used, that is, studies supported by data produced in a strictly controlled way using sampling techniques that limit its scope, temporality, size, variety and that they tried to capture and define its levels of error, bias, uncertainty, and origin (Miller, 2010). Unlike Big Data, it is characterized by its generally limited volume, controlled data speed, limited data variety, usually structured data, and is generally used to answer specific questions.

This has led some to ponder whether Big Data could lead to the disappearance of Small Data, or whether studies based on Small Data could be diminished due to its limitations of size, temporality, relativity, and cost. Indeed, Sawyer notes that funding agencies are increasingly pushing their limited funding resources into data-rich areas and big data analytics at the expense of small data studies, a trend that has continued in recent years (Kitchin, 2013).

The distinction between small and large data is recent. Before 2008, data was rarely considered in terms of "small" or "large." All data was, in effect, what is now sometimes called "small data", regardless of its volume. Due to factors such as cost, resources, and difficulties in generating, processing, analyzing, and storing data, limited volumes of high-quality data were produced through carefully designed studies using sampling frames designed to ensure representativeness (Kitchin & Lauriault, 2015).

So, the term "large" is somewhat misleading, as big data is characterized by much more than volume. Some "small" data sets can be very large, such as national censuses that also seek to be comprehensive. However, census data sets lack speed (usually done once every 10 years), variety (usually around 30 structured questions), and flexibility (once a census is established and administered, it is almost impossible to modify questions or add new questions) (Kitchin, 2014).

There are a variety of definitions about Small Data, which have been put forward since the early 1990s, but more recently, Thinyane described Small Data as: A perspective of Small Data as a human-centered approach to data valuation

(Thinyane, 2017). In turn, Table 3.15 (Definitions of Small Data) shows the most important definitions of Small Data through the years.

Table 3.15 Definitions of Small Data.

Autor	Definition
Miller, 2010	***“Studies supported by data produced in strictly controlled ways using sampling techniques that limited their scope, temporality., size and variety, and that they tried to capture and define their levels of error, bias, uncertainty, and provenance”.***
Bonde, 2013	***“Small data connects people with timely, meaningful insights (derived from big data and/or “local” sources), organized and packaged – often visually – to be accessible, understandable, and actionable for everyday tasks”.***
Shea, 2014	The few key pieces of meaningful, actionable information that we can uncover by analyzing big data. Those insights you extract from your big data become the last steps along the way to making better decisions.
Best, 2015	An alternative framing that focuses on the micro level analysis, and that focuses on undertaking analysis of data at the same unit at which the data is sampled, is the small data approach.
Song & Zhu, 2016	Meaning those data that do not necessarily possess all the first 4Vs of big data but still have value. Hence, small data are not a concept that describes the volume but is a relative concept to big data. Similarly, by ‘small data analytics’, we mean data analytics that does not necessarily involve big data specific technologies (i.e. Hadoop and NoSQL), but involve general techniques (i.e. statistics, data mining, machine learning, and visualization).

Comparative Big Data in Large Business and Big Data in Small Business

Table 3.16 Differences between Big Data in Large Business and Big Data in Small Business.

Characteristics	Big Data in Small Business	Big Data in Large Business
Volume	In the range of GB to TB (10,000 – 100,000 records).	In the range of TB to ZB (1,000,000 – 1,000,000,000 records).
Velocity	Controlled and steady flow of data, accumulation of data is Slow.	Data arrives at very fast speeds; Huge amount of data gets accumulated within a short period of time.
Variety	Limited to wide (Structured Data).	Wide (huge variety of data).
Veracity	Contains less noise as data is collected in a controlled manner.	The quality of data is not guaranteed. Rigorous validation of data is required prior it's processing.
Value	High.	High.
Data Location	Data is located with an enterprise, local servers, regional servers, among others.	The data is present mainly in distributed storages in the cloud and in external unstructured databases of other owners and open data, combined with structured databases
Relationality Data	Strong.	Weak to strong.
Flexibility and Scalability	Low to middling.	High.

3.1.3.2 REVIEW OF ARCHITECTURES OF BIG DATA SOFTWARE SYSTEMS DEVELOPMENT PLATFORM

Managing the information captured from companies and their clients to obtain a competitive advantage has become a very expensive process when using traditional data analysis methods, which are based on structured relational databases (Sawant & Shah, 2013). This dilemma not only applies to large companies, but also to small and medium-sized companies, research organizations, governments, and educational institutions, which need less expensive computing and storage power to analyze complex scenarios and models involving images, videos, and other data, as well as textual data (Sawant, & Shah, 2013).

New sources of information include social media data, website clickstream data, mobile devices, sensors, and other machine-generated data. All these data sources must be managed in a consolidated and integrated way so that organizations obtain valuable inferences and knowledge (Chang et al., 2019).

The main objective of Big Data architecture is the analysis and processing of large amounts of data that cannot be carried out in a conventional way, because the capacities of standard storage, management, and processing systems are exceeded (Chang et al., 2019). A Big Data management architecture should be able to design systems and models for the processing of large volumes of data from innumerable data sources in a fast and economical way, which allows better decision-making.

Big Data architecture has 5 main characteristics; these characteristics are the following:

- **Scalability:** It must be possible to easily increase data processing and storage capacities.
- **Fault tolerance:** System availability must be guaranteed, even if some machines fail.
- **Distributed data:** Data is stored between different machines, thus avoiding the problem of storing large volumes of data.
- **Distributed processing:** Data processing is performed on different machines to improve execution times and make the system scalable.

- **Data locality:** The data to be processed and the processes that process it must be close to each other to avoid network transmissions that add latency and increase execution times.

With the growth of the study and development of Big Data, data architecture designs have grown exponentially. They have migrated their operation to dynamic and flexible structures that leave behind the classic rigid structures, to give way to structures with the ability to assimilate structured and unstructured data. The architectural design of Big Data must be oriented to address five characteristics recognized in Big Data, known as the "5V". These five characteristics refer to volume, speed, variety, truthfulness, and value.

Figure 3.21 Big Data architecture style shows us an example of the components that the Big Data architecture has, as well as Table 3.17 Components of Big Data architecture, which describes the function of each of these components.

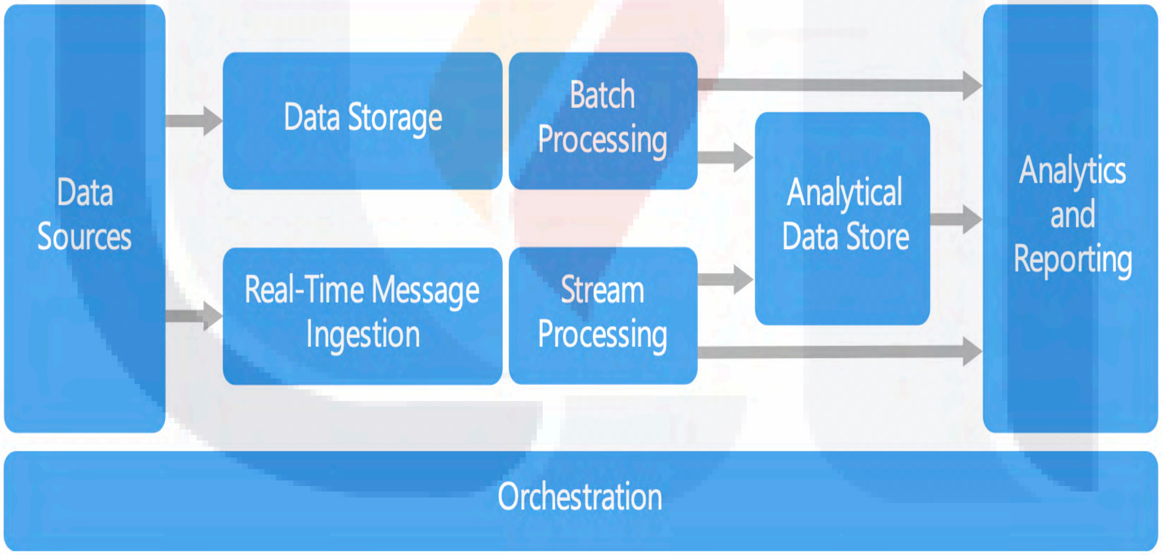


Figure 3.21 Big Data architecture style (Microsoft, 2021).

Table 3.17 Components of Big Data architecture (Microsoft, 2021).

Componentes	Descripción
Data Source	Data can be obtained from one or more sources, some of the examples can be: Data warehouses, relational and non-relational databases, statistical files produced by applications, web server log files, real-time data source, among others.
Data Storage	The data for batch processing operations is generally stored in a distributed file store that can contain large volumes of large files in various formats. This type of store is often called a data lake.
Batch Processing	Because the data sets are so large, a big data solution must often process data files using long-running batch jobs to filter, aggregate, and prepare the data for analysis.
Real Time Message Ingestion	If the solution includes real-time sources, the architecture must include a way to capture and store messages in real time for transmission processing. This could be a simple data store, where incoming messages are put into a folder for processing.
Steam Processing	After capturing messages in real time, the solution must process them by filtering, aggregating, and preparing the data for analysis.
Analytical Data Store	Many Big Data solutions prepare the data for analysis and then serve the processed data in a structured format that can be queried using analytical tools.
Analytics and Reporting	The goal of most Big Data solutions is to provide insight into the data through analysis and reporting
Orchestration	Most Big Data solutions consist of repeated data processing operations, encapsulated in workflows, that transform source data, move data between multiple sources and receivers, load the processed data into an analytical data warehouse, or push data. results directly to report or dashboard.

Before using Big Data, you must ensure that all Big Data architecture components are in place. Without this proper setup, it will be quite difficult to obtain valuable information and make correct inferences. If any of these components are missing, valuable data or correct decision-making cannot be obtained. Another example of Big Data architecture can be seen in Figure 3.22. The Big Data architecture shows us in greater detail the components of the Big Data architecture. The architecture adapts to choose Open-Source frameworks or licensed products. For the case of this thesis, we will focus on Open-Source type products only.

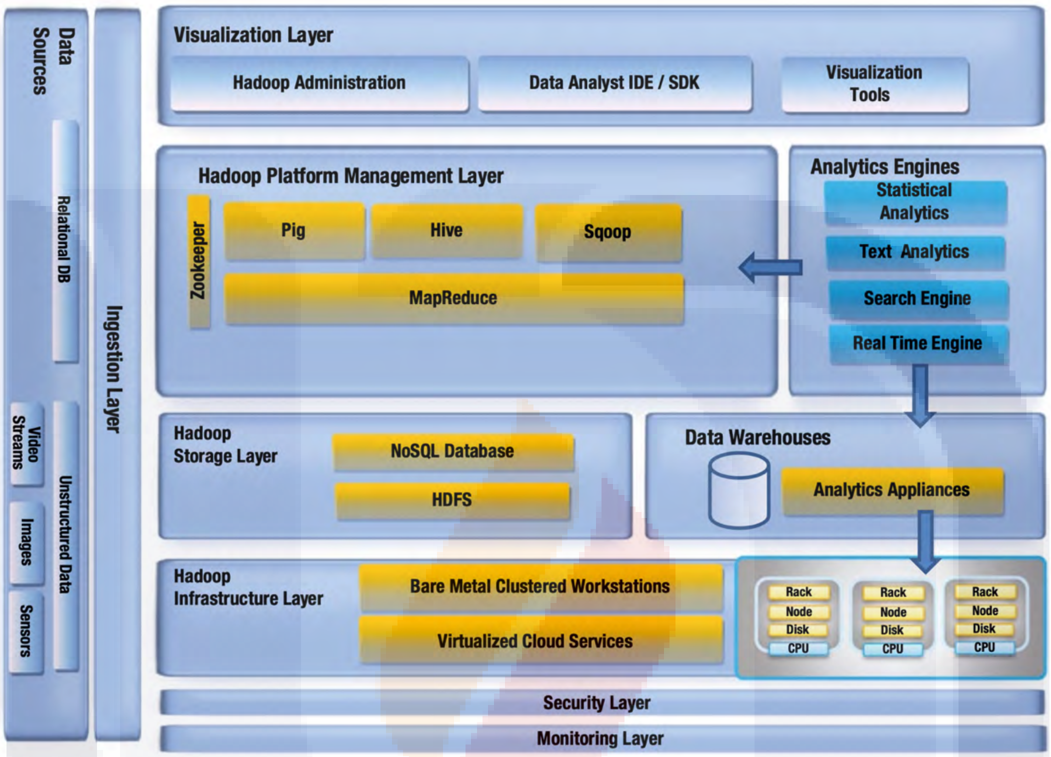


*Figure 3.22 The Big Data architecture (*Sawant, & Shah, 2013*).*

3.1.3.3 REVIEW OF TOP-6 EXEMPLARY BIG DATA SYSTEMS

Gartner Survey (2014): In 2014, only 13% of respondents said their IT organizations put big data projects into production this year, but that's 5% higher than last year. But 24% of those polled voted against the use of big data technologies in their business. 73% of respondents have invested or plan to invest in big data in the next 24 months, up from 64% in 2013. As in 2013, much of the current work revolves around strategy development and the creation of pilots and experimental projects.

There are a lot of Big Data, Analytics, Data Science or Big Data Analytics projects these types of projects can vary in technologies, timing, budgets, number of personnel required where these factors are closely related to the technology of the company the key point of these projects are the goals, they seek to meet according to the Business goals. These projects are not only limited to companies or IT research, for example at the European Bioinformatics Institute (EBI) in Hinxton (UK), which is part of the European Molecular Biology Laboratory and one of the world's largest repositories of biological data, currently stores 20 petabytes (1 petabyte is 1015 bytes) of data and backups on genes, proteins, and small molecules. Genomic data accounts for 2 petabytes, a figure that doubles every year (Marx, 2013).

Big data burst onto the scene in the first decade of the 21st century, and the first organizations to adopt it were online companies and startups. Arguably, companies like Google, eBay, LinkedIn, and Facebook were built around big data from the start.

They didn't have to reconcile or integrate big data with more traditional data sources and the analytics that came from them, because they didn't have those traditional ways. They didn't have to merge big data technologies with their traditional IT infrastructures because those infrastructures didn't exist. Big data could stand alone, big data analytics could be the only approach to analytics, and big data technology architectures could be the only architecture (Davenport & Dyché, 2013).

This is something interesting because these topics are the projects that "are fashionable" so there are many new research related to these, however due to the complexity of these projects and because they are new technologies not any company has the resources (personnel, knowledge, technologies, budget) for this type of projects so it is not so easy that any company can successfully carry out this type of projects, that is why we can see that the typical companies that are known to meet these requirements end up being those that have many resources or companies focused on technological innovation. We mention some examples, we start by mentioning cases where it can be seen that these types of projects or companies were large in terms of personnel, economic, information, or other resources. Continuing with the traditional projects, we will also see in more detail cases where these projects or technologies are not exclusive to companies with hundreds of employees, millions of data points, or extremely robust infrastructures.

Example 1: Big Data at UPS (Davenport & Dyché, 2013).

Companies like GE, UPS, and Schneider National are increasingly putting sensors into things that move or spin and capturing the resulting data to better optimize their businesses. Even small benefits provide a large payoff when adopted on a large scale. GE estimates that a 1% fuel reduction in the use of big data from aircraft engines would result in a $30 billion savings for the commercial airline industry over 15 years. Similarly, GE estimates that a 1% efficiency improvement in global gas-fired power plant turbines could yield a $66 billion savings in fuel consumption.

UPS is no stranger to big data, having begun to capture and track a variety of package movements and transactions as early as the 1980s. The company now tracks data on 16.3 million packages per day for 8.8 million customers, with an average of 39.5 million tracking requests from customers per day. The company stores over 16 petabytes of data.

Much of its recently acquired big data, however, comes from telematics sensors in over 46,000 vehicles. The data on UPS package cars (trucks), for example, includes their speed, direction, braking, and drive train performance. The data is not only used to monitor daily performance, but also to drive a major redesign of UPS drivers' route structures. This initiative, called ORION (On-Road Integrated Optimization and Navigation), is arguably the world's largest operations research project. It also relies heavily on online map data and will eventually reconfigure a driver's pickups and drop-offs in real time. The project has already led to savings in

2011 of more than 8.4 million gallons of fuel by cutting 85 million miles off daily routes. UPS estimates that saving only one daily mile driven per driver saves the company $30 million, so the overall dollar savings are substantial. The company is also attempting to use data and analytics to optimize the efficiency of its 2000 aircraft flights per day.

Example 2: Big Data at an International Financial Services Firm (Davenport & Dyché, 2013).

For one multinational financial services institution, cost savings is not only a business goal, but also an executive mandate. The bank is historically known for its experimentation with new technologies, but after the financial crisis, it is focused on building its balance sheet and is a bit more conservative with new technologies. The current strategy is to execute well at lower cost, so the bank's big data plans need to fit into that strategy. The bank has several objectives for big data, but the primary one is to exploit "a vast increase in computing power on a dollar-for-dollar basis." The bank bought a Hadoop cluster, with 50 server nodes and 800 processor cores, capable of handling a petabyte of data. IT managers estimate an order of magnitude in savings over a traditional data warehouse. The bank's data scientists, though most were hired before that title became popular, are busy taking existing analytical procedures and converting them into the Hive scripting language to run on the Hadoop cluster.

According to the executive in charge of the big data project, "This was the right thing to focus on given our current situation. Unstructured data in financial services is somewhat sparse anyway, so we are focused on doing a better job with structured data. In the near to medium term, most of our effort is focused on practical matters—those where it's easy to determine ROI, driven by the state of technology and expense pressures in our business. We need to self-fund our big data projects in the near term. There is a constant drumbeat of 'We are not doing 'build it and they will come'—we are working with existing businesses, building models faster, and doing it less expensively. This approach is more sustainable for us in the long run. We expect we will generate value over time and will have more freedom to explore other uses of big data down the road."

International financial services firm initially acquired a big data infrastructure to exploit faster processing power. But in every case, analytics is the next frontier. Managers we talked to are building out their big data roadmaps to solve a combination of both operational and analytical needs, many of them still unforeseen.

“The opportunities for cross-organizational analytics are huge,” the Executive in charge of big data told us. “But when the firm’s executives started discussing big data, the value-add was still esoteric. So, we started instead by focusing on process efficiencies. We have 60 terabytes of what we consider to be analytics data sets, and we use compiled, multi-threaded code...and do periodic refreshes. We’re past some of the challenges associated with ‘fail fast’ and are tapping into all the advantages of Hadoop.”

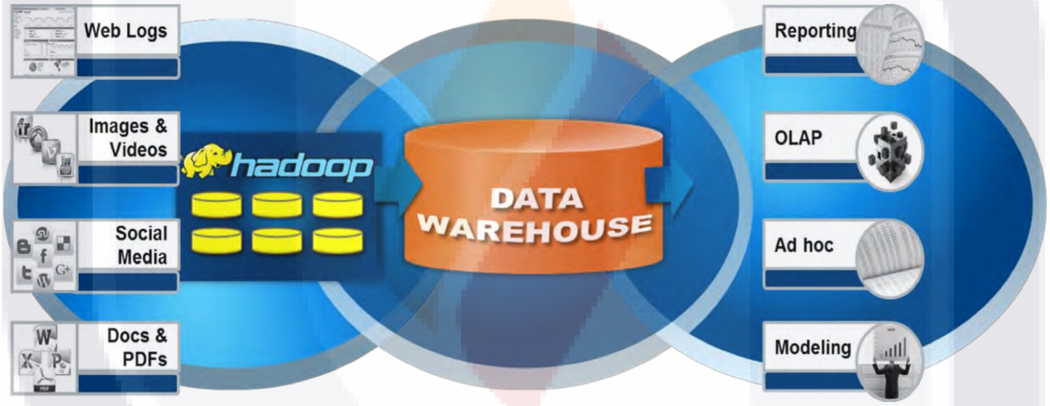


Figure 3.23 Big Data and Data Warehouse Coexistence (Davenport et al., 2013).

Example 3: Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study (Núñez et al., 2014).

Currently, in different countries, a huge amount of railway track condition-monitoring data is being collected from different sources. However, the data are not yet fully used because of the lack of suitable techniques to extract the relevant events and crucial historical information. Thus, valuable information is hidden behind a huge number of terabytes from different sensors. Considering the available data for railway condition monitoring, particularly when an increased measurement frequency is suggested to optimize maintenance decisions, these datasets qualify as Big Data. Thus, the popular 5V for railway infrastructure is analyzed.

- **Volume:** Railway infrastructure is a distributed parameter system, which implies that the assessments should consider spatial and temporal dimensions. Monitoring the entire Dutch railway (more than 6500 km of tracks) with the ABA system, only one time with different measurements provides a data volume of several terabytes. For example, when the system is implemented on commercial passenger trains to collect data all day, the data volume can exceed 100 terabytes a day because of the sampling speed of the required sensors (at least 25600 Hz for sampling and 16 sensors). A reduction/simplification of the specifications can compromise hit rates of defects and the quality of the high-frequency analysis.
- **Velocity:** With the requirement for early detection of problems and the desire to obtain good sight in the growth of defects, daily or weekly data acquisition is necessary. The main challenge with the current system is the processing time, which partly depends on human analysis of the data. Thus, the system update is currently a slow manual procedure. Moreover, when we collect data with an even higher frequency, this processing velocity is simply not feasible. Thus, computational intelligence is required to effectively process the available data, draw conclusions, and decide on the best maintenance action.
- **Variety:** In the railway infrastructure, different data-collecting systems are used, which leads to a wide variety of available data. In this paper, the data range from raw acceleration data of the wheels to images of the rail.

- **Veracity:** Different data sources have their challenges when they are used to analyze railway track conditions. The results extracted from the ABA data can be different for the same defect in two runs, which depend on the wheel position on the track concerning the defect. Although this problem is not present in the ultrasonic and eddy-current data, defects may go unnoticed because of reflections and other side effects of these techniques. For video imaging, only visible problems can be noticed. Deep cracks that do not penetrate the surface may be unobserved. Thus, the quality of each data source and the reliability of the conclusions drawn may differ.
- **Value:** Social aspects, such as reduction of delays and the optimal track usage, are the most evident benefits when the performance and availability of public transport services are improved. Collecting railway infrastructure data daily will provide valuable data to facilitate maintenance decisions and a valuable data source for further research on the causes and growth of rail defects.

There is great potential for using Big Data to facilitate maintenance decisions on Dutch railways. First, the ABA system can be implemented on a selected number of passenger trains and combined with night data from separate runs of video imaging and other systems. This method results in the collection of approximately 1 terabyte of raw data per day for the ABA data. By using selective data processing, based on previous results and experience in the growth rate of defects, all parts of the track can be monitored with appropriate intervals while maintaining the processing load within feasible limits. By also incorporating the failure and maintenance information in the system, the system can be adaptive and self-learning. In addition to the significant reduction of maintenance costs, this system can prove to be highly valuable for research by providing unprecedented amounts of track degradation data. Further studies that include the analysis of computational intelligence methodologies are considered.

Example 4: Big Data Techniques for Public Health: A Case Study (**Katsis et al., 2017).**

Public health researchers increasingly recognize that to advance their field they must grapple with the availability of increasingly large (i.e., thousands of variables) traditional population-level datasets (e.g., electronic medical records), while at the same time integrating additional large datasets (e.g., data on genomics, the microbiome, environmental exposures, socioeconomic factors, and health behaviors). Leveraging these multiple forms of data might well provide unique and unexpected discoveries about the determinants of health and well-being. However, we are in the very early stages of advancing the techniques required to understand and analyze big population-level data for public health research.

To address this problem, this paper describes how we propose that big data can be efficiently used for public health discoveries. We show that data analytics techniques traditionally employed in public health studies are not up to the task of the data we now have in hand. Instead, we present techniques adapted from big data visualization and analytics approaches used in other domains that can be used to answer important public health questions, utilizing these existing and new datasets. Our findings are based on an exploratory big data case study carried out in San Diego County, California, where we analyzed thousands of variables related to health to gain interesting insights on the determinants of several health outcomes, including life expectancy and anxiety disorders. These findings provide a promising early indication that public health research will benefit from the larger set of activities in contemporary big data research.

A Big Data Case Study

To explore how big amounts of population-level data can be leveraged to make interesting public health discoveries, we worked on a case study centered on public health issues in San Diego County, California. The choice of location was made primarily for two reasons: First, the ease of getting access to large datasets, since it is the county where UC San Diego is located. Second, the diversity of the county, which makes it especially interesting for public health researchers: San Diego

County’s location (being close to the US border with Mexico and covering a large area from the Pacific Ocean coast to the desert), magnitude (being the fifth most populous county in the US), and population characteristics give it a unique environmental, ethnic, and socioeconomic diversity.

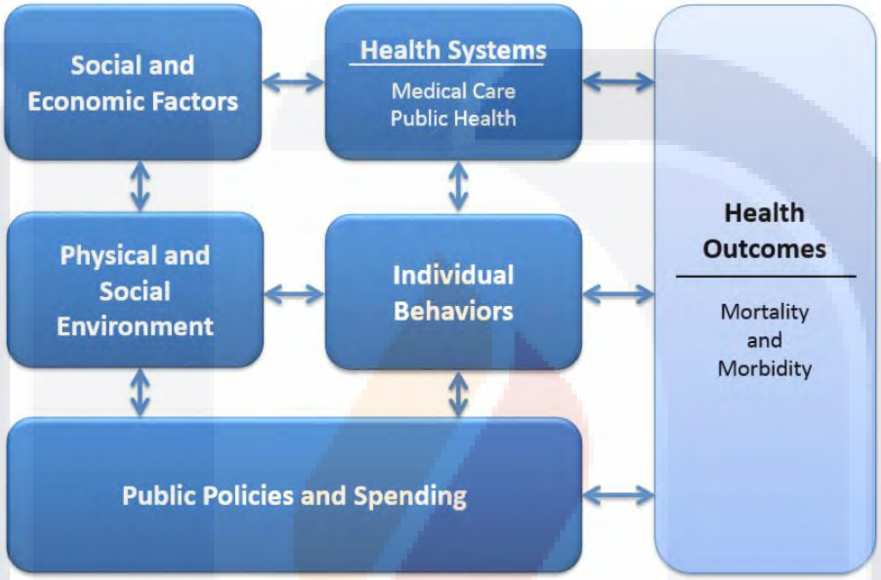
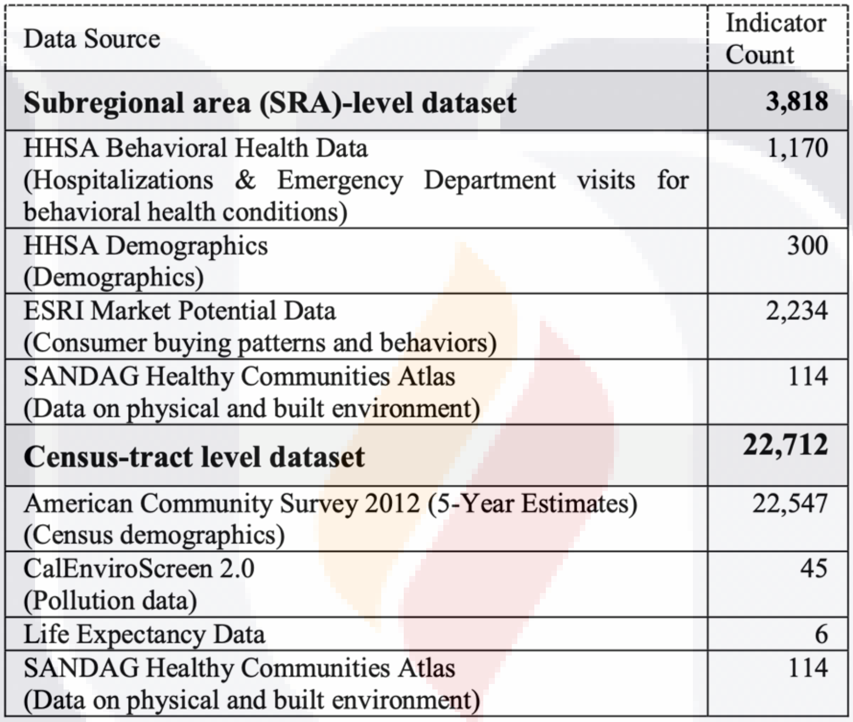


Figure 3.24 High-level grouping of determinants of health (Katsis et al., 2017).

To bootstrap our study, we identified and integrated a large number of representative data (in the order of thousands of indicators) covering the high-level groups of factors that are known to affect our health (shown on the past Figure)social and economic factors (such as education and income), physical and social environment (such as traffic density and air pollution), individual behaviors (such as smoking, exercising, and consumer buying patterns), health systems (such as insurance status), and health outcomes (such as hospitalization and emergency department visits for different conditions).

Since different datasets were provided at different geographic granularities, we ended up with two sets of integrated data: The first dataset contained 3,818 indicators at the level of the subregional areas (SRAs) (of which there are 41 in San Diego County). While this dataset contained important health outcome information (i.e., hospitalization and emergency department visit data for different conditions), its geographic granularity was restricted due to privacy reasons.

Therefore, we also created a second dataset that contained 22,712 indicators at the level of census tracts (of which there are 628 in San Diego County). The next Figure shows the data that were integrated into each of the two datasets.



Data Source	Indicator Count
Subregional area (SRA)-level dataset	**3,818**
HHSA Behavioral Health Data (Hospitalizations & Emergency Department visits for behavioral health conditions)	1,170
HHSA Demographics (Demographics)	300
ESRI Market Potential Data (Consumer buying patterns and behaviors)	2,234
SANDAG Healthy Communities Atlas (Data on physical and built environment)	114
Census-tract level dataset	**22,712**
American Community Survey 2012 (5-Year Estimates) (Census demographics)	22,547
CalEnviroScreen 2.0 (Pollution data)	45
Life Expectancy Data	6
SANDAG Healthy Communities Atlas (Data on physical and built environment)	114

Figure 3.25 Contents of the two integrated datasets used in the case study (Katsis et al., 2017).

To analyze the data, we experimented with two broad classes of big data analytics techniques that cover the two ends of the spectrum between targeted hypothesis-driven discovery and open-ended data-driven exploration: To answer specific questions, such as computing the factors that affect the life expectancy of the county's residents, we used traditional data analytics techniques, borrowed from the machine learning literature. To allow more open-ended discoveries we implemented a visual data exploration platform that allows public health researchers to visually explore the data and their correlations.

Example 5: Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft (Robbes et al., 2013).

Microsoft has a search group dedicated to empirical software engineering1 and Google employs at least 100 engineers to improve its analytics-based tools (www.infoq.com/ presentations/Development-at-Google). Software analytics has been widely accepted in the large enterprise sector. However, most companies are not able to invest as much in software analytics because most of them are small. According to It Richardson and Christiane Gresse von Wangenheim, 85% of software companies have fewer than 50 employees2; in Brazil, 70% have fewer than 20 employees3; in Canada, 78% have fewer than 25 employees4; and in the United States, approximately 94% have fewer than 50 employees5. Are software analytics viable for small software companies that are not able to exploit economies of scale, have less spare labor, and have less historical information in their software repositories than companies dealing with large software systems, such as Google or Microsoft? We decided to explore this question in a small company called Amisoft by conducting interviews (see sidebar "Note on methodology").

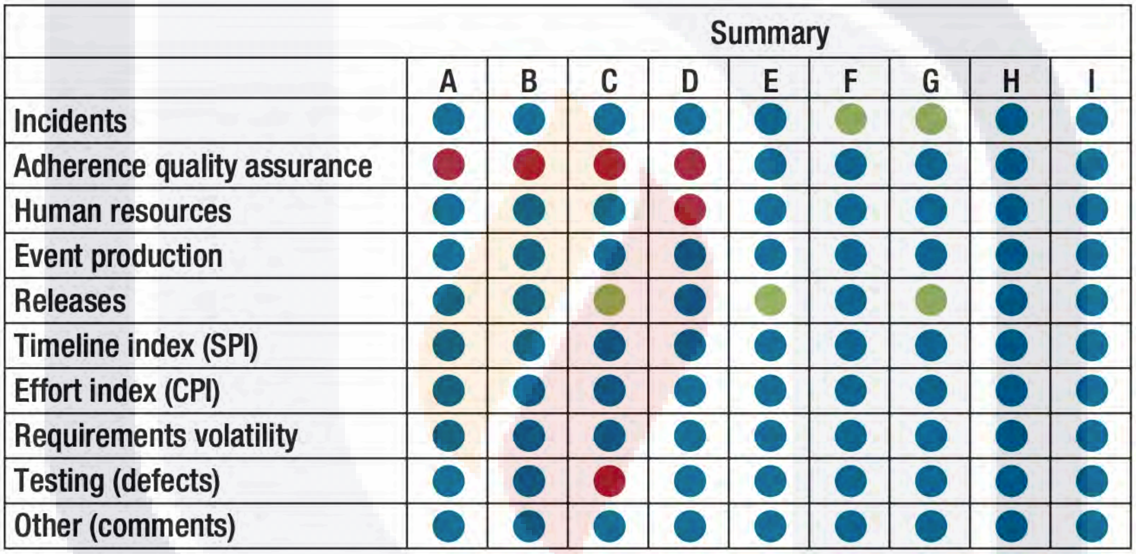
Amisoft is a 15-year-old software company based in Santiago, Chile. Its main activity is custom software development and maintenance of existing systems. Amisoft is also starting to develop standard products to complete its service offering. The company averages two new development projects per year; however, its seven definitive maintenance contracts are the projects that provide financial stability. Amisoft has 43 employees: 40 work directly in software maintenance and development. Each employee performs more than one of the company's traditional software engineering functions (developer, analyst, tester, etc.).

Case study: Increasing Reactivity to reduce Work Overload

One characteristic of our data collection process is that most of the metrics are updated weekly. Project managers have used analytics to react to delays (for instance, by rescheduling) and get back on track quickly rather than letting delays accumulate; increased effort is punctual rather than sustained.

Given the absence of hard data for the period before the analytics were introduced at Amisoft, we must rely on anecdotal evidence. Based on the CEO’s experience, the situation at Amisoft (once the improved process was introduced) was that most projects were delivered on time but had very high cost in staff-hours and required sustained effort later in the project. Today, the effort is much more evenly distributed but achieves the same results.

To evaluate the reduction in sustained late efforts and the associated burnout, we analyzed the evolution of the CPIs and SPIs of individual iterations to locate rapid adjustments to trends. Iterations usually last between three and six weeks, so weekly metric updates let the team adjust its workload accordingly. We analyzed the data from 29 iterations of five projects and classified each of the resulting 58 metric trends in three categories (see the next Figure 28).



Summary									
	A	B	C	D	E	F	G	H	I
Incidents	🔵	🔵	🔵	🔵	🔵	🟢	🟢	🔵	🔵
Adherence quality assurance	🔴	🔴	🔴	🔴	🔵	🔵	🔵	🔵	🔵
Human resources	🔵	🔵	🔵	🔴	🔵	🔵	🔵	🔵	🔵
Event production	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
Releases	🔵	🔵	🟢	🔵	🟢	🔵	🟢	🔵	🔵
Timeline index (SPI)	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
Effort index (CPI)	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
Requirements volatility	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
Testing (defects)	🔵	🔵	🔴	🔵	🔵	🔵	🔵	🔵	🔵
Other (comments)	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵

Figure 3.26 High-level status of projects at Amisoft. From this view, project managers and general managers can drill down and inspect metrics and their evolutions, reacting to deviations from set objectives. (Robbes et al., 2013).

Furthermore, we looked at the CPI and SPI values at the end of each iteration to determine whether the stated goal of 0.8 or above was reached. This occurred 81 percent of the time; 66 percent of the time, it was above 0.9. This shows that projects react quickly to delays during an iteration. Before Amisoft implemented analytics, delays would often go unnoticed until much later in the iterations, at which point they could have grown to be as large as 50 percent. This would cause considerable risks to the projects, including burnout of employees working long hours or significant delays if a critical employee fell sick at the wrong time. By monitoring the status more often, these situations are much rarer.

Software analytics are worthwhile if you follow a process. The main lesson we extracted from this experience is that software analytics are worthwhile, even for a small company like Amisoft. They bring visibility and predictability to the software development process and allow companies to gather evidence in support of a wide range of decisions, from decisions too small to be recorded to long-term changes in company strategies. But data analysis practices lack maturity. Such practices need to be formalized and shared: each project manager used the metrics differently. With additional experience and practice sharing, we expect patterns of data analysis to emerge and be consistently adopted by managers. The discovery and consolidation of said patterns should be the data analysts' responsibility.

Example 6: Intelligent decision-making of online shopping behavior based on the Internet of Things (Yan et al., 2020).

With the rapid development of artificial intelligence technology and network technology, the Internet of Things has gradually become mainstream in social development in the future. Under this background, the trade retail industry needs to establish its customer relationship network in combination with artificial intelligence technology. At the same time, it needs to conduct law mining in combination with customer selection behavior in the network and carry out personalized excavation of customers under the support of data mining technology to help customers make decisions. On this basis, it can effectively enhance the customer experience. The research on intelligent customer networks has entered a climax since 2010, and related research also provides the basis for the creation of this article.

The intelligent customer relationship network usually uses the customer's equipment movement trajectory data, customer platform operating data, customer network base stations, and other content as customer behavior data. Using this data, researchers started relevant research (Wang & Yu, 2017). Mariscal et al. designed and implemented a time-awareness system that can be used to personalize the taxi drivers travel route with the greatest benefit per unit of time (Mariscal et al., 2010).

Based on the different advertising platforms, Purtova et al. proposed an advertisement delivery system, TMAS, that is suitable for mobile web pages and mobile phone apps by analyzing customer location and related situational information and fully exploiting the mobility of customers in the mobile commerce system (Purtova, 2011). Saponara et al. designed a personalized travel package recommendation system based on tourist interest preferences, which can recommend a set of personalized and best-suited attraction collections for tourists (Saponara & Bacchillone, 2012). Kroeckel et al. studied and analyzed the mobile customers' check-in data to obtain various features of the location social network; based on this, a location-based recommendation algorithm was designed and implemented (Kröckel & Bodendorf, 2012). With the progress of research, many personalized recommendation systems for mobile clients have also been successfully launched, such as the Facebook mobile application of personalized push ads, the personalized Bizzy recommended by local shops, and the personalized reading system Zite (Palomo et al., 2012).

Long proposed a detection method for mobile App ranking fraud by exploring a personalized preferences mining method for mobile customers based on context awareness (Akhilomen, 2013). Long discussed security privacy issues under personalized recommendation technology, and he proposed a mobile App recommendation algorithm to protect customer information security against this issue. Feng, based on statistical analysis of many microblog customer data, proposes a method for personalizing popular micro topics by calculating similarities between microblog customers and micro topics. In addition, in terms of data sparsity and cold-start problems faced by collaborative filtering, Bedi et al. proposed the use of the K-nearest neighbor method to map “attribute-feature” and calculate the feature vectors of new customers and new projects (Watters et al., 2013). Islam proposed using a combination of data migration and data clustering to solve the system could start problem (Tsai et al., 2014). To solve the problem of sparseness in collaborative filtering algorithms, Zuech proposes a way of thinking that the clustering is based on the attributes of the project and uses the mean of the project categories to fill in the null values in the original scoring data (Ravizza et al., 2014). At present, major e-commerce platforms at home and abroad have developed their mobile terminals. However, the search and application of personalized recommendation systems for mobile platforms is still in its infancy (Li et al., 2014), and there is still room for improvement in their recommendation quality and operating efficiency (Chin et al., 2018).

Ravizza first proposed the idea of considering the trust between customers in the recommendation process. The trust between customers is established through the displayed customer trust evaluation and debilitating spread (Liu et al., 2018). The trust is divided into reliability trust and decision trust (Kim & Park, 2013). The reliability trust is the subjective probability that entity A acts according to entity B's expectations, and decision trust refers to the subjective degree of relative security feeling obtained by an individual trusting a certain entity in a certain environment (Banker, 2014). Watters uses the ratio of the number of customer recommendations to the total number of recommendations as the degree of trust between customers and applies this calculation method to the recommendation system, where the confidence value ranges from [0,1]. Saponara et al. proposed a trust model based on fuzzy logic representation, based on the fuzzy nature of trust relationships (Zuech et al., 2015).

Benefiting from the development of Internet of Things technology and data mining technology (Dijkman et al., 2015), the spread of consumer trust has become multi-directional. As Kim and Park mentioned, all the characteristics of s-commerce (except for economic feasibility) had significant effects on trust, and that trust had significant effects on purchase intentions. Hence, the characteristics of consumer trust, communication, and decision-making behavior under the Internet of Things are necessary to study.

Based on the above analysis, we can see that the current decision model based on the Internet of Things to build a customer relationship network is less researched, and most of them are recommending unilateral information to customers based on personalized recommendations (Chen et al., 2018). Therefore, based on the Internet of Things technology, this study builds a more complete customer relationship network based on personalized recommendations, and adopts a proven collaborative filtering recommendation algorithm as a basis for decision models to extract contextual features that characterize customer trust. At the same time, this research uses the analytic hierarchy process to complete the model-building process, helps customer relationship network service objects to provide decision support, completes product information recommendation, solves new customer cold start problems, and improves existing scoring prediction formulas.

Seeing these 6 examples, we can conclude that topics such as Big Data, Analytics, Data Mining, or decision making, can be performed in any type of company, even in infrastructure or quantity, and types of data.

This is because according to the theory, to apply to Big Data projects, it is necessary to have an amount of data over Terabytes, an amount that is not possible to process with the resources of a standard organization due to the traditional way of processing, But as Adibuzzaman mentions, in the health area there are not always millions and millions of data which even when being analyzed from thousands of records that can be had on the subject according to the requirements of the research or the limitation of public data become even just a few tens of data to analyze, but this does not mean that the study or the results have no relevance (Adibuzzaman, et al., 2017).

Even Garner publishes " Top 10 Data and Analytics Trends for 2021 " where Trend 4 is from big to small and wide data, just where he mentions that " Small and wide data, as opposed to big data, solves several problems for organizations facing increasingly complex questions about AI and challenges with sparse data use cases. Big data - leveraging "X-analytics" techniques - enables the analysis and synergy of a variety of small and varied (big), unstructured and structured data sources to improve contextual knowledge and decisions. Small data, as the name implies, is capable of using data models that require less data but still provide useful insights." (Gartner,2021).

Below in Table 3.18 is a comparison between the examples mentioned above, where the 5 main characteristics of Big Data are compared, such as volume, speed, variety, truthfulness, and value. In this table we can see how the first 3 examples use the Big Data approach, that is, a Big Data in Large Business, while the three subsequent examples use a reduced Big Data approach, with much less volume and speed of data but preserving the value the veracity and the variety of the data, this is called Big Data in Small Business.

Table 3.18 Comparative table of Big Data software systems for large Business vs Small Business characteristics in 6 the examples.

Case	Characteristics				
	Volume	Velocity	Variety	Veracity	Value
1. Big Data at UPS (Davenport & Dyché, 2013).	16 petabytes	16.3 million new packs daily	High, data on packages, customers, requests, maps, vehicles, and sensors	Storage of your own data, generated by your processes or actions, your sensors, or modules.	High, UPS estimates that saving only one daily mile driven per driver saves the company $30 million,
2. Big Data at an International Financial Services Firm (Davenport & Dyché, 2013)	60 terabytes	Hit, millions of daily transactions for dollar-for-dollar calculations	Structured	50 server nodes and 800 processor cores, capable of handling a petabyte of data	Hit, a big data infrastructure to exploit faster processing power
3. Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study (Núñez et al., 2014)	100 terabytes accumulation day by day	Higt, 100 terabyte a day	Different data-collecting systems are used, which leads into a wide variety of available data	The quality of each data source and the reliability of the conclusions drawn may differ	High. Social aspects, such as the reduction of delays and the optimal use of roads and the availability of public transport services

4. Big Data Techniques for Public Health: A Case Study (Katsis et al., 2017)	Data from only 26,530 indicators	The generation of these data is slow since they are indicators recorded for years, for example, for 5 years a total of 22547 were generated.	Structured, diferentes Datasets	High, due to the source of the product	High. Big data was effectively used to analyze thousands of health-related variables to gain interesting insights into the determinants of various health outcomes.
5. Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft (Robbes et al., 2013)	The data from 29 iterations of five projects and classified each of the resulting 58 metric trends	Data of the processes captured weekly (less than 100 weekly records)	Structured	Given the absence of hard data for the period before the analytics were introduced at Amisoft, we must rely on anecdotal evidence	To evaluate the reduction of late efforts and associated attrition. To locate rapid trend adjustments.
6. Intelligent decision-making of online shopping behavior based on internet of things (Yan et al., 2020)	298 customers' click browsing records as training data, and collected 50 customers who used the platform for the first time as research objects	Data captured at the beginning of the experiment	The customer's equipment's movement trajectory data, customer platform operating data, customer network base stations, and other content as customer behavior data.	Data may vary due to user behavior and the way in which they are obtained	Customer's consumer experience can be enhanced with the support of data mining technology in cyber intelligence

3.1.3.4 REVIEW OF OPEN-SOURCE SOFTWARE DEVELOPMENT PLATFORMS FOR BIG DATA SYSTEMS

There is a wide range of systems and tools that are used for the development of Data Science / Analytics systems. The Data Science / Analytics community is, in general, quite open and generous, which means that many of the tools and libraries are Open-Source.

This indicates that there are many programming languages that allow us to develop in Data Science / Analytics. A study by Kdnuggets shows the most popular languages for the development of Data Science / Analytics projects in the industry. As we can see in Table 3.19, Programming languages for Data Science / Analytics, Python and R are the two most used languages, with a wide advantage over the others.

Table 3.19 Programming languages for Data Science / Analytics (Kdnuggets, 2019).

Platform	2019 % share	2018 % share	% change
Python	65.8%	65.6%	0.2%
R Language	46.6%	48.5%	-4.0%
SQL Language	32.8%	39.6%	-17.2%
Java	12.4%	15.1%	-17.7%
Unix shell/awk	7.9%	9.2%	-13.4%
C/C++	7.1%	6.8%	3.7%
Javascript	6.8%	na	na
Other programming and data languages	5.7%	6.9%	-17.1%
Scala	3.5%	5.9%	-41.0%
Julia	1.7%	0.7%	150.4%
Perl	1.3%	1.0%	25.2%
Lisp	0.4%	0.3%	46.1%

That is why, for this thesis, we will analyze three of the most widely used languages in the world, Python, R, and Java, which we will analyze with different criteria that allow us to select one of the languages to be used in this thesis. Below is a brief description of each of these programming languages focused on Data Science / Analytics developments, as well as the tools and libraries that each of them would use.

Python

Python is a general-purpose object-oriented programming language due to its extensive library that primarily enables the development of Big Data, Artificial Intelligence (AI), Data Science, Test Frameworks, and Web Development applications. Released in 1989, Python is easy to learn and a favorite with programmers and developers. Python is one of the most popular programming languages in the world, second only to Java and C (IBM, 2021).

There are several libraries and tools allow us to carry out tasks and Data Science / Analytics developments for this specific thesis, we will consider 4 of the most important tools and libraries that exist for the development of Data Science / Analytics in Python, these are the following:

- Jupyter is a web-based iterative development environment for notebooks.
- Numpy is used to handle large matrices.
- Pandas for data manipulation and analysis.
- Matplotlib is used to create data visualizations.

Also, Python is especially well-suited for implementing machine learning on a large scale. Its suite of specialized libraries enables data scientists to develop sophisticated data models that connect directly to a production system.

R Language

R is an Open-Source programming language that is optimized for statistical analysis and data visualization. Developed in 1992, R has a rich ecosystem with complex data models and elegant data reporting tools (IBM, 2021).

The interface and structure are very suitable for tasks related to algorithms and data modeling. R has hundreds of libraries, which have made it one of the most developed systems that has thousands of packages to solve a wide variety of problems.

Popular among Data Science / Analytics academics and researchers, R provides a wide variety of libraries and tools for creating Data Science / Analytics tasks. For this thesis, we will focus on three main tools for this task. These tools and libraries are:

- RStudio is an integrated development environment for simplified statistical analysis, visualization, and reporting.
- Dplyr for data cleaning and preparation.
- Ggplot2 for creating visualizations.

Java

Java is an object-oriented programming language specifically designed to allow developers a continuity platform. It is an extremely popular language that runs on a virtual machine, allowing it to be run on any type of device without having to compile it repeatedly. Java was created by Sun Microsystems in 1991 as a programming tool and an object-oriented language, allowing programmers to generate autonomous code fragments, which interact with other objects to solve a problem, offering support for different technologies.

Compared to other specific languages such as R and Python, Java does not have many libraries for advanced statistical methods, which makes languages such as R and Python much more recommended for the development of Data Science / Analytics tasks. However, different tools and libraries will allow us to develop this type of application. For this thesis, we will take three of the most important tools for the development of Data Science / Analytics applications, these are:

- Weka is a collection of machine learning algorithms for data mining tasks.
- Rapid Miner is a data mining tool.
- KNIME is a data mining platform that allows the development of models in a visual environment.

These three languages are evaluated with the criteria and attributes proposed in the work A MADM Risk-based Evaluation-Selection Model of Free-Libre Open-Source Software Tools, proposed by Mora et al. (2016), where they propose an evaluation model based on risks of Open-Source tools. They propose 4 criteria and 32 attributes for the evaluation of Open-Source tools. For this thesis, we will take only three of these criteria and ten attributes, since these are the ones that best adapt and contain enough attributes to evaluate our three programming languages.

- **Operational Risks:** External Reviews, Internal Experience, Interested IT Staff, Project Leader, Trained End User Group, Top Management Support, Training, Usability, and User Engagement.
- **End user risks:** Functionality-quality, market image, performance-efficiency, and utility-relevance.
- **Technical risks:** Community support, development process, developer community, and developer organization. Structure, documentation, interoperability-portability, maintainability, maturity-longevity, project fork, security-reliability, test information, compliance with standards, technical environment, and user community.

Figure 3.27 MADM risk-based evaluation-selection FLOSS tool model shows the three criteria and the 10 attributes that will be used in this thesis; these criteria are Organizational Risks, End-user Risks, and Technical Risks, with their respective attributes that were evaluated.

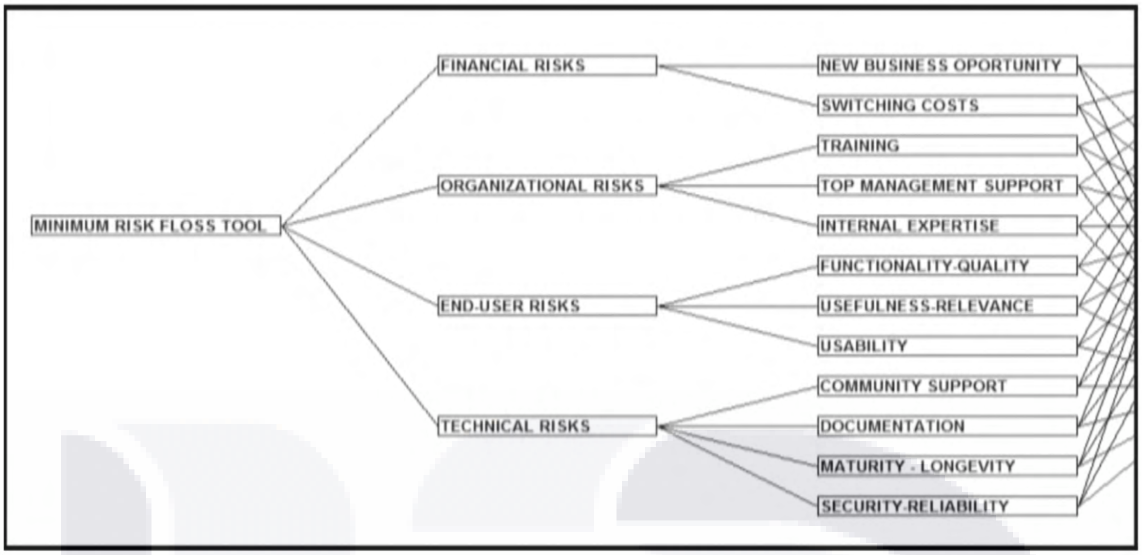
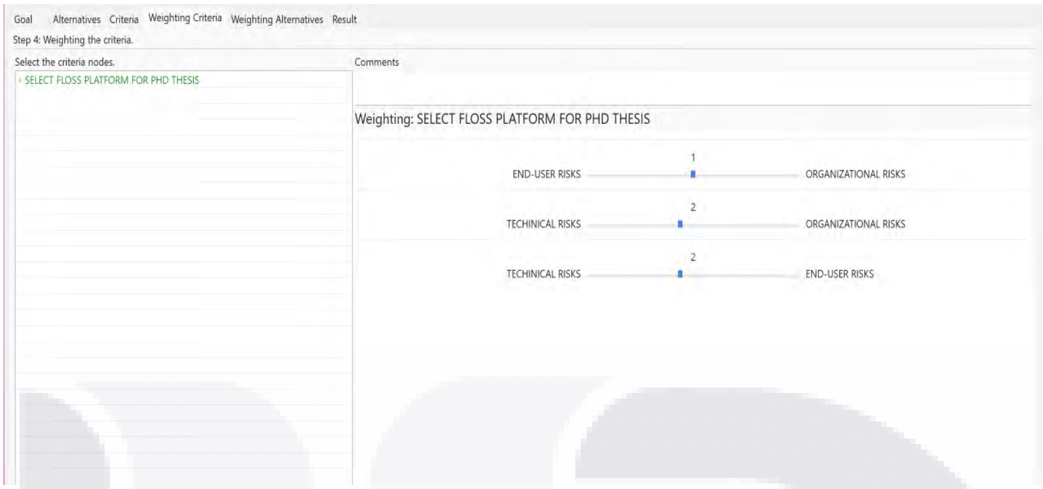


Figure 3.27 MADM risk-based evaluation-selection FLOSS tool model (Mora et al., 2016).

All these criteria and attributes were evaluated with decision-making software, which allows us to enter the alternatives, which in this case are our three programming languages, and our three evaluation criteria, together with their attributes. Each of the criteria and attributes is assigned a weight based on the research carried out on each of the languages and their tools and libraries, as well as the knowledge and experience available in each one. Of these programming languages. From Figure 3.28 to Figure 3.30, there are screenshots of the results produced by the decision-making software for our three programming languages, based on the research and experience with these.



Step 4: Weighting the criteria.

Select the criteria nodes.

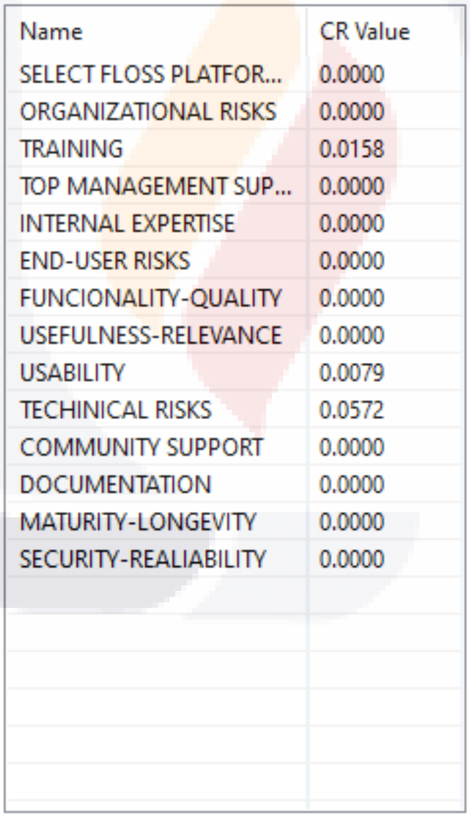
› SELECT FLOSS PLATFORM FOR PHD THESIS

Comments

Weighting: SELECT FLOSS PLATFORM FOR PHD THESIS

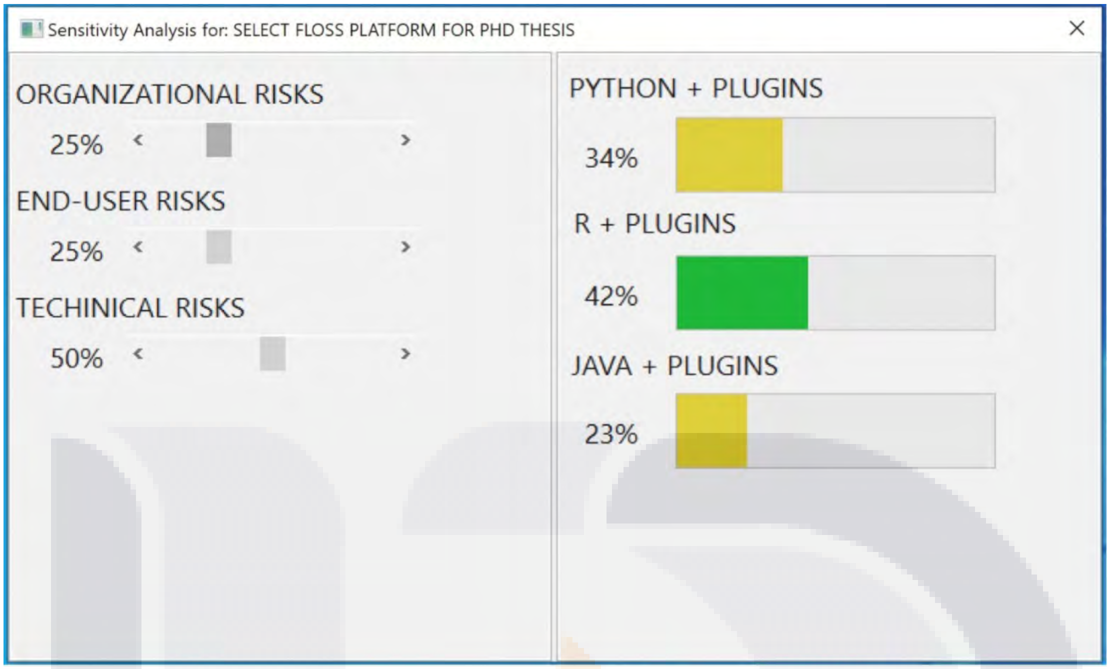
END-USER RISKS	1	ORGANIZATIONAL RISKS
TECHINICAL RISKS	2	ORGANIZATIONAL RISKS
TECHINICAL RISKS	2	END-USER RISKS

Figure 3.28 Weighting Criteria.



Name	CR Value
SELECT FLOSS PLATFOR...	0.0000
ORGANIZATIONAL RISKS	0.0000
TRAINING	0.0158
TOP MANAGEMENT SUP...	0.0000
INTERNAL EXPERTISE	0.0000
END-USER RISKS	0.0000
FUNCIONALITY-QUALITY	0.0000
USEFULNESS-RELEVANCE	0.0000
USABILITY	0.0079
TECHINICAL RISKS	0.0572
COMMUNITY SUPPORT	0.0000
DOCUMENTATION	0.0000
MATURITY-LONGEVITY	0.0000
SECURITY-REALIABILITY	0.0000

Figure 3.29 Consistency Ratios.



Sensitivity Analysis for: SELECT FLOSS PLATFORM FOR PHD THESIS

ORGANIZATIONAL RISKS 25%

END-USER RISKS 25%

TECHINICAL RISKS 50%

PYTHON + PLUGINS 34%

R + PLUGINS 42%

JAVA + PLUGINS 23%

Figure 3.30 Result Ranking.

As we can see, the Technical Risks criterion was given greater weight since it is considered that the attributes it has are of greater relevance for the study of this thesis; in turn, the two remaining criteria had the same weight among them.

At the same time, we can see that each of the criteria meets the consistency ratios, since all the attributes are below 0.1, which indicates that the weights assigned to each of the attributes are consistent and valid for research.

Finally, Figure 3.25 Ranking of Results shows us that when evaluating the criteria and attributes, the programming language that has the most value for this thesis is R + Plugins, this since R has a greater weight in the attributes of usability and functionality- quality, this because R is a language more focused on statistics and is much more used in research areas, in addition to being one of the most used by experts in Data Science / Analytics issues worldwide.

For computer science purists, Python always stands out as the right programming language for Data Science / Analytics. Rather, R is a specific language used for data analysis and statistics, uses a specific syntax used by statisticians, and is a vital part of the world of data science and research. On the contrary, for the design of Data Science / Analytics applications with the Java language, much less is used, since it is not a language with so many specific tools and libraries for the development of this type of application, which gives it a clear advantage over R and Python.

The main distinction between these two languages is in their approach to data science. Both programming languages are open source and are supported by large communities, which continually expand their libraries and tools. But while R is used primarily for statistical analysis, Python provides a more general approach to data analysis (IBM, 2021).

It is for these reasons that R is the language chosen for the use of the methodology proposed in this thesis, because it is one of the most widely used languages in Data Science / Analytics issues due to its focus on statistics and data analysis, in addition to be a language created for the development of this type of project and the most used for research and data science.

3.1.3.5 REVIEW OF THE 3 MAIN ANALYTICS/DATA SCIENCE SDM (KDD, SEMMA AND CRISP-DM)

A System Development Method (SDM) is a method or technique used to develop software. It is a broad concept that includes several phases of software development, such as design, development, and testing. It is also known as the system development life cycle (SDLC). An SDM defines the specific requirements and deliverables necessary for a project team to develop or optimize an application. In this segment, we focus on the classic SDMs for Analytics/Data Science development, both the basis for the first methodologies and the most widely used in the area today. Efforts in data mining have focused mostly on the investigation of techniques for the exploitation of information and extraction of patterns (such as decision trees, cluster analysis, and association rules). However, the process of how to execute this process until obtaining the “new knowledge”, that is, in the methodologies (Moine et al., 2011), has been deepened to a lesser extent. The methodologies allow the data mining process to be carried out in a systematic and non-trivial way. They help organizations understand the knowledge discovery process and provide guidance for planning and executing projects.

Mariscal et al. (2010) captured the state of the art of methods for data mining and knowledge discovery by comparing and adding 15 methods. The authors suggested that there are three main methodologies for the development of this type of project, which are KDD, SEMMA, and CRISP-DM. Furthermore, they argued that KDD (Knowledge Discovery in Databases) represents the groundwork for many other methods and is the ancestor of methods like CRISP-DM and SEMMA. Figure 3.31(Evolution of data mining process models and methodologies) shows the evolution of 14 data mining process models and methodologies. In which we can point to KDD as the initial focus and CRISP-DM as the central focus of evolution.

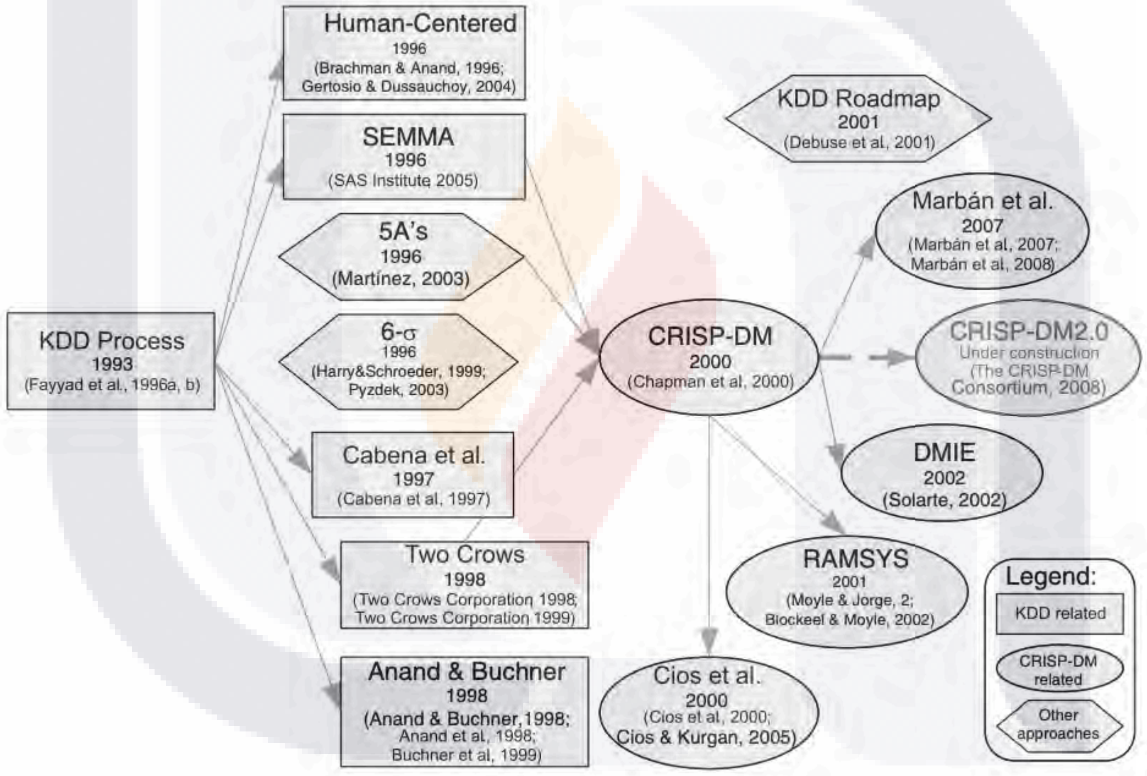


Figure 3.31 Evolution of data mining process models and methodologies (Mariscal et al., 2010).

Next, we will present these three fundamental methods, describing the phases that each of the methodologies consists of, as well as a small comparison between these three methodologies.

KDD

Data mining (DM), knowledge discovery in databases (KDD), knowledge discovery and data mining and knowledge discovery (DM and KD) are terms used to refer to research results, techniques, and tools used to extract useful information from large volumes of data (Agrawal et al., 1996). The whole process of information extraction is known as the KDD process (Frawley et al., 1991). Data mining is only one step in the entire KDD process (Fayyad et al., 1996).

In the early 1990s, when the term KDD was first coined (Piatetsky-Shapiro, 1991), there was a race to develop data mining algorithms that could solve all problems related to finding useful knowledge in large volumes of data. In addition to developing algorithms, some specific tools were also developed, such as Clementine, IBM Intelligent Miner, Weka, and DBMiner, to simplify the application of data mining algorithms and provide some support for all KDD-related activities.

KDD is the non-trivial process of finding valid, new, possibly useful, and ultimately understandable patterns in the data (Costa & Aparicio, 2020). The KDD process is an iterative and interactive, that involves numerous steps with many decisions made by the analyst.

It is essential to develop an understanding of the data, create a target data set, and clean and process it. Then, various tasks must be performed, such as data reduction and projection. The analyst also must match the objectives of the KDD process with a data extraction method, exploratory analysis, and a selection of models and hypotheses. An essential task is to interpret extracted patterns and use the knowledge directly (Costa & Aparicio, 2020).

KDD focuses on the general process of discovering knowledge from data, including how data is stored and accessed, how algorithms can be used for massive data sets, how they can be executed efficiently, and how to interpret and visualize the results (Daderman & Rosander, 2018).

The KDD process involves numerous steps with many decisions made by the user. Brachman and Anand (1996) offer a practical vision of the KDD process, emphasizing the iterative nature of the processes, the steps that KDD consists of are described below, as well as in Figure 3.32 (An Overview of the Steps That Compose the KDD Process) it shows a general description of the steps for the process. by KDD.

1. Develop an understanding of the application domain and relevant prior knowledge, and identify the goal of the KDD process from the customer's point of view.
2. **Create a target dataset:** select a dataset or focus on a subset of variables or data samples, on which discovery is to be performed.
3. **Data cleaning and pre-processing:** basic operations such as denoising if appropriate, gathering the information needed to model or account for noise, deciding strategies to handle missing data fields, accounting for time sequence information, and known changes.
4. **Data reduction and projection:** Find useful features to represent mosaic data, depending on the mosaic objective of the mosaic task. Use dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.
5. Match the mosaic goals of the KDD mosaic process to a particular data mining method: for example, summary, classification, regression, grouping, and more.
6. Choose the data mining algorithm (s): select the method (s) that will be used to look for patterns in the data. This includes deciding which models and parameters may be appropriate and matching a particular data mining method to the general criteria of the KDD process.
7. **Data mining:** search for patterns of interest in a particular form of representation or a set of such representations: classification rules or trees, regression, grouping, among others.

8. Interpreting extracted patterns, possibly go back to any of the steps 1-7 for further iteration. This step may also involve viewing the extracted patterns/models or viewing the data given the extracted models.
9. **Consolidate discovered knowledge:** incorporate this knowledge into another system for further action, or simply document it and report it to stakeholders. This also includes checking and resolving potential conflicts with previously believed (or extracted) knowledge.

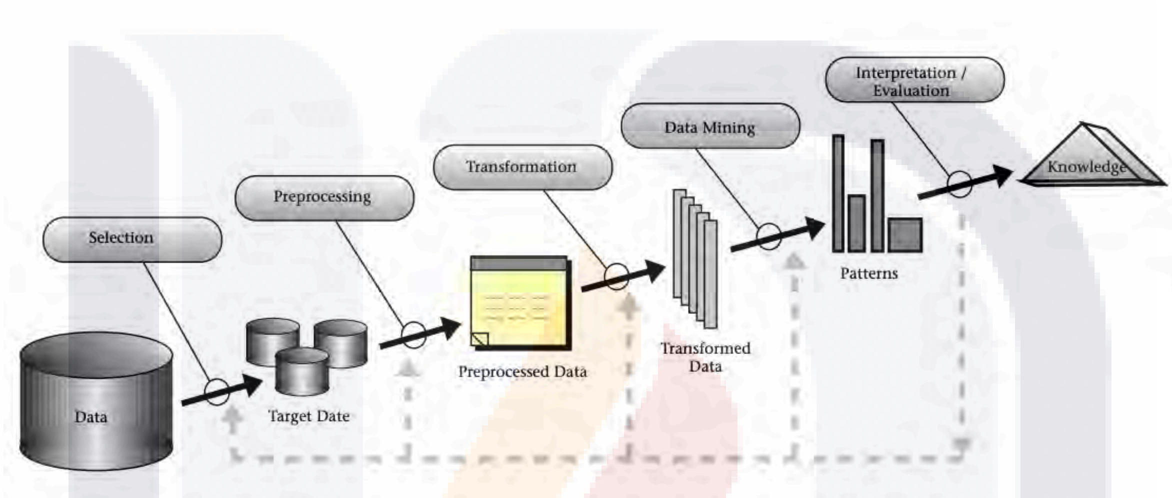


Figure 3.32 An Overview of the Steps That Compose the KDD Process (Fayyad et al., 1996).

SEMMA

SEMMA (Sample, Explore, Modify, Model, and Assess), based on KDD, was developed by SAS Institute in 2005 (SAS Institute Inc., 2017). And it is defined by these as a logical organization of the set of functional tools of SAS Enterprise Miner to carry out the core tasks of data mining. SAS Institute defines data mining as the process of sampling, exploring, modifying, modeling, and evaluating (SEMMA) large amounts of data to discover previously unknown patterns, which can be used to the business advantage. The data mining process is applicable in a variety of industries and provides methodologies for business problems as diverse as customer churn, database marketing, market segmentation, risk analysis, affinity analysis, and customer satisfaction, among others.

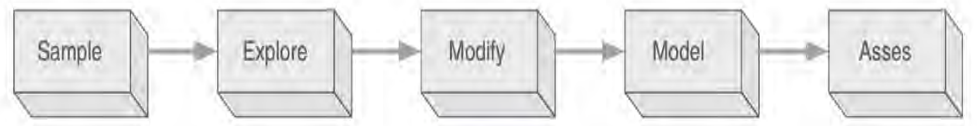


Figure 3.33 SEMMA methodology steps (Mariscal et al., 2010).

SAS Enterprise Miner software is an integrated product that provides an end-to-end business solution for data mining. A graphical user interface (GUI) provides an easy-to-use interface to the SEMMA data mining process, consisting of 5 phases described below:

- **Sample:** The data by extracting and preparing a sample of data for model building using one or more data tables. Sampling includes operations that define or subset rows of data. The samples should be large enough to efficiently contain the significant information.

- **Explore:** The data by searching for anticipated relationships, unanticipated trends, and anomalies to gain understanding and ideas.
- **Modify:** The data by creating, selecting, and transforming the variables to focus the model selection process on the most valuable attributes.
- **Model:** The data by using the analytical techniques to search for a combination of the data that reliably predicts a desired outcome.
- **Assess:** The data by evaluating the usefulness and reliability of the findings from the data mining process.

Starting with a statistically representative sample of your data (sample), SEMMA aims to facilitate the application of visualization techniques and exploratory statistics (explore), select, and transform the most significant predictive variables (modify), model the variables to predict results (model), and finally confirm the precision of a model (evaluate) (Olson & Delen, 2008).

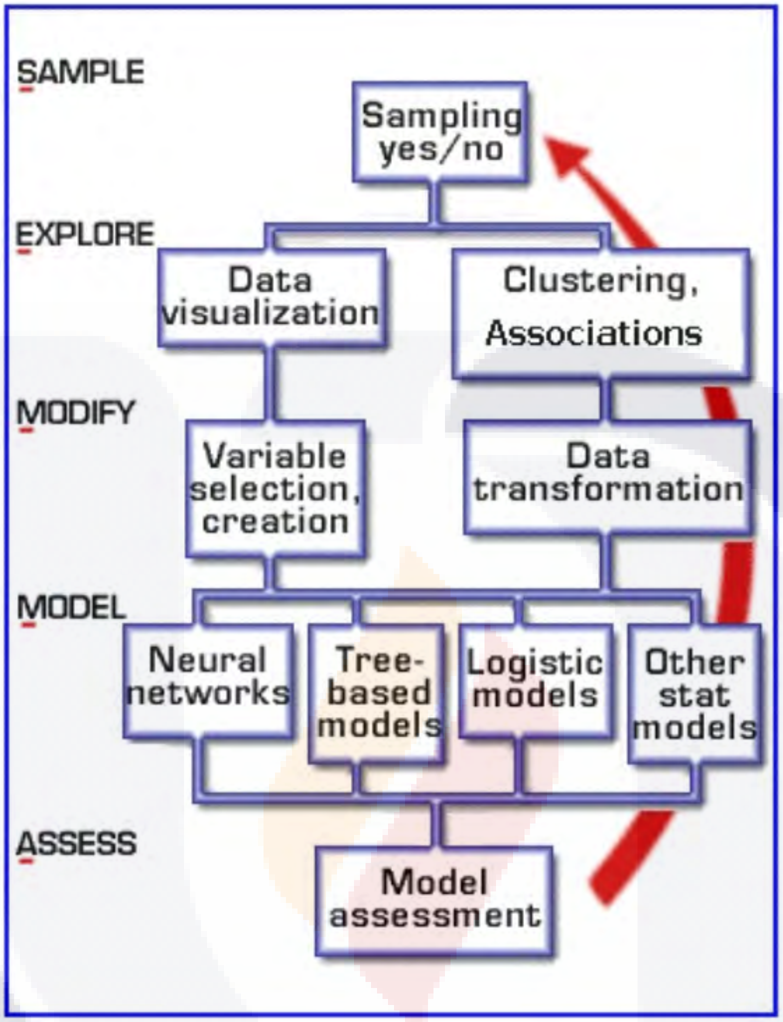


Figure 3.34 SEMMA methodology diagram (SAS Institute Inc., 2017).

The SEMMA data mining process is driven by a process flow diagram, which you can modify and save. The GUI is designed in such a way that the business analyst who has little statistical expertise can navigate through the data mining methodology, while the quantitative expert can go "behind the scenes" to fine-tune and tweak the analytical process.

Enterprise Miner contains a collection of sophisticated analysis tools that have a common user-friendly interface that you can use to create and compare multiple models. Statistical tools include clustering, self-organizing maps, variable selection, trees, linear and logistic regression, and neural networks. Data preparation tools include outlier detection, variable transformations, data imputation, random sampling, and the partitioning of data sets (into train, test, and validation data sets). Advanced visualization tools enable you to quickly and easily examine large amounts of data in multidimensional histograms and to graphically compare modeling results.

The main difference between the original KDD process and SEMMA is that SEMMA is integrated into SAS tools such as Enterprise Miner, and it's unlikely to use SEMMA methodology outside of them, while KDD is an open process, and it can be applied in very different environments. There are two other important differences between SEMMA and the original KDD process. On the one hand, SEMMA skips the first step of the KDD process, learning the application domain, and starts directly with the sample step. On the other hand, SEMMA does not include an explicit step to use the discovered knowledge, while KDD includes a step to use the discovered knowledge. These two steps are considered essential to carry out a data mining project successfully.

CRISP-DM

In response to common issues and needs in data mining project in the mid 90's, a group of organizations involved in data mining (Teradata, SPSS -ISL-, Daimler-Chrysler and OHRA) proposed a reference guide to develop data mining projects, named CRISP-DM (CRoss Industry Standard Process for Data Mining) (Chapman et al., 2000). CRISP-DM is considered the de facto standard for developing data mining and knowledge discovery projects. One important factor of CRISP-DM success is the fact that CRISP-DM is industry-, tool-, and application-neutral.

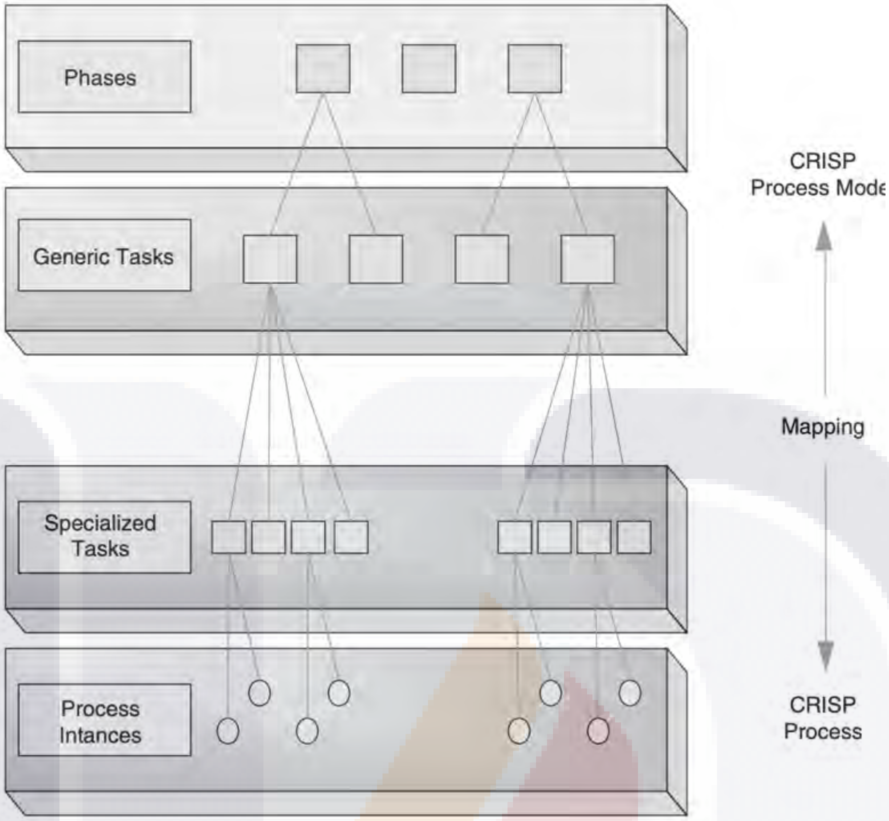


Figure 3.35 Four-level breakdown of the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology (Mariscal et al., 2010).

The CRISP-DM data mining methodology is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction (from general to specific) (Figure 3.35: Four-level breakdown of the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology).

At the top level, the data mining process is organized into several phases; each phase consists of several second-level generic tasks. This second level is called generic because it is intended to be general enough to cover all possible data mining situations. The third level, the specialized task level, is the place to describe how actions in the generic tasks should be carried out in certain specific situations. The fourth level, the process instance, is a record of the actions, decisions, and results of an actual data mining engagement.

The reference model presents a quick overview of phases, tasks, and their outputs, and describes what to do in a data mining project. The user guide gives more detailed tips and hints for each phase and each task within a phase, and depicts how to do a data mining project.

CRISP-DM distinguishes between four different dimensions of data mining contexts:

- The application domain is the specific area in which the data mining project takes place.
- The data mining problem type describes the specific classes of objectives that the data mining project deals with.
- The technical aspect covers specific issues in data mining that describe different (technical) challenges that usually occur during data mining.
- The tool and technique dimension specifies which data mining tool(s) and/or techniques are applied during the data mining project.

The CRISP-DM process model for data mining provides an overview of the life cycle of a data mining project. It contains the corresponding phases of a project, their respective tasks, and relationships between these tasks.

The life cycle of a data mining project, according to CRISP-DM, consists of six phases; the sequence of phases is not strict. It is always necessary to move forward and back between the different phases. The arrows indicate the most important and frequent dependencies between phases.

In the following statements, we outline each phase briefly (Chapman et al., 2000):

1. **Business Understanding:** The business situation should be assessed to get an overview of the available and required resources. The determination of the data mining goal is one of the most important aspects in this phase. First, the data mining type should be explained (e. g. classification) and the data mining success criteria (e.g., precision). A compulsory project plan should be created.
2. **Data Understanding:** Collecting data from data sources, exploring, describing it, and checking the data quality are essential tasks in this phase. To make it more concrete, the user guide describes the data description task by using statistical analysis and determining attributes and their collations.

3. **Data Preparation:** Data selection should be conducted by defining inclusion and exclusion criteria. Bad data quality can be handled by cleaning the data. Depending on the used model (defined in the first phase), derived attributes must be constructed. For all these steps, different methods are possible and are model-dependent.
4. **Modeling:** The data modelling phase consists of selecting the modeling technique, building the test case, and the model. All data mining techniques can be used. In general, the choice depends on the business problem and the data. Another important aspect is defining how to explain the choice. For building the model, specific parameters must be set. For assessing the model, it is appropriate to evaluate the model against evaluation criteria and select the best ones.
5. **Evaluation:** In the evaluation phase, the results are checked against the defined business objectives. Therefore, the results must be interpreted, and further actions must be defined. Another point is that the process should be reviewed in general.
6. **Deployment:** The deployment phase is described generally in the user guide. It could be a final report or a software component. The user guide describes that the deployment phase consists of planning the deployment, monitoring, and maintenance.

The Figure 3.36 Cross-Industry Standard Process for Data Mining (CRISP-DM) process model (Chapman et al., 2000). The sequence of the phases is not strict. Moving back and forth between different phases is always required. It depends on the outcome of each phase, which phase, or which task of a phase, must be performed next. The arrows indicate the most important and frequent dependencies between phases.

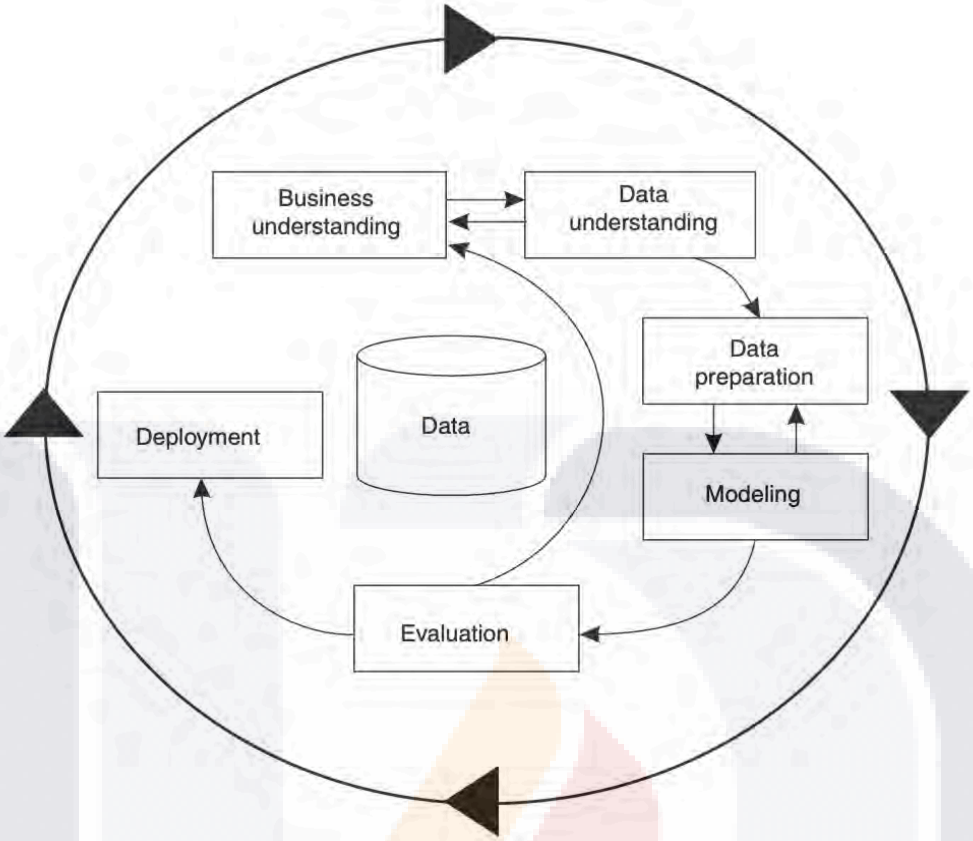
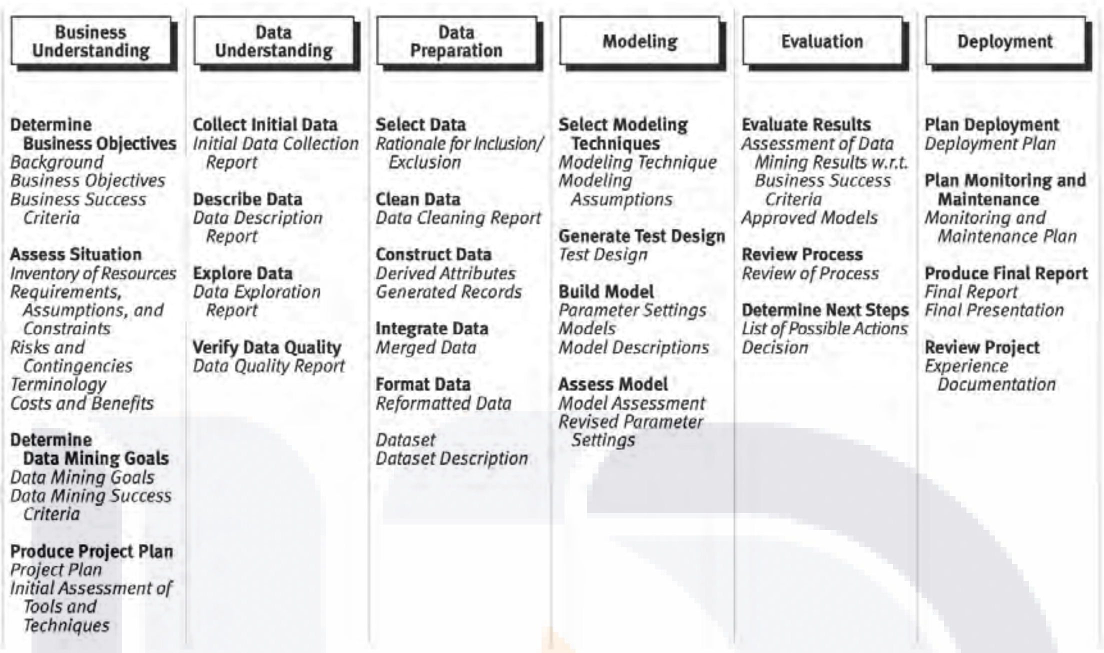


Figure 3.36 Cross-Industry Standard Process for Data Mining (CRISP-DM) process model (Chapman et al., 2000).

The Figure (3.37 Generic tasks and results of the CRISP-DM reference model) presents a scheme of phases accompanied by tasks and results, where we know the tasks and artifacts of this methodology.



Business Understanding

Determine Business Objectives
Background
Business Objectives
Business Success Criteria

Assess Situation
Inventory of Resources
Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques

Data Understanding

Collect Initial Data
Initial Data Collection Report

Describe Data
Data Description Report

Explore Data
Data Exploration Report

Verify Data Quality
Data Quality Report

Data Preparation

Select Data
Rationale for Inclusion/ Exclusion

Clean Data
Data Cleaning Report

Construct Data
Derived Attributes
Generated Records

Integrate Data
Merged Data

Format Data
Reformatted Data

Dataset
Dataset Description

Modeling

Select Modeling Techniques
Modeling Technique
Modeling Assumptions

Generate Test Design
Test Design

Build Model
Parameter Settings
Models
Model Descriptions

Assess Model
Model Assessment
Revised Parameter Settings

Evaluation

Evaluate Results
Assessment of Data Mining Results w.r.t. Business Success Criteria
Approved Models

Review Process
Review of Process

Determine Next Steps
List of Possible Actions
Decision

Deployment

Plan Deployment
Deployment Plan

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan

Produce Final Report
Final Report
Final Presentation

Review Project
Experience Documentation

Figure 3.37 Generic tasks and results of the CRISP-DM reference model (Chapman et al., 2000).

As we can see, the main difference of CRISP-DM concerning KDD and SEMMA is that this methodology is much more complete and clearly defines the phases, activities, and artifacts that the methodology has, however this methodology does not correctly define the roles since it does not mention roles in any section in the same way as the other two methodologies analyzed KDD and SEMMA.

Table 3.20 (Summary of KDD, CRISP-DM and SEMMA Processes (Shafique & Qaiser, 2014)) show us a comparison between the three methodologies, the first table shows us a comparison based on the number of steps that each of the methodologies follows to carry out Data Mining and obtain value from the data we have. On the other hand, the second table shows a comparison of the three methodologies concerning the phases, activities, roles, and artifacts of each one of them.

Table 3.20 Summary of KDD, CRISP-DM and SEMMA Processes (Shafique & Qaiser, 2014).

Data Mining Process Model	KDD	SEMMA	CRISP-DM
No. of Steps	9	5	6
Name of Steps	Developing and Understanding of the Application	------------	Business Understanding
	Creating a Target Data Set	Sample	Data Understanding
	Data Cleaning and Pre-processing	Explore	
	Data Transformation	Modify	Data Preparation
	Choosing the suitable Data Mining Task	Model	Modeling
	Choosing the suitable Data Modeling Model Mining Algorithm		
	Employing Data Mining Algorithm		
	Interpreting Mined Patterns	Assessment	Evaluation
	Using Discovered Knowledge	------------	Deployment

DATA INNOVATION

There is a significant divergence between traditional data analysis and big data analysis. The project, objective, scope, and functional requirements in traditional data analysis or software projects are considered relatively more explicit than those in big data projects.

Currently, big data projects fail to achieve a high completion rate: the completion rate stands at approximately 55 percent, whereas the incomplete rate for general software projects is around 38 percent. The difference can be attributed to an inaccurate scope and the value of the outcomes (Lin et al. 2018).

Considering the impact of variety in big data, an appropriate process was designed for big data projects using inductive analysis and comparison.

For general data analysis projects or software projects, the defined goals or functions of the project serve as the requirements for specification, followed by the work plan and implementation. However, the variety in big data projects makes it impossible to fully verify the results of information applications. The objective, according to the variety, should involve innovative data processing and corresponding approaches (Lin et al. 2018).

When working with data innovation, it is recommended not to be overly constrained by certain factors such as goal orientation, data readability, data integrity, and information quality. The implementation of data innovation should seek any possible data trends and relationships through different perspectives, ranges, properties, and dimensions, or other scientific techniques such as statistics and multivariate methods (Lin et al., 2018).

There are four main elements involved in designing an appropriate process for big data projects: one characteristic, one concept, and two processes. The characteristic refers to data variety, the concept is data innovation, and the processes are software engineering and data analysis.

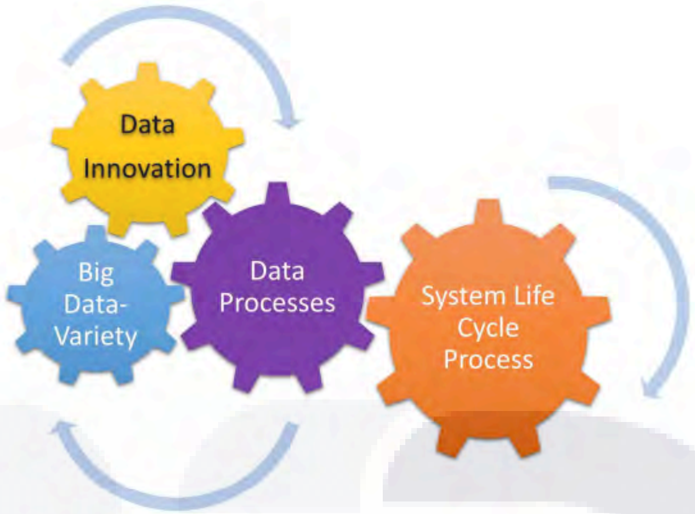


Figure 3.38 Major elements of the big data project lifecycle process.

To deal with the variety in big data projects, it is recommended to establish the processes described below (Lin et al. 2018):

- Value of data, outcome, and innovation process (according process). It is considered risky to commit to big data project contracts by defining only the project goals without including the data scope. Project risks can be mitigated by first defining and controlling the data scope.
- Domain specialist resource management process (organizational project enablement process). Due to the variety of big data, managing interdisciplinary personnel is likely to become more complex. There should be a set of separate processes in place to be reviewed by a specialist, with resources coming from client-side or external experts.
- Data inventory process (data process). Once the data is collected, a data inventory is conducted for management purposes. The data inventory is expected to contain information such as data format, type, source, quantity, timestamp, states, renewal period, owner, etc.
- Data requirements analysis process (data process). This is carried out to understand and define the necessary data to achieve the expected outcomes and value.

- Data cleansing process (data process). To prevent the loss of data variety, it is recommended to clean the data after the data innovation process has been completed.

It is recommended that data processing serve as an independent process from project processes and technical processes. Additionally, data processes should include the following processes: data collection, data inventory, data requirements analysis, data integration, data verification, data analysis, data modeling, data simulation, data prediction, data innovation, data validation, data cleansing, and data maintenance.

To deal with data processes, it is recommended to establish the following technical processes (Lin et al. 2018).

- Data automation and tracking process (technical process). These processes are primarily concerned with establishing a mechanism, through technical approaches, to collect and monitor data automatically and continuously. The mechanism is expected to prevent data source anomalies so that only accurate results are obtained.
- Data visualization process (technical process). Data visualization deserves significant emphasis as it is considered a crucial part of a big data project. It is also important to ensure that the results can be integrated with a visual tool or platform.
- Data-driven decision support process (technical process). Most data projects are applied in supporting decision-making for businesses or government entities. This process primarily deals with the analysis and application of the results to provide actionable insights and support informed decision-making.

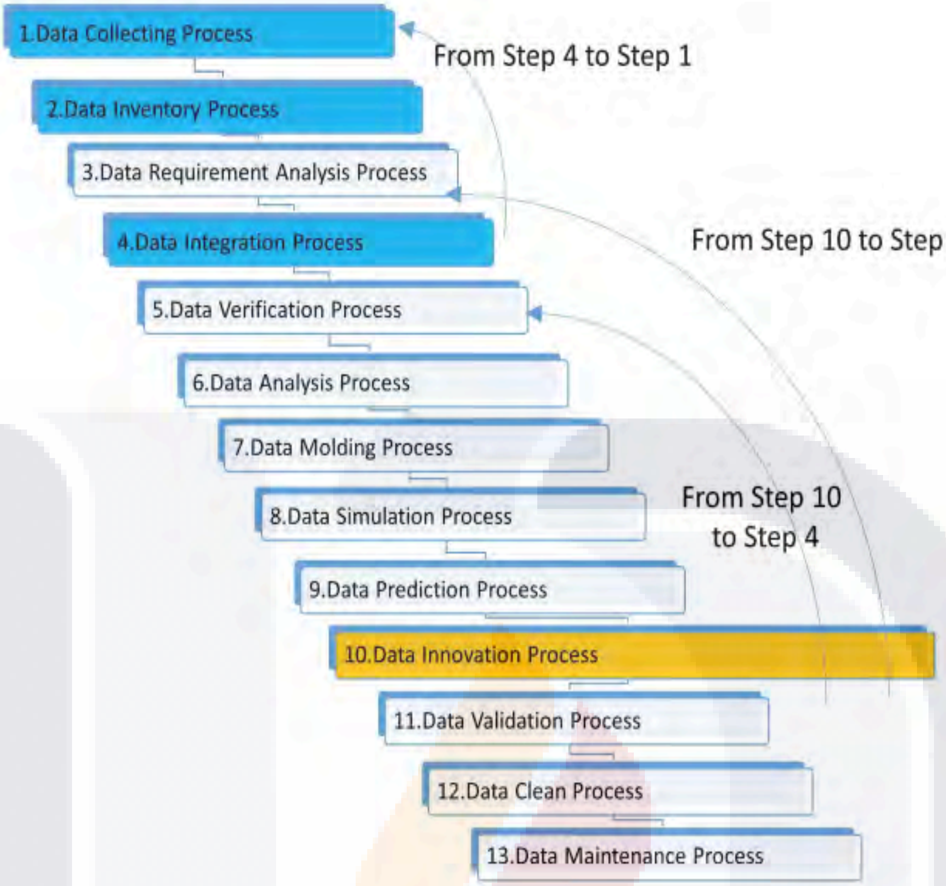

1.Data Collecting Process
2.Data Inventory Process
3.Data Requirement Analysis Process
4.Data Integration Process
5.Data Verification Process
6.Data Analysis Process
7.Data Molding Process
8.Data Simulation Process
9.Data Prediction Process
10.Data Innovation Process
11.Data Validation Process
12.Data Clean Process
13.Data Maintenance Process
From Step 4 to Step 1
From Step 10 to Step 2
From Step 10 to Step 4

Figure 3.39 Data Innovation process and Cycle (Lin, et al. 2018).

The processes were used, together with ISO/IEC 15288:2008, with which life cycle processes for big data projects were designed, which are shown in Table 3.21.

Table 3.21 Major elements of the big data project lifecycle process (Lin, et al., 2018).

Agreement processes	Project processes	Data processes	Technical processes
Data value, result, and innovation process	Project planning process	Data collecting process	Stakeholder requirement definition process
Acquisition process	Project assessment and control process	Data inventory process	Requirement analysis process
Supply process	Decision management process	Data requirement analysis process	Architectural design process
	Risk management process	Data integration process	Data automation and monitoring process
	Configuration management process	Data verification process	Data visualization process
	Information management process	Data analysis process	Data decision support process
	Measurement process	Data modeling process	Implementation process
Organizational project-enabling process		Data simulation process	Integration process
Lifecycle mode management process		Data prediction process	Verification process
Infrastructure management process		Data innovation process	Transition process
Project portfolio management process		Data validation process	Validation process
Domain specialist resource management process		Data cleaning process	Operation process
Human resource management process		Data maintenance process	Maintenance process
Quality management process			Disposal process

Table 3.22 Comparison of traditional methodologies.

Phase workflow components categories	KDD: Knowledge Discovery in Databases (Fayyad et al., 1996)	SEMMA: Sample, Explore, Modify, Model and Assess (SAS Institute Inc., 2017).	CRISP-DM: Cross-Industry Standard Process for Dara Mining (Chapman, P et al., 2000)	DATA INNOVATION (Lin, et al. 2018).
Phases	1. Selection 2. Preprocessing 3. Transformation 4. Data Mining 5. Interpretation / Evaluation	1. Sample 2. Explore 3. Modify 4. Model 5. Assess	1. Business Understanding 2. Data Understanding 3. Data Preparation 4. Modeling 5. Evaluation 6. Deployment	1. Agreement Process 2. Project Process 3. Data Process 4. Technical Process
Roles	No reported	No reported	No reported	• Project Process User • Manager • Operator User • Developer Maintainer • Acquirer Supplier

Activities	**Phase.1 Selection:** {Learning the application domain, Creating a target dataset.}	**Phase.1 Sample:** {Append Node, Data partition node, File import node, Filternode, Input data node, Merge node, Sample node.}	**Phase.1 Business Understanding:** {Determine Business Objectives, Assess Situation, Determine Data Mining Goals, Produce Project Plan.}	**Phase.1.A Project Process:** {Project planning process.}
				Phase.1.B Data process: {Data collecting process, Data inventory process.}
				Phase.1.C Technical process: {Stakeholder requirement definition process.}

	Phase.2 Preprocessing: {Data cleaning and preprocessing.}	**Phase.2 Explore:** {Association node, Cluster node, DMDB node, Graph explore node, Link analysis node, Market basket node, Multiplot node, Path analysis node, SOM/kohonen node, StatExplore node, Variable, Clustering node, Variable selection.}	**Phase.2 Data Understanding:** {Collect Initial Data, Describe Data, Explore Data, Verify Data Quality.}	
	Phase.3 Transformation: {Data reduction and projection.}	**Phase.3 Modify:** {Drop node, impute node, Interactive binning node, Principal components node, Replacement node, Rules builder node, Transform variables node.}	**Phase.3 Data Preparation:** {Select Data, Clean Data, Construct Data, Integrate Data, Format Data.}	**Phase.2.A Project Process:** {Project assessment and control process, Decision management process, Risk management Process, Configuration management process, Information management process, Measurement process.} **Phase.2.B Data process:**

	Phase.4 Data Mining: {Choosing the function of data mining, Choosing the data mining algorithm(s), Data mining.}	**Phase.4 Model:** {AutoNeural Node, Decision Tree Node, Domine Regression Node, DMNeural Node, Ensemble Node, Gradient Boosting Node, Interactive Decision Tree Application, LARs Node, Memory-Based Reasoning (MBR) Node, Model Import Node, Neural Network Node: Reference, Neural Networking Node: Usage, Partial Least Squares Node, Regression Node, Rule Induction Node, TwoStage Node.}	**Phase.4 Conceptual Modeling:** {Select Modeling Techniques, Generate Test Desing, Build Model, Assess Model.}	{Data requirement analysis process, Data integration process, Data verification process, Data analysis process, Data modeling process.} **Phase.2.C Technical process:** {Requirement analysis process, Architectural design process, Data automation and monitoring process, Data visualization process, Data decision support process.}

			Phase.5 Evaluation: {Evaluate Results, Review Process, Determine Next Steps.}	**Phase.3.A Agreement Processes:** {Data value, result, and innovation process, Acquisition process, Supply process.}
				Phase.3.B Project Process: {Project assessment and control process, Decision management process, Risk management process, Configuration management process, Information management process, Measurement process.}
				Phase.3.C Data Process: {Data simulation process, Data prediction process, Data innovation process, Data validation process, Data cleaning process, Data maintenance process.}
				Phase.3.D Technical Process: {Implementation Process, Integration process, Verification process, Transition process, Validation process, Operation process, Maintenance process, Disposal process.}

	Phase 5. Interpretation / Evaluation: {Interpretation, Using discovered knowledge.}	**Phase.5 Assess:** {Cutoff, Decisions node, Model comparison node, Score node, Segment profile node.}	**Phase.6 Deployment:** {Plan Deployment, Plan Monitoring and Maintenance, Produce Final Report, Review Project.}	
Artifacts	**Phase.1 Selection:** {Data, Target Data.}	No reported	**Phase.1 Business Understanding:** {Background, Business Objectives, Business Success Criteria, Inventory of Resources, Requirements, Assumptions, and Constraints, Risks and Contingencies, Terminology, Costs and Benefits, Data Mining Goals, Data Mining Success Criteria, Project Plan, Initial Assessment of Tools and Techniques.}	No reported

	Phase.2 Preprocessing: {Preprocessed.}		**Phase.2 Data Understanding:** {Initial Data Collection Report, Data Description Report, Data Exploration Report, Data Quality Report.}	
	Phase.3 Transformation: {Transformed Data.}		**Phase.3 Data Preparation:** {Rationale for Inclusion/Exclusion, Data Cleaning Report, Derived Attributes, Generated Records, Merged Data, Reformatted Data, Dataset, Dataset Description.}	
	Phase.4 Data Mining: {Patterns.}		**Phase.4 Conceptual Modeling:** {Modeling Technique, Modeling Assumptions, Test Design, Parameter Settings, Models, Model Descriptions, Model Assessment, Revised Parameter Settings.}	
			Phase.5 Evaluation: {Assessment of Data Mining Results, Approved Models, Review of Process, List of Possible Actions, Decision.}	
	Phase 5. Interpretation / Evaluation: {Knowledge.}		**Phase.6 Deployment:** {Deployment Plan, Monitoring and Maintenance Plan, Final Report, Final Presentation, Experience Documentation.}	

3.1.3.6 REVIEW OF THE MAIN AGILE ANALYTICS/DATA SCIENCE SDM

In recent decades, the capacity of electronic devices and sensors, in addition to the use of social networks and the ability to store and exchange this data, have dramatically increased the opportunities to extract knowledge through data mining projects (Martinez-Plumed et al., 2019). The diversity of data has increased in origin, format, and modalities, as has the variety of techniques coming from machine learning, data management, visualization, causal inference, and other areas (Martinez-Plumed et al., 2019). In other words, not only has the nature of the data changed, but also the processes for extracting value from it.

The need for fast delivery of business intelligence has increased in the last 5 years due to the demand for real-time data analysis (Halper, 2015). The Internet of Things (IoT), where data collection is built into devices, contributes to this demand for more up-to-date data. Equipment failure monitoring will be possible with data that is seconds old versus data that is hours or days old (Halper, 2015).

All this makes Big Data, Data Science, and Analytics more relevant for today's companies, since with these practices, companies can generate competitive advantages. In turn, with the data landscape changing so quickly, big data projects, Data Science, and Analytics, the methodologies used are also changing.

In 2019, VentureBeat revealed that 87% of data science projects never make it to production (VentureBeat, 2019), and a New Vantage survey reported that for 77% of companies, the adoption of big data and artificial intelligence (AI) initiatives continues to represent a great challenge (New Vantage, 2019). All this due to the lack of use of methodologies for the development of this type of project, in a survey carried out in 2018 to professionals from both the industry and non-profit organizations, 82% of the respondents did not follow an explicit methodology of process for developing data science projects, but 85% of respondents believed that using an improved and more consistent process would produce more consistent and effective data science projects (Saltz et al., 2018).

All this indicates the lack of clear methodologies for the development of Data Science-type projects, since, according to a survey carried out in 2014 by KDnuggets, the main methodology used by 43% of those surveyed was CRISP-DM.

This Methodology has been considerably the most used for analysis, data mining, and data science projects (Piatetsky, 2014). Despite its popularity, CRISP-DM was created in the mid-1990s and has not been revised since its inception. In turn, there are some other methodologies for this type of project, but they are not clear when defining their roles, their activities, or their artifacts. There is little research on the application of agile principles for this type of project; however, the available research suggests that Agile would align well, but would need to be "short-cycle agile," suggesting faster results are needed (Davenport, 2014). Agile methodologies also align well with Big Data, where little time is spent defining requirements up front and the emphasis is on developing small projects quickly. Agile methodologies will align well with iterative discovery and validation that support prescriptive and predictive analytics (Ambler & Lines, 2016).

Organizations are focusing more on prescriptive and predictive analytics using machine learning and rapid analytics through visualization. Rapid analysis refers to the ability to rapidly acquire and visualize data (Halper, 2015; Jarr, 2015). Table 3.23 Traditional BI vs Rapid analysis with Big Data (Halper, 2015; Jarr, 2015) illustrates the different characteristics between traditional BI and rapid analysis with Big Data.

Table 3.23 Traditional BI vs Rapid analysis with Big Data (Halper, 2015; Jarr, 2015).

Criteria	Traditional Business Intelligence	Fast Analytics with Big Data
Analytics Type	Descriptive, Predictive	Predictive, Prescriptive
Analytics Objectives	Decision Support, Performance Management	Drive the Business
Data Type	Structured and defined	Unstructured, Undefined
Data Age	24 hours	Minutes

Data science includes techniques developed in some traditional fields like artificial intelligence, statistics, or machine learning, data science. Therefore, it is essential to use a methodology that can contribute to improving the results of knowledge creation. In this context, we will address some of the different agile methodologies for Big Data, Data Science, and Analytics projects that currently exist.

TDSP (Team Data Science Process)

The Team Data Science Process (TDSP) is an agile and iterative data science methodology to efficiently deliver predictive analytics solutions and intelligent applications (Microsoft, 2107). TDSP helps improve team collaboration and learning by suggesting how team roles work best together. Its main objective is to help companies take full advantage of the benefits of their analysis program. It is very well documented and provides several tools and utilities that make it easy to use.

TDSP provides a life cycle to structure the development of your projects. The TDSP project life cycle is like CRISP-DM and includes five iterative stages: Business Understanding, Data Acquisition and Understanding, Modeling, Implementation, and Customer Acceptance. It is an iterative and cyclical process.

This lifecycle has been designed for data science projects that focus on applications or learning models, more focused on predictive analytics. Exploratory data science projects or impromptu analytics projects can also benefit from using this process, but in such cases, some of the steps may not be necessary (Microsoft, 2107).

TDSP addresses the weakness of CRISP-DM's lack of role definition by defining four distinct roles (solution architect, project manager, data scientist, and project leader) and their responsibilities during each phase of the project life cycle (Microsoft, 2107).

These roles are very well defined from a project management perspective, and the team works under agile methodologies, which improve collaboration and coordination (Microsoft, 2107). Their responsibilities regarding the creation, execution, and development of the project are clear (Microsoft, 2107).

TDSP is one of the best documented methodologies that exist for this type of project, since it specifies roles, tasks, and artifacts, as well as being a methodology that can be easily combined with other existing methodologies such as CRISP-DM or KDD. Unfortunately, TDSP relies heavily on Microsoft services and policies, and this complicates wider use, as all documentation provided by Microsoft for this methodology only mentions and suggests the use of Microsoft tools. TDSP provides a life cycle to structure the development of its projects. The TDSP project life cycle is like CRISP-DM and includes five iterative stages: commercial understanding, data acquisition and understanding, modeling, implementation, and customer acceptance; in fact, it is an iterative and cyclic process (Microsoft, 2107).

In the following statements, we outline each phase briefly (Microsoft, 2107):

- **Business Understanding:** Initially, a question that describes the problem objectives should be defined clearly and explicitly. The relevant predictive model and required data source/s must also be identified in this step.
- **Data Acquisition and Understanding:** Data collection starts in this phase by transferring data into the target location to be utilized by analytic operations. The raw data needs to be cleaned. Also, either incomplete or incorrect values should be identified. Data summarization and visualization might help to find the required cleaning procedures. Data visualization could also help to measure if data features and the collected amount of data are adequate over time. At the end of this stage, it might be necessary to go back to the first step for more data collection.
- **Modeling:** Feature engineering and model training are two elements of this phase. Feature engineering provides attributes and data features that are required for the machine learning algorithm. Algorithm selection, model creation, and predictive model evaluation are also subcomponents of this step. Collected data should be divided into training and testing datasets to train and evaluate the machine learning model. It is important to employ different algorithms and parameters to find the best suitable solution to support the problem.
- **Deployment:** Predictive model and data pipeline need to be produced in this step. It could be either a real-time or a batch analysis model, depending on the required application. The final data product should be accredited by the customer.

- **Customer Acceptance:** The final phase is customer acceptance, which should be performed by confirming the data pipeline, predictive model, and product deployment.

Figure 3.40 TDSP Lifecycle provides an overview of the TDSP lifecycle, mentioning the 5 stages of its lifecycle as well as some of its tasks and artifacts.

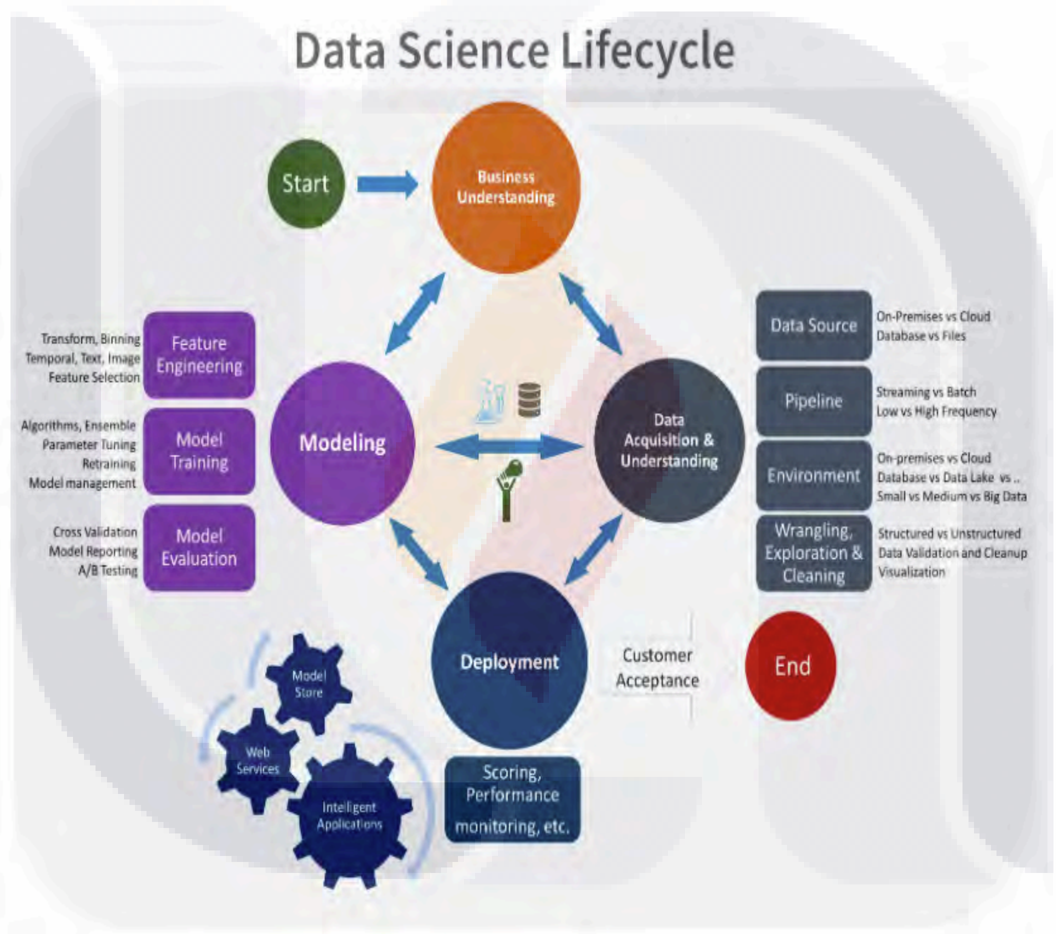


Figure 3.40 TDSP Lifecycle (Microsoft, 2017).

In turn, Table 3.24 TDSP Roles, activities, and artifacts compiles the life cycle phases, roles, activities, and artifacts that TDSP has.

Table 3.24 TDSP Roles, activities and artifacts.

Phases	Activities	Roles	Artefacts
Business Understanding	• Define objectives • Identify data source	Project Lead, Project manager	• Charter Document • Data Source • Data Dictionaries
Data Acquisition and Understanding	• Ingest the data • Explore the data • Set up a data pipeline	Project Lead, Data Scientist, Solution architecture	• Data Quality Report • Solution Architecture • Checkpoint Decision
Modeling	• Feature engineering • Model training • Model evaluation	Data Scientist, Solution Architecture, Application developer, Data engineer	• Feature Sets • Model Report • Checkpoint Decision
Deployment	• Operationalize a model	Data Scientist, Solution Architecture, Application developer, Data engineer	• A status dashboard that displays the system health and key metrics • A final modeling report with deployment details • A final solution architecture document
Customer acceptance	• System validation • Project hand-off	Project Lead, Project manager, Data Scientist	• Exit report of the project for the customer

Analytics Solutions Unified Method (ASUM-DM)

IBM defines ASUM-DM (Analytics Solutions Unified Method for Data Mining and Predictive Analytics) as an iterative process for conducting a comprehensive implementation of the lifecycle of a predictive analytics or data mining project. It was created based on the CRISP-DM methodology, which has been expanded and improved to accelerate the time to value and reduce risk by establishing coherent approaches and processes that increase implementation efficiency (IBM, 2015).

The ASUM-DM methodology consists of 5 phases: analyze, design, configure and build, implement, and operate and optimize. However, the methodology combines three phases into one (analyze, design, configure, and build) due to the iterative nature of data analysis projects (IBM, 2015).

ASUM-DM is based on the CRISP-DM methodology but in a broader and refined manner. The activities of CRISP-DM and the data extraction cycle are retained, but the "implementation" phase is strengthened, which is considered one of the weaker points of CRISP-DM. Additionally, ASUM-DM adds structured steps, development activities, roles and responsibilities, templates, and guidelines that enhance the methodology (IBM, 2015).

One of the important points that ASUM-DM improves with respect to CRISP-DM is the implementation of roles that have different responsibilities and perform different tasks to comply with the provisions of the methodology. The different roles and a brief description of them are mentioned below. Your responsibilities.

The method Work Breakdown Structure (WBS) already incorporates adequate project management elements, but an additional optional Project Management Process has been added here for supplemental use when needed (IBM, 2015).

The ASUM-DM Life Cycle, shown in Figure 3.41, illustrates the phases and how they interact with each other. It is worth noting that the project management part is managed independently from the methodology, as mentioned earlier, and is considered an optional phase within the methodology. Additionally, Table 3.25 presents the roles along with descriptions of their responsibilities and the activities they perform. Lastly, Table 3.26 provides a list of activities along with brief descriptions.

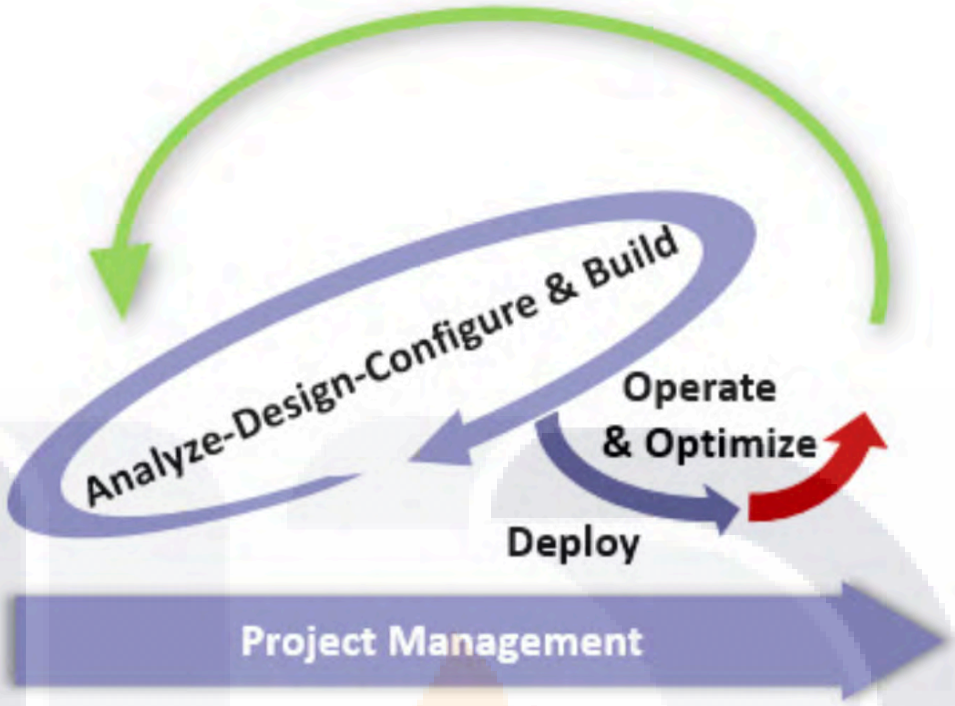


Figure 3.41 ASUM-DM Life Cycle (IBM, 2015).

Table 3.25 ASUM-DM Roles (IBM, 2015).

Roles	Descripción
Client Application Administrator	Responsible for the maintenance, data management, and administration of the solution.
Client Business Sponsor	• Approves project scope. • Ultimate owner of the project and key decision maker. • Demonstrates sponsorship through active and visible participation (i.e., influences within the organization to solicit project support). • Strategically direct and support the overall project and set priorities. • Proactively identify and resolve cross-functional & divisional issues and communicate decisions / reasoning in a timely fashion. • Provides consent on key project deliverables. • Provides input to important project decisions. • Participates in creating an environment that encourages open two-way communication.
Client Data Analyst	Assess source data quality and prepare data-cleansing specifications for the ETL process.
Client Database Administrator	Responsible for the design, load, monitor and tune of the SPSS target databases.
Client Key System Users	• Act as the main solution users and builders post-implementation. • Design UAT testing strategies.

	• Develop testing scripts. • Provide test data. • Carry out UAT. • Assist in the execution of other tests.
Client Network Administrator	Maintains the network environment.
Client Project Manager	• Liaise with the IBM project manager and ensures efficient utilisation of time and resources and progress the project on a day-to day basis. • Lead client resources and participating in elements of project management.
Client Security Administrator	Ensures that security requirements are defined and that security features are tested across all tools and databases.
Client Stakeholders	Handle limited responsibilities on the SPSS project, such as reviewing and ratifying the cross-organizational standards and business rules the SPSS project team uses or develops.
Client Subject Matter Expert	Provide business knowledge about data, processes, and requirements.
Client Support Manager	• Assists users with functionality issues, technical issues, and troubleshooting. • Acts as the contact with IBM support. • Ensures that the solution is running efficiently post implementation.
Client Tool Administrator	Assist with the installation and maintenance of the IBM SPSS software.
Data Miner/Data Scientist	• Responsible for understanding business, understanding data, preparing data, building models, and evaluating models. • Responsible jointly with the Enterprise Architect for testing the solution in non-Analytical environments and deployment of the solution.
Enterprise Architect	• Responsible for designing and validating infrastructure. • Responsible for the installation and configuration of the IBM SPSS software. • Responsible for integration of the solution with other systems. • Responsible jointly with the Data Scientist testing the solution. in non-Analytical environments and deployment of the solution.
Project Manager	• Responsible for the overall project planning and coordination. • Own the project deliverables and is responsible for day-to-day project management. • Anticipate project deviations proactively and be responsible for taking immediate corrective actions. • Provide administrative, functional direction and support to the project team. • Set project standards and milestones and monitor work against those standards to ensure completion on-time. • Monitor project costs vs. budget and take corrective actions to ensure project completion within budget. • Identify key issues and communicate key team decisions and reasoning in a timely fashion. • Identify required project resources. • Respond to project team members' concerns and work with problem resources. • Organize the team resources in an effective and efficient manner.

	• Manage and communicate scope and potential scope changes. • Manage, monitor and communicate risks. • Develop and maintain processes to identify and resolve integration issues.
SPSS Project Manager	• Responsible for the overall project planning and coordination. • Own the project deliverables and is responsible for day-to-day project management. • Anticipate project deviations proactively and be responsible for taking immediate corrective actions. • Provide administrative, functional direction and support to the project team. • Set project standards and milestones and monitor work against those standards to ensure completion on-time. • Monitor project costs vs. budget and take corrective actions to ensure project completion within budget. • Identify key issues and communicate key team decisions and reasoning in a timely fashion. • Identify required project resources. • Respond to project team members' concerns and work with problem resources. • Organize the team resources in an effective and efficient manner. • Manage and communicate scope and potential scope changes. • Manage, monitor and communicate risks. • Develop and maintain processes to identify and resolve integration issues.

Table 3.26 ASUM-DM Activities (IBM, 2015).

Activitie	Description
Prepare for Implementation	This is where a hand over meeting from Sales takes place where project details and customer expectations are reviewed and resources for the project are identified.
Conduct Readiness Assessment	Assess how ready is the customer to commence with the project.
Conduct Project Kick-off	This activity covers preparing a deck to use during the kick-off session, orienting and aligning with the IBM project team members and then conducting a kick off session to be attended by IBM and the client.
Understand Business	The purpose of this activity is to understand the project objectives and requirements from a business perspective, then convert this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.
Understand Data	This activity involves taking a closer look at the data available for mining. This step is critical in avoiding unexpected problems during the next activity—data preparation—which is typically the longest part of a project. Data understanding involves accessing the data and exploring it. This enables you to determine the quality of the data and describe the results of these steps in the project documentation.
Design and Validate Infrastructure	Design the environments architecture and the authentication and authorization strategies.
Set up Environments	Set up the Analytical, QA, and Production environments as per design and requirements onsite or on cloud. Delete which ever is not applicable.

Prepare Data	Data preparation is one of the most important and often time-consuming aspects of data mining. In fact, it is estimated that data preparation usually takes 50-70% of a project's time and effort. It is highly dependent on the "understand data" and "understand business" activities, so devoting adequate energy to these earlier activities can minimize this overhead, but you still need to expend a good amount of effort preparing and packaging the data for mining.
Build Model	Modeling is usually conducted in multiple iterations. Typically, data miners run several models using the default parameters and then fine-tune the parameters or revert to the Prepare Data activity for manipulations required by their model of choice. It is rare for an organization's data mining question to be answered satisfactorily with a single model and a single execution.
Evaluate Model	Assess the models using the business success criteria.
Conduct Analytical Knowledge Transfer	Orient and educate the client's team on the Data Mining /Predictive Analytics process which has been set up.
Define Deployment Approach	Determine and describe how the solution is going to be rolled over to all users.
Design Operational Testing Strategy	Discuss and agree with the project team the testing strategy for the Operational Stream of the project and how The Performance, System, and UAT tests will be conducted and run and create tests plans that will be updated as the project progresses.
Validate and Test in QA Environment	Ensure that all the correct steps so far has been taken, test the solution in QA environment, and make production deployment decision based on validated steps and successful testing.
Conduct Operational Knowledge Transfer	Orient and educate the client on the non-analytical aspect of the solution so that the solution could run effectively and efficiently once IBM leaves site.
Prepare for Ongoing Maintenance	Ensure all of the supporting functions and activities are in place prior to deploying the solution.
Deploy Solution	Move the solution into the production environment as per Deployment Plan, and validate that the production environment is configured properly.
Transit to IBM Support	Transit solution from project team to IBM Support.
Launch	Go live with the solution and communicate to the end-user community and stakeholders that the solution is live and review the launch to gather lessons-learned and successes.
Prepare for Project Closure	Prepare and execute tasks to close the project.
Monitor Model	Monitor the results of the deployed model(s) continuously to ensure their accuracy and that they still satisfy the data mining goals of the organization and business objectives.
Operate, Optimize and Imrove System	Conduct Post-launch activities to keep the system operating properly.
Support User Community	Conduct post-launch activities to support the end-user community.
Manage Infrastructure	Conduct post-launch activities to manage and maintain the infrastructure.
Govern System Lifecycle Program	Conduct Post-launch activities to manage the life-cycle of the solution.

Data Driven Scrum (DDS)

Data Driven Scrum (DDS) is an agile framework specifically designed for Data Science projects, aiming to enhance collaboration and communication within a Data Science team. This agile framework was developed to address the lack of adaptation of approaches such as Scrum and Kanban to Data Science projects (Saltz, 2022).

To achieve this, DDS focuses on achieving three key agility concepts, which allow a Data Science team to obtain agility benefits within a project (Saltz, 2022).

- Agile aims to be a sequence of iterative cycles of experimentation and adaptation.
- The objective of each cycle should be to have an idea or experiment in mind, which is then built, observed, and analyzed. Once analyzed, the next idea or experiment is created.
- Moving from an initial idea through implementation and analysis of results should form the basis for an iteration. The completion of the empirical process should mark the end of that iteration (not a predetermined number of hours elapsed).

DDS mentions the implementation of 4 phases in its workflow. Firstly, teams brainstorm possible questions to answer or experiments to conduct. Then, the team prioritizes these questions, selecting the highest-priority item to work on. This includes identifying the data to be used and the models that need to be created. Once this is done, the team collectively interprets the results of their work. Lastly, based on the results, the team implements them and prioritizes future work (Saltz, 2022).

In Figure 3.42, you can see the DDS workflow, where we can see the 4 phases mentioned in the previous description.

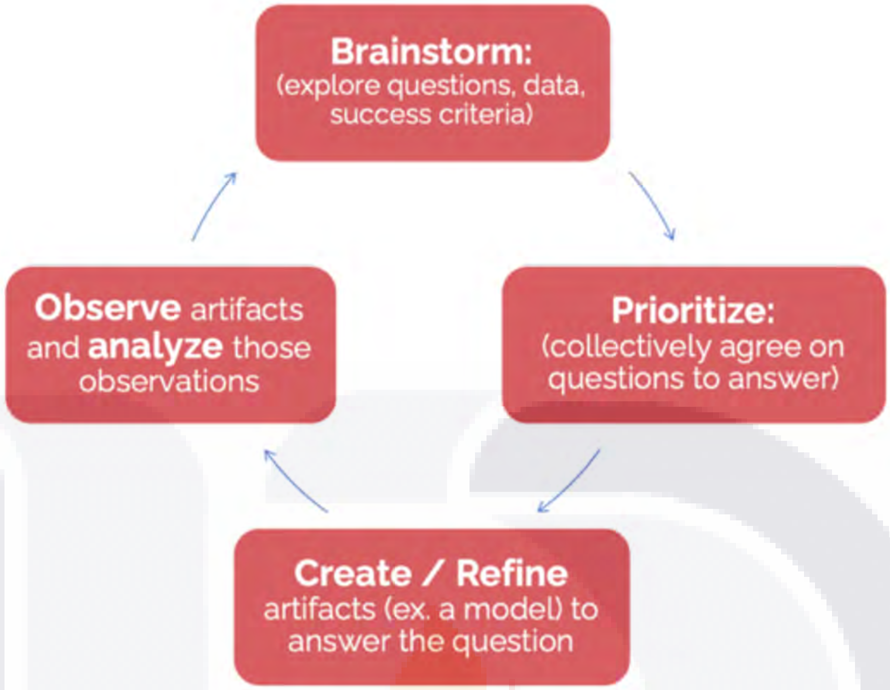

Brainstorm:
(explore questions, data, success criteria)
Prioritize:
(collectively agree on questions to answer)
Create / Refine
artifacts (ex. a model) to answer the question
Observe artifacts and analyze those observations

Figure 3.42 DDS A High-Level Flow of Work (Saltz, 2022).

DDS, like other methodologies, defines different roles, activities, and artifacts; each of these is described below, where we can observe many similarities with the Scrum methodology.

Table 3.27 DSS Roles (Saltz, 2022).

Roles	Descripción
Product Owner	The person who decides on the product increments, prioritizes which features and functionalities to build, the order in which they are built, and which aspects of them to observe and analyze is the Product Owner.
Process Expert	The role described, responsible for acting as a coach, facilitator, and impediment remover, helping the team understand and adopt the values and practices of DDS, is indeed similar to that of a Scrum Master. The Scrum Master in Scrum methodology plays a similar role in guiding and supporting the team, ensuring the proper implementation of the agile practices, and removing any obstacles that may hinder their progress.
DDS Team Members	They are typically groups of three to nine people, composed of a cross-functional collection of members (e.g., data scientists, software engineers, among others), who have all the skills to create the necessary artifacts (i.e., to design, build, test, and deploy the desired product).

Table 3.28 DSS Activities (Saltz, 2022).

Activities	Descripción
Backlog Refinement	The team and the product owner allocate time to evaluate the items in the backlog so that they can prioritize. This evaluation includes: • A relative estimation of the value of the item when completed. • A relative estimation of the effort required to complete the item. • A relative estimation of the likelihood of success in creating the item.
Prioritization of the Backlog	The team explores the Items in their Backlog by providing high level estimates of: (1) the value of the work, (2) the amount of work (team effort), and (3) the probability of success of that work. The Product Owner, with input from the stakeholders and the other team members, is responsible for maintaining the Backlog, which evolves and changes throughout the project.
Iterations	It is a collection of one or more pending items that enable the release of a logical portion of work.
Iteration Duration	Each iteration is based on capacity (not calendar events with a time limit). It should aim to be a minimally viable set of work that generates value and should not last longer than one month. An iteration is completed when the work required to answer the question is finished (i.e., not on a specific date). An iteration is based on capacity and is the set of minimally viable items that can deliver value.
Product Increments	It is achieved within a fixed period of time through multiple iterations. These increments help teams prioritize iterations within the increment and set expectations with customers.

Table 3.29 DSS Artifacts (Saltz, 2022).

Artifacts	Descripción
Item	An element can take various forms such as "user stories," "experiments," or "testable hypotheses."
Backlog	It is a prioritized list of items (work to be prioritized).
Item Breakdown Board	It is the place where each element (Backlog) is divided into tasks. The backlog items are broken down into their component tasks before the team works on them.
Task Board	It is a visual representation of the elements currently in progress. In order for work to start on an item (i.e., for the team to start working on it), the tasks for that item are moved from the Product Backlog to the Task Board. These tasks are displayed on the Task Board, typically in the "To Do" column. The Task Board has several additional columns (at a minimum, 'To Do', 'In Progress', 'Done'), and each task flows through the board, visually showing the work being done within the team. The team strives to complete tasks on the Task Board as quickly as possible.

In DDS, there are 4 regularly occurring events (the events occur according to the calendar, not based on the completion of an iteration). These events help the team stay coordinated, aid in planning iterations through the selection of backlog items, review the outcomes of iterations through reviews (and learn for future iterations), reflect on how to improve through retrospectives, and understand potential impediments in the iteration through daily meetings.

Table 3.30 DSS Events (Saltz, 2022).

Events	Descripción
Backlog Item Selection	It occurs when the team has capacity to start a new iteration (e.g., when a previous iteration has been completed or when the ongoing iteration does not require full-time focus, usually during the "observation" phase).
Daily Meeting	It occurs every workday, where the team gathers for a 15-minute inspection and adaptation activity. The main objective of this meeting is to help the team better manage their workflow and assist any team member in overcoming any issues they may be facing.
Iteration Review	It occurs regularly and repetitively and is scheduled by the product owner. Reviews can be weekly and are based on the calendar to account for the fact that there may be multiple iterations per week. The purpose of the review is to encourage conversation about the completed functionality and the observations and analysis that the team has generated regarding the performance of the completed iterations.
Retrospective	It occurs at regular intervals (for example, once a month) and is a time for inspecting and adapting the process. With the spirit of continuous improvement, the team gathers to analyze what is working and what is not working with the current process and associated technical practices.

Figure 3.43 shows the conceptual flow of a project using the DSS methodology, where several of the functions performed by each of those involved in the project can be observed. We also note that, unlike Scrum, iterations go from 1 day to 20 days without each iteration being the same as the previous one. This is because DSS allows a logical part of the work to be done in one iteration. In other words, DDS iterations have unknown and variable-length iterations (compared to traditional Scrum sprints, which have fixed-time durations).

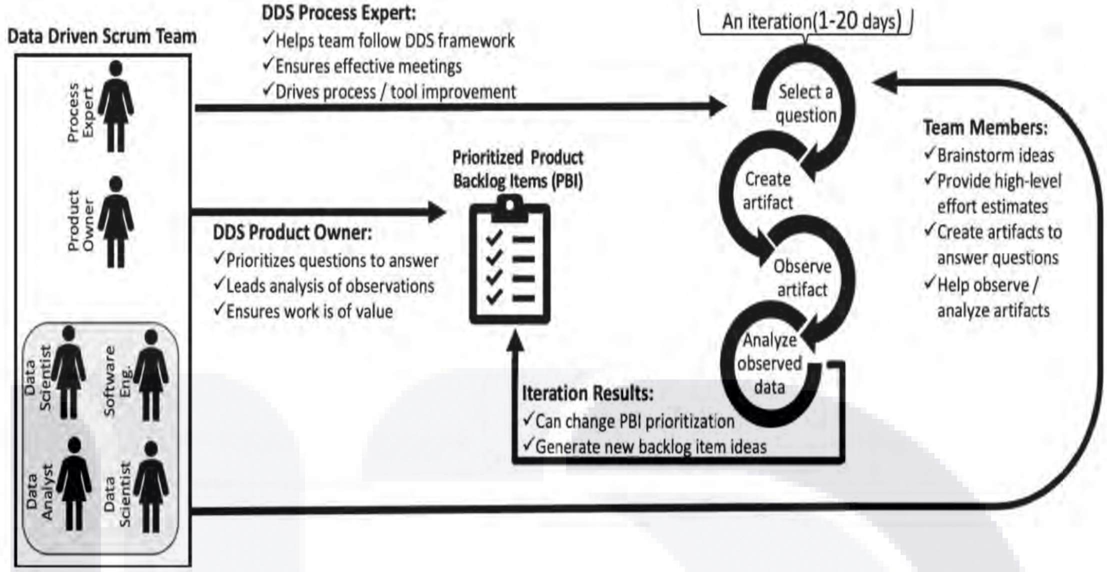

Data Driven Scrum Team
Process Expert
Product Owner
Data Scientist
Software Eng.
Data Analyst
Data Scientist
DDS Process Expert:
✓Helps team follow DDS framework
✓Ensures effective meetings
✓Drives process / tool improvement
DDS Product Owner:
✓Prioritizes questions to answer
✓Leads analysis of observations
✓Ensures work is of value
Prioritized Product Backlog Items (PBI)
An iteration(1-20 days)
Select a question
Create artifact
Observe artifact
Analyze observed data
Iteration Results:
✓Can change PBI prioritization
✓Generate new backlog item ideas
Team Members:
✓Brainstorm ideas
✓Provide high-level effort estimates
✓Create artifacts to answer questions
✓Help observe / analyze artifacts

Figure 3.43 Conceptual Flow of a DDS Project (Saltz, 2022).

Table 3.31 Comparative.

Phase workflow components categories	Team Data Science Process (TDSP) (Microsoft, 2107)	Analytics Solutions Unified Method (ASUM-DM) (IBM, 2015)	Data Driven Scrum (DSS) (Saltz, 2022)
Phases	1. Business Understanding. 2. Data Acquisition and Understanding. 3. Modeling. 4. Deployment. 5. Customer Acceptance.	1. Analyze. 1. Design. 1. Configure & build. 1. Deploy. 1. Operate & optimize.	1. Brainstorm. 2. Prioritize. 3. Create / Refine. 4. Observe & analyze.
Roles	• Group manager • Team lead • Project lead • Project individual contributors	• Client Application Administrator. • Client Business Sponsor. • Client Data Analyst. • Client Database Administrator. • Client Key System Users. • Client Network Administrator. • Client Project Manager. • Client Security Administrator. • Client Stakeholders. • Client Subject Matter Expert. • Client Support Manager. • Client Tool Administrator • Data Miner / Data Scientist. • Enterprise Architect. • Project Manager. • SPSS Project Manager.	• Product Owner. • Process Expert. • DDS Team Members.
Activities	**Phase.1 Business Understanding:** {Define Objectives, Identify data source.}	**Phase.1 Analyze- Desing- Configure & Build:** {Prepare for implementation, Conduct readiness assessment, Conduct project kick-off, Understand business, Understand business, Understand data, Design and validate infrastructure, Set up	**Phase.1 Brainstorm:** {Backlog Refinement}
	Phase.2 Data acquisition and Understanding: {Ingest the data, Explore the data, Set up a data pipeline.}		**Phase.2 Prioritize:** {Prioritization of the Backlog}
	Phase.3 Modeling: {Feature engineering, Model training, Model evaluation.}		**Phase.3 Create / Refine:** {Iterations, Iteration Duration. Product Increments.}

		environments, Prepare data, Build model, Evaluate model, Conduct analytical knowledge transfer, Define deployment approach, Design operational testing strategy, Validate and test in QA environment.}	
	Phase.4 Deployment: {Operationalize a model.}	**Phase.4 Deploy:** {Conduct operational knowledge transfer, Prepare for ongoing maintenance, Deploy solution, Transit to IBM support, Launch, Prepare for project closure.}	**Phase.4 Observe & analyze:** {Backlog Item Selection, Daily Meeting, Iteration Review, Retrospective.}
	Phase.5 Customer acceptance: {System validation, Project hand-off.}	**Phase.5 Operate & optimize:** {Monitor model, Operate, Optimize and imrove system, Support user community, Manage infrastructure, Govern system lifecycle program.}	
Artifacts	**Phase.1 Business Understanding:** {Charter Document, Data source, Data dictionaries.}	No reported	**Phase.1 Brainstorm:** {Item.}
	Phase.2 Data acquisition and Understanding: {Data quality report, Solution architecture, Checkpoint Decision.}		**Phase.2 Prioritize:** {Backlog.}
	Phase.3 Modeling: {Feature engineering, Model training, Model evaluation.}		**Phase.3 Create / Refine:** {Task board.}
	Phase.4 Deployment: {Operationalize a model.}		
	Phase.5 Customer acceptance: {System validation, Project hand-off.}		**Phase.4 Observe & analyze:** {Item breakdown board.}

3.2 ANALYSIS OF CONTRIBUTIONS AND LIMITATIONS

Table 3.32 Analysis of Contributions and Limitations.

Topic	Contributions	Opportunities of Improvement
Software Engineering	Within software engineering there are different process models, these have the objective of ordering and structuring software development, facilitating development for software engineers (Bourque et al., 2014). One of the main contributions of software engineering is the identification of roles, activities and artifacts that generate different practices and methodologies.	In 2019, VentureBeat revealed that 87% of data science projects never reach production (New Vantage, 2019). This indicates to us how there are fields of computing where software engineering is not as widely used and can be exploited in a better way.
Agile Development Paradigm	Agile development models promised higher customer satisfaction, lower defect rates, faster development times, and a solution to the changing requirements of the organizational environment (Boehm & Turner, 2003). This has caused agile processes, methodologies, and standards to be the most widely used worldwide, which allow more agile developments while preserving quality.	Both agile and plan-based approaches have a base of project characteristics where each works best and where the other will struggle (Boehm, 2002). This tells us that not all projects are convenient to be carried out with agile methodologies, projects where greater stability and high security are required will be better developed with other types of methodologies.
Big Data / Data Science / Analytics System	Currently, making the right, timely and better decisions has become fundamental, but also a matter of survival in today's complex and competitive business context (Demirkan & Delen, 2013). This need, combined with the enormous amount of data that is produced, generated the concepts of Big Data, Data Sciences and Analytics that allow us to correctly process this data for decision-making in companies.	Organizational and socio-technical challenges that arise when executing a data science project, for example: lack of clear vision, strategy and goals, biased emphasis on technical issues, lack of reproducibility and role ambiguity are among these challenges (Saltz, 2015). This is due to the low use of methodologies, processes, and standards by the developers of this type of project.
Main Analytics/Data Science SDM	The methodologies allow the data mining process to be carried out in a systematic and non-trivial way. They help organizations understand the knowledge discovery process and provide guidance for planning and executing projects.	The need for rapid delivery of business intelligence has increased in the last 5 years due to the demand for real-time data analysis (Halper, 2015). This causes the need for companies to implement Big Data, in a much faster way for decision making.
Main Agile Analytics/Data Science SDM	Agile would align well but would need to be "short-cycle agile," suggesting faster results are needed (Davenport, 2014). Agile methodologies will align well with iterative discovery and validation that support prescriptive and predictive analytics (Ambler & Lines, 2016). This indicates that the generation of agile methodologies is viable for projects such as Big Data, Data Sciences, Analytics.	A 2018 survey of professionals from both industry and nonprofit organizations, where 82% of respondents stated that they did not follow an explicit process methodology to develop data science projects (Saltz et al., 2018). In addition to the fact that there is an absence of complete or comprehensive methodologies in the literature.

4. DEVELOPMENT OF THE SOLUTION

As mentioned in Chapter 2, this research was conducted using a Design Science Research Methodology (DSRM) (Peffers et al., 2007), which is detailed in Table 2.2 and is divided into the following steps:

- DSRM step 1 - Design problem identification and motivation.
- DSRM step 2 - Definition of the Design Objectives, Design Restrictions, Design Approach, Design Theoretical Sources, and Design Components for the expected Artifact.
- DSRM step 3 - Design and development of the artifact.
- DSRM step 4 - Demonstration of the artifact (Proof of Concept).
- DSRM step 5 - Evaluation of the artifact.
- DSRM step 6 - Communication of research results.

4.1 DSRM STEP 1 DESIGN PROBLEM IDENTIFICATION AND MOTIVATION

Chapter 1 of this document contains all the detailed information for Problem Identification and its Motivation.

4.2 DSRM STEP 2 - DEFINITION OF THE DESIGN OBJECTIVES, DESIGN RESTRICTIONS, DESIGN APPROACH, DESIGN THEORETICAL SOURCES, AND DESIGN COMPONENTS FOR THE EXPECTED ARTIFACT FOR THE EXPECTED ARTIFACT: AGILE DATA SCIENCE ANALYTICS METHODOLOGY (AGILEDSA)

To create the artifact, we used an agile SDLC commonly employed in the market, such as SCRUM, and combined it with another agile SDLC, XP. When selecting this combination of methodologies, we evaluated three main criteria: (1) The research methodology guides the development of a new conceptual or physical artifact through a systematic research process. (2) The research methodology is suitable for addressing complex conceptual components to be analyzed. (3) The research methodology addresses the identified relevance of having agile IT design practices.

To establish an agile and detailed workflow, specifically a value stream to develop, build, and implement a minimum viable IT service, a heuristic design approach (DA) (Newell A, Simon HA., 1972) was employed. The heuristic DA approach is based on the iterative application of rational judgment by designers, collectively considered as a team of experts in the field. This leads to the selection of appropriate design components (DC) from theoretical design sources (DTS) and the analysis of their impacts concerning the expected design objectives (DO).

4.2.1 DEFINITION OF THE DESIGN OBJECTIVES

The expected design objectives (DO) that will be addressed in this work are:

- DO.1 The designed artifact provides an agile workflow (i.e., responsive, flexible, fast, simple, lightweight, and thoroughly documented (Conboy, 2009), (Qumer & Henderson-Sellers, 2008)), specifically, a value stream: to design, build, and implement a new minimum viable Agile BDAS methodology.
- DO.2 The designed artifact is useful, easy to use, and valuable (Galvan et al., 2021) for small businesses, software developers, and IT professionals.
- DO.3 The designed artifact is thoroughly documented, including the role set component, the phase-activity set component, and the template-artifact set component.

4.2.2 DESIGN RESTRICTIONS

For design constraints (DR), we must consider parameters such as time, budget, theoretical sources, and available software. The agreed-upon DRs are:

- DR.1 The designed artifact must be composed of basic design elements sourced from relevant theoretical design sources (DTS).
- DR.2 The designed artifact must be developed within a short-term period (maximum 6 months) and under the assigned research budget.
- DR.3 The designed artifact should be documented in an Electronic Process Guide.

4.2.3 DESING THEORETICAL SOURCES

The theoretical design sources (DTS) represent the main sources of design components (DC) that will be selected to create the designed artifact. These DTS are suggested and agreed upon by the research team based on their collective knowledge and a selective review involving an analysis of over 2000 articles focused on those with an impact factor greater than 1.0 in the most prominent journals. These articles were specifically selected from leading journals in fields such as Big Data, Analytics, Data Sciences, and Data Mining, as well as top software engineering journals. All of this was evaluated using resources available for this research, including access to free literature (Google Scholar) and supplementary journals accessible through this platform.

Table 4.1 Design components.

Design component number	SDLC	References
DTS.1	**CRISP-DM:** Cross-Industry Standard Process for Dara Mining	(Chapman et al., 2000).
DTS.2	**Scrum-XP**	(Schwaber & Sutherland, 2020) (Dudziak, 1999)
DTS.3	**TDSP:** Team Data Science Process	(Microsoft, 2107).
DTS.4	**DDS:** Data Driven Scrum	(Saltz, 2022).

Each element, such as roles, activities, and artifacts for the DTS, will be considered and discussed with the team to obtain the design components.

4.2.4 DESIGN COMPONENTS FOR THE EXPECTED ARTIFACT

After thoroughly evaluating the DTS, we have selected potential design components (DCs) that will be used in designing the artifact. It's possible that some components may not be used in the final design.

Tables 4.2, 4.3, 4.4, and 4.5 contain all selected design components from the four DTS by the research team based on their expertise and knowledge. An iterative process will be conducted to obtain the most important components for designing the artifact.

Table 4.2 DTS.1 CRISP-DM (Chapman et al., 2000).

Design Compoent	Design theoretical source (DTS)	Specific elements of the design component (DC) potentially to be used in the designed artifact
DC.1 CRISP-DM Phases	DTS.1 CRISP-DM (Chapman et al., 2000).	{Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment}
DC.2 CRISP-DM Activities	DTS.1 CRISP-DM (Chapman et al., 2000).	{Determine the objectives of data mining, Create a plan for your data mining project, Collect initial data, Describe the data, Explore the data, Check the quality of the data, Select data, Data cleansing, Data construction, Integrate the data, Format the data, Select modeling technique, Build the model, Assess model}
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	{Data mining goals, Data Mining Success Criterial, Initial Data Collection Report, Data Description Report, Data Exploration Report, Data Quality Report, Data Cleaning Report, Merged Data, Reformatted Data, Dataset, Dataset Description, Modeling Technique, Models, Model Assessment, Assessment of Data Mining Results}

Table 4.3 DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).

Design Compoent	Design theoretical source (DTS)	Specific elements of the design component (DC) potentially to be used in the designed artifact
DC.4 Scrum-XP Roles	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).	{Customer-Product Owner; Coach-Master; Development Team}
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).	{Exploration, Product Planning, Iteration-Sprint Planning, Iteration-Sprint, Product Release}
DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).	{Product vision definition, Product backlog definition, Product backlog prioritization, Spike testing, Product backlog effort estimation, Product backlog negotiation, Style codifying standard definition, Iteration-sprint user story selection, Iteration sprint user story task planning, Iteration-sprint user story plan negotiation, Stand-up meeting, customer functional tests elaboration, Simple design, Codification and unit testing, Increment Integration and customer functional testing, Iteration-sprint review and retrospective, Product releasing}
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).	{Product vision, Product backlog, Product backlog plan, Iteration-sprint plan, Kanban board, Burndown chart, Customer functional tests, Simple architecture design, Unit tests, Unit codes, Build increment, Iteration-sprint agreements, Product done}

Table 4.4DTS.3 TDSP (Microsoft, 2017).

Design Compoent	Design theoretical source (DTS)	Specific elements of the design component (DC) potentially to be used in the designed artifact
DC.8 TDSP Roles	DTS.3 TDSP (Microsoft, 2107).	{Group manager, Team lead, Project lead, Project individual contributors}
DC.9 TDSP Phases	DTS.3 TDSP (Microsoft, 2107).	{Business Understanding, Data Acquisition and Understanding, Modeling, Deployment, Customer Acceptance}
DC.10 TDSP Activities	DTS.3 TDSP (Microsoft, 2107).	{Define Objective, Identify Data Source, Ingest Data, Explore the Data, Set up a Data Pipeline, Feature Engineering, Model Training, Model Evaluation, Operationalize a Model, System Validation, Project hand-off}
DC.11 TDSP Artifacts	DTS.3 TDSP (Microsoft, 2107).	{Charter Document, Data Source, Data Dictionaries, Data Quality Report, Solution, Architecture, Checkpoint Decision, A status Dashboard, A final modeling report, A final solution architecture document, Exit report}

Table 4. 5 DTS.4 DDS (Saltz, 2022).

Design Compoent	Design theoretical source (DTS)	Specific elements of the design component (DC) potentially to be used in the designed artifact
DC.12 DDS Roles	DTS.4 DDS (Saltz, 2022).	{Product Owner, Process Expert, DDS Team Members}
DC.13 DDS Phases	DTS.4 DDS (Saltz, 2022).	{Brainstorm, Prioritize, Create / Refine Observe & analyze}
DC.14 DDS Activities	DTS.4 DDS (Saltz, 2022).	{Backlog Refinement, Prioritization of the Backlog, Iterations, Iteration Duration, Product Increments, Backlog Item Selection, Daily Meeting, Iteration Review, Retrospective}
DC.15 DDS Artifacts	DTS.4 DDS (Saltz, 2022).	{Item, Backlog, Item Breakdown Board, Task Board}

4.3 DESIGN AND DEVELOPMENT OF THE ARTIFACT

To design the BDAS methodology, the research team applied the means-ends analysis heuristic (Newell & Simon, 1972; Greeno et al., 1987) in four steps:

- Step1. To represent the design problem, an initial state Si is defined, a desired final state Sf, a set of heuristic operators {HOx(Sy, Sz), ...} that can transform state Sy to state Sz, a set of design objectives {DOj, ...}, and design constraints {DRk, ...} expected to be satisfied by the final state Sf. Additionally, two qualitative functions, EvalDOs(DO's) and EvalDRs(DR's), are used to evaluate the logical satisfaction of the DO's and DR's.

- Step 2. Set the initial state Si and the desired final state Sf, and determine the initial qualitative evaluations EvalDOs(DO's) and EvalDRs(DR's) for the initial state Si and the desired final state Sf.
- Step 3. Applying a sequence of heuristic operators {HO? (Si, S2); HO?(S2, S3); ...; HO?(Sn, Sf)} based on a logical analysis of the operators that can transform the initial state Si into the desired final state Sf.
- Step 4. Evaluate the degree of compliance of the desired final state Sf concerning the design objectives {DOj, ...} and the design constraints {DRk, ...}.

The process for creating our SDLC is divided into three stages, also known as iterations, to refine the DC of our SDLC in each iteration.

In the first iteration, all DCs that the working team establishes and considers necessary for implementation in our SDLC are selected. Once we have selected the DCs that the design team considers necessary, the working team discusses heuristically and based on each team member's experience which design components may be essential for our SDLC, excluding those that are unnecessary. This results in a second batch of DCs more aligned with our desired SDLC. Finally, for the third iteration, the working team discusses the final DCs needed for implementing our SDLC based on the Scrum-XP methodology (Schwaber & Sutherland, 2020), (Dudziak, 1999).

Appendix 10.2 contains all the information about this process, with the first and second iterations of the selected Design Components. Tables 4.6, 4.7, and 4.8 show the final selected DCs for roles, phases/activities, and artifacts. Figure 4.1 depicts the final BDAS methodology with all selected Design Components.

Table 4.6 Final Design Components for roles.

Roles						
Design Component	**Source**	**Why this could be helpful**	**SDLC that is also using it**			
			DTS.1	**DTS.2**	**DTS.3**	**DTS.4**
DC.4 **Scrum-XP Roles**	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**R.1 Customer-Product Owner**: The closest role to the stakeholders, this is the person who knows how to provide value to the project.		X		X
		R.2 Coach-Master: The person who is in charged to remove all the obstacles, coaching the team, ensuring the transparency, and promoting the self-organization.		X		X
DC.12 DDS Roles	DTS.4 DDS (Saltz, 2022)	**R.3 Team Members:** The team is made up of a cross-functional collection of team members, which can generate increment in each sprint.		X		X

Table 4.7 Final Design Components for Phases and Activities.

Phases and Activities						
Design Component	**Source**	**Why this could be helpful**	**SDLC that is also using it**			
			DTS.1	**DTS.2**	**DTS.3**	**DTS.4**
DC.5 **Scrum-XP Phases**	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 1 – Exploration**: The goal of the phase is to identify the needs of the project and select the highest priority items to work on, including the BDAS requirements.	X	X	X	
DC.6 **Scrum-XP** **Activities**	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.1.1 Product vision definition** Identify the objectives of the project, to generate a clear vision of the product and what you want to develop.	X	X	X	
DC.2 CRISP-DM Activities	DTS.1 CRISP-DM (Chapman et al., 2000).	**Activity A.1.2 Identify Data Architecture:** The required data sets available are defined, in addition to establishing a component diagram of the data architecture.	X		X	
DC.6 **Scrum-XP** **Activities**	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.1.3 Product backlog:** Create the user stories or tasks that need to be developed.		X		X

DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.1.4 Product backlog prioritization:** User stories are prioritized based on those that provide the most value to the project.		X		X
DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.1.5 Product backlog effort estimation:** Estimate every single user stories by the developer, it is possible to use fixed time or user stories points.		X		X
DC.9 TDSP Phases	DTS.3 TDSP (Microsoft, 2107).	**Phase 2 - Data Acquisition and Understanding:** In this phase, a clean and high-quality dataset is generated.	X		X	
DC.10 TDSP Activities	DTS.3 TDSP (Microsoft, 2107).	**Activity A.2.1 Ingest Data:** Data is extracted from the source destination to the location where the data will be processed.	X		X	
DC.2 CRISP-DM Activities	DTS.1 CRISP-DM (Chapman et al., 2000).	**Activity A.2.2 Clean Data:** Data is explored and processed to remove noise, improve quality, discrepancies or missing data.	X		X	
DC.10 TDSP Activities	DTS.3 TDSP (Microsoft , 2107).	**Activity A.2.3 Set up Architecture:** The data ingestion architecture is specified based on business needs and constraints.	X		X	
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 3 - Iteration-Sprint**: Build the increment in a Iterative process.		X		X
DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.3.1 Sprint Planning Modeling**: Select the most valuable user stories that the Product Owner will develop during the model generation sprint. The development team chooses the task based on their skills.		X		X
DC.14 DDS Activities	DTS.4 DDS (Saltz, 2022)	**Activity A.3.2 Iteration Duration:** Each iteration is capability-based (not time-boxed calendar events). Furthermore, each iteration should aim to be a minimally viable set of work that can deliver value.				X
DC.14 DDS Activities	DTS.4 DDS (Saltz, 2022)	**Activity A.3.3 Daily Meeting:** It is a daily 15-minute meeting that occurs every workday, where the activities being carried out by the work team are inspected.		X		X
DC.14 DDS Activities	DTS.4 DDS (Saltz, 2022)	**Activity A.3.4 Product Increments Modeling:** Implement requirements to develop user stories where the model is generated.		X		X

DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.3.5 Review and retrospective**: Conduct a retrospective of the entire team to know what is working in the development of the product and how to improve for the next sprints.		X		X
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 4 - Product Release**: Release the increment with the most important features chosen by the Owner.	X	X	X	
DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Activity A.4.1 Product releasing:** Release the increment.	X	X	X	

Table 4. 8 Final Design Components for Phases and Artifacts.

Processes Artifacts						
Design Component	**Source**	**Why this could be helpful**	**SDLC that is also using it**			
			DTS.1	**DTS.2**	**DTS.3**	**DTS.4**
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 1 - Exploration**: The goal of the phase is to identify the needs of the project and select the highest priority items to work on.	X	X	X	
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Artifact AR.1.1 Product vision:** Describes the overarching long-term mission of your product.	X	X	X	
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	**Artifact AR.1.2 Data mining goals:** Describe the intended outputs of the project that enables the achievement of the business objectives.	X			
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Artifact AR.1.3 Product backlog:** A prioritized list of work for the development team that is derived from the product roadmap and its requirements.		X		X
DC.9 TDSP Phases	DTS.3 TDSP (Microsoft, 2107).	**Phase 2 - Data Acquisition and Understanding:** Identify the objectives of the project, to generate a clear vision of the product and what you want to develop.	X		X	
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	**Artifact AR.2.1 Data Description Report:** Describe the data which has been acquired, including: the format of the data, the quantity of data.	X		X	

DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	**Artifact AR.2.2 Data Quality Report:** List the results of data quality verification; Data Cleansing Report: Describe what decisions and actions were taken to address data quality issues.	X		X	
DC.11 TDSP Artifacts	DTS.3 TDSP (Microsoft, 2107).	**Artifact AR.2.3 Solution Architecture:** Such as a diagram or description of your data pipeline that your team uses to run predictions on new data.			X	
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 3 - Iteration-Sprint**: Build the increment in a Iterative process.		X		X
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Artifact AR.3.1 Iteration-sprint plan: I**nvolves a planning meeting at the beginning of each sprint where the team analyzes the backlog items and divides them into tasks and tests.		X		X
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	**Artifact AR.3.2 Modeling Technique:** Document the actual modeling technique that is to be used.	X		X	
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	**Artifact AR.3.3 Model Assessment:** Summary of results of the evaluation of the applied models.	X		X	
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Artifact AR.3.4 Build increment:** A product increment is whatever you previously built, plus anything new you just finished in the latest sprint, all integrated, tested, and ready to be delivered or deployed.		X		X
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Phase 4 - Product Release**: Release the increment with the most important features chosen by the Owner.	X	X	X	
DC.11 TDSP Artifacts	DTS.3 TDSP (Microsoft, 2107).	**Artifact AR.4.1 Exit report:** This technical report contains details about the project that the customer can use to learn how to operate the system.	X	X	X	
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	**Artifact AR.4.2 Product done:** The final release with the final increment.	X	X	X	

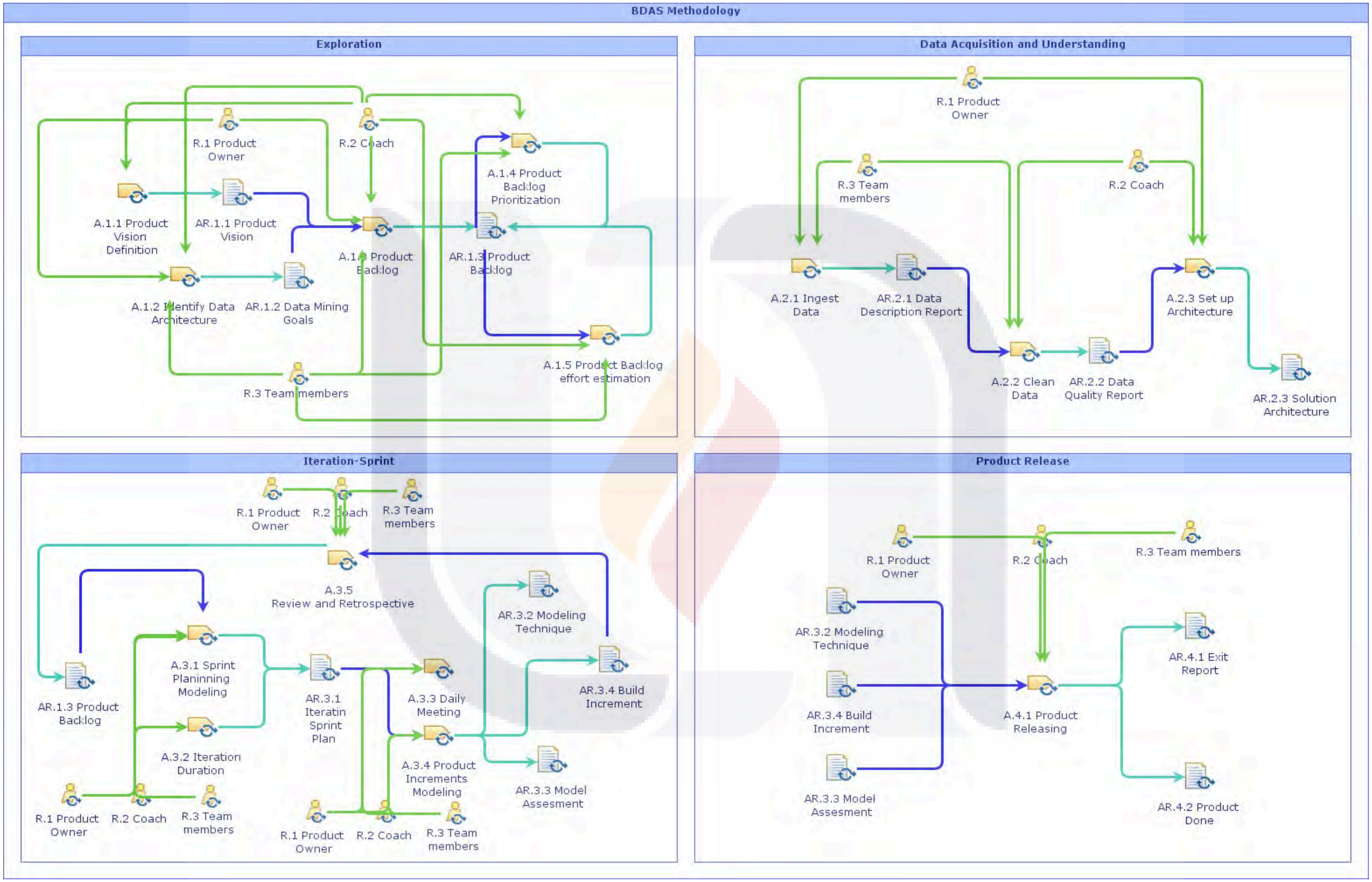


Figure 3.44 BDAS Methodology Conceptual Map.

4.4 EVALUATION OF THE ARTIFACT

It was agreed to design and develop the Process Management Document (EPG) for the newly proposed methodology, AgileDSA (Agile Data Science Analytics Methodology). However, to design and develop the EPG, it was necessary to have the source content of the EPG structure, which we referred to as AgileDSA (EPG).

Therefore, to design a methodology that software developers perceive as agile, user-friendly, useful, compatible, and valuable, while incorporating the key Big Data Analytics System (BDAS) features highlighted in other methodologies, four theoretical sources were identified. From these sources, the design components such as roles, phases, activities, and work products were derived.

This process was thoroughly carried out by the principal researcher and discussed with both the primary thesis advisor and the external advisor. Multiple iterations were required to refine the methodology at various general levels, and this iterative process is documented in Appendix 10.1. As a result, the AgileDSA Process Management Document (AgileDSA EPG) was developed, providing a detailed description of each component of the proposed methodology. Additionally, freely available templates are suggested to facilitate the use of the methodology by any individual or organization.

4.5 Design Electronic Process guide (EPG)

The Electronic Process Guide (EPG) of AgileDSA – Agile Data Science Analytics Methodology - was developed using Visual Studio Code with HTML, CSS, and JavaScript.

This final product, AgileDSA – Agile Data Science Analytics Methodology EPG, is freely available for consultation at the following web link (or may be requested via email at gerardo.salazar@edu.uaa.mx):

https://agile-data-science-analytics-development-methodology-gss.on.drv.tw/DCAT.RESEARCH.GSS/.

5. APLICATION AND EVALUATION OF RESULTS

5.1 CONCEPTUAL EVALUATION OF AGILEDSA (AGILE DATA SCIENCE ANALYTICS METHODOLOGY)

Before building the AgileDSA (EPG), it was required to establish an adequate theoretical validity level for the content of the AgileDSA (EPG) document. It was used the technique called **"Validation by Panel of Experts"** (Beecham et al., 2005) was used. This technique has been previously used in several important studies in the domain of Software Engineering (Dybå, 2000; Niazi et al., 2005; Beecham et al., 2005) This validation technique has been considered relevant and useful, and necessary to be applied to establish a validity of the content (also called "model validation" in the simulation domain (Sargent, 2000; 2013) on textual documents (sentences, paragraphs, or pages). We consider "validity of the content" as "the overall level of veracity and congruence with the overall purpose of the content" (Mora, 2009). This definition implies that "valid content" is expected to be finally used for the planned purpose and to be in an adequate range of overall veracity. It can be considered like the concept of a model, that no entity to be validated can have an overall 100%, because any model is only a partial representation of a real situation, and it is impossible to elaborate a model equal to this real situation.

Thus, in this section, it was applied a "validity of content" technique was applied with a Panel of Experts, based on similar techniques used in Simulation (Sargent, 2000; 2013). As Sargent (2013; p. 14) establishes: **"Conceptual model validation is defined as determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is 'reasonable' for the intended purpose of the model".**

The steps followed for this validation were the following:

1. **To have the textual document validated.** A user guide (EPG) for the proposed AgileDSA methodology (AgileDSA EPG) was developed for validation purposes. A comprehensive version of the EPG was prepared. The research team involved in this doctoral study conducted an internal review. After minor corrections, the AgileDSA EPG was deemed ready for evaluation. It was then published on a public

website: https://agile-data-science-analytics-development-methodology-gss.on.drv.tw/DCAT.RESEARCH.GSS/

2. **To define the criteria for expert inclusion.** The criteria were defined as follows: 2.1) holding at least a master's degree, either for academics or for professionals; 2.2) having relevant experience in BDAS projects, or relevant experience in projects involving the use of the SCRUM methodology or another agile methodology. For this phase, evaluations were collected from both researchers and academics, as well as industry professionals. The objective of AgileDSA (EPG) is to support both of these communities—academics and professionals—at all levels of expertise, from beginners to experts.

3. **To have ready a suitable questionnaire to be applied to the Panel of Experts.** This questionnaire was taken from Mora (2009). This questionnaire contains three constructs: C1 Demographic Data of the Panel of Experts, C2 Pilot Evaluation, and C3 Conceptual Evaluation by Panel of Experts. The C1 contains 8 items, the C2 contains 17 items, and the C3 contains 7 items. This questionnaire is relatively new, but it has been used in previous studies (Mora, 2009; Reyes-Delgado et al., 2016). This questionnaire is available through gerardo.salazar@edu.uaa.mx (author's email). This questionnaire also asked for demographic data (required to identify whether the 3 selection criteria were achieved by each evaluator). The constructs of interest to be evaluated for the sample of international academics and professionals are presented in Table 5.1. (The surveys themselves can be found in Appendices 10.2, 10.3, 10.4, and 10.5).

Table 5.1 Conceptual Metrics.

CONSTRUCT	SCALE
The conceptual product (_) is supported by robust theoretical knowledge (e.g. based on scientific literature).	5-points Likert (1: strongly disagree to 5: strongly agree)
The theoretical knowledge used for elaborating this conceptual product (_) is relevant for the addressed topic.	5-points Likert (1: strongly disagree to 5: strongly agree)
The scientific literature considered for elaborating this conceptual product (_) does not present important omissions for the topic.	5-points Likert (1: strongly disagree to 5: strongly agree)
The conceptual product (_) is logically coherent.	5-points Likert (1: strongly disagree to 5: strongly agree)
The conceptual product (_) is adequate for achieving the purpose of its utilization.	5-points Likert (1: strongly disagree to 5: strongly agree)
The conceptual product (_) provides new scientific-based knowledge (e.g. it is not a just a duplication of an existent conceptual product).	5-points Likert (1: strongly disagree to 5: strongly agree)
The presentation style of the conceptual product (_) is adequate for a scientific report.	5-points Likert (1: strongly disagree to 5: strongly agree)

4. **To define a list of potential experts to be contacted.** A set of international groups was defined for outreach. Specifically, a list of three international groups was established: 1) academic contacts provided by senior doctoral advisors; 2) professionals from international LinkedIn groups related to BDAS or SCRUM, XP; and 3) professional contacts of doctoral students and their advisors. The criterion used to distinguish expert profiles from basic ones was based on the number of years of experience in BDAS projects or the use of the SCRUM, XP methodologies.

   The survey was created and administered online using the Google Forms tool and was distributed to a sample of 20 individuals who agreed to participate. For the conceptual validation, a filtering process was applied to classify respondents according to their experience level, distinguishing between expert and basic profiles in BDAS projects or SCRUM, XP methodology. Table 5.2 presents the demographic data of the sample, consisting of the 8 evaluators who passed the screening process.

Table 5.2 Demographic Data of the Panel of Experts (Approved).

VARIABLE	FREQUENCY	PERCENTAGE
Academic background level:		
Master's degree or PhD	7	90.0
without master's degree or doctorate	1	10.0
Main work setting:		
Business enterprise	4	50.0
University/Research Unit	2	25.0
Government Unit	2	25.0
Scope of work setting:		
Regional	0	0.0
Nationwide	3	37.5
Worldwide	5	62.5
Region of working setting:		
Latin America	8	100
USA/CAN	0	0.0
Europe	0	0.0
Main Work Position:		
Academic/Researcher	2	25.0
IT Project Manager / IT Consultant	5	62.5
Business Manager / Business Consultant	0	0.0
IT Senior Developer	1	12.5
Self-evaluation on the expertise level AGILE PROCESS (Scrum, XP):		
very high level of expertise	2	25.0
high level of expertise	6	75.0
moderate level of expertise	0	0.0
low level of expertise	0	0.0
very low level of expertise	0	0.0
Self-evaluation on the expertise level on Data Science Analytics Systems:		
very high level of expertise	0	0.0
high level of expertise	3	37.5
moderate level of expertise	4	50.0
low level of expertise	1	12.5
very low level of expertise	0	0.0

5. **To define a list of potential experts to be contacted.** Debido al tamaño muestral de 8, se empleó la técnica estadística PLS (Chin, 2010). Esta técnica es una técnica estadística multivariante de segunda generación que se utiliza con muestras pequeñas. La

fiabilidad se calculó con el índice de fiabilidad compuesta, la validez convergente con las cargas factoriales y la validez discriminante con la AVE (varianza media extraída para cada constructo).

6. To calculate mean and standard deviation of each item in the questionnaire. The mean and standard deviation are reported in the Table 5.3 It was used a Likert scale from 1 (total disagreement with asked item) to 5 (total agreement with asked item).

Table 5.3 Mean and Standard Deviation of the Constructs/Items C1 and C2.

CONSTRUCT / ITEMS	MEAN	STD.DEV.
C1 THEORETICAL VALIDITY	**4.42**	**0.73**
ITEM#1. The conceptual product is supported by robust theoretical knowledge (e.g. based on scientific literature).	4.28	0.75
ITEM#2. The theoretical knowledge used for elaborating this conceptual product is relevant for the addressed topic.	4.57	0.78
C2 THEORETICAL CONSISTENCY	**4.40**	**0.56**
ITEM#3. The scientific literature considered for elaborating this conceptual product does not present important omissions for the topic.	4.14	0.89
ITEM#4. The conceptual product is logically coherent.	4.42	0.53
ITEM#5. The conceptual product is adequate for achieving the purpose of its utilization.	4.57	0.53
ITEM#6. The conceptual product provides new scientific-based knowledge (e.g. it is not a just a duplication of an existent conceptual product).	4.14	1.00
ITEM#7. The conceptual product is supported by robust theoretical knowledge (e.g. based on scientific literature).	4.71	0.48

In addition, a one-sample, one-tailed t-test of means was performed with the null hypotheses H0.1 “The mean of construct C1 is less than or equal to 3.0” and H0.2 “The mean of construct C2 is less than or equal to 3.0”. Both null hypotheses were rejected, so the means achieved by constructs C1 and C2 are considered satisfactory. Table 5.4 shows these results.

7. To assess the level of validity achieved by the document. Based on the reliability and validity results (convergent and discriminant) of the instrument used to measure the theoretical validity perceived by a panel of experts, and on the results obtained on the means of constructs C1 and C2, it can be assessed that the document is considered theoretically valid and, therefore, conceptually the EPG of AgileDSA (Agile Data Science Analytics Methodology) can be used.

Table 5.4 Null Hypotheses Tests on Means of Constructs C1 and C2.

NULL HYPOTHESIS	MEAN OF CONSTRUCT	STD.DEV OF CONSTRUCT	T-VALUE	P-VALUE	REJECT HO?
H0.1 "The mean of the construct C1 is less or equal to 3.00"	**4.42**	**0.731**	**5.16**	**< 0.0020**	**YES**
H0.1 "The mean of the construct C2 is less or equal to 3.00"	**4.4**	**0.565**	**6.54**	**< 0.0006**	**YES**

5.2 EVALUATION OF AGILEDSA (AGILE DATA SCIENCE ANALYTICS METHODOLOGY)

The AgileDSA SDLC was shared with DSA academics and professionals through the web-based Application Programming Guide (EPG), and they were asked to evaluate its usability metrics via a questionnaire based on widely cited studies (Moore & Benbasat, 1991; Karahanna et al., 1999; Lee et al., 2001). The constructs of interest used to assess the usability of the AgileDSA methodology by the panel of BDAS academics and professionals are presented in Table 5.2.

Table 5.5 Constructs to be Evaluated for the Panel DSA Academics and Practitioners on the AgileDSA SDLC.

CONSTRUCT	ITEMS	SCALE	SOURCE
USEFULNESS – *is the degree to which using the new TOOL is perceived as being better than using the current used TOOL.*	4	5-points Likert (1: strongly disagree to 5: strongly agree)	Moore & Benbasat (1991); Karahanna *et al.* (1999)
EASE OF USE - *is the degree to which using the new TOOL is perceived as being free of effort.*	3	5-points Likert (1: strongly disagree to 5: strongly agree)	Moore & Benbasat (1991); Karahanna *et al.* (1999)
COMPATIBILITY - *is the degree to which using new the TOOL is perceived as compatible with what people do.*	3	5-points Likert (1: strongly disagree to 5: strongly agree)	Moore & Benbasat (1991); Karahanna *et al.* (1999)
VALUE - *the degree to which using the new TOOL is perceived as a value delivery entity for users by savings on money, time, and the provision of a variety of valuable resources, and by an overall value.*	4	5-points Likert (1: very low to 5: very high)	Lee *et al.* (2001)
ATTITUDE - *it reflects the individual's positive and negative evaluations of performing the behavior (of adopting the evaluated artifact).*	3	7-point Semantic differential scale (-3 to +3)	Karahanna *et al.* (1999)

A total of 20 academics and professionals from Latin America participated in the study to provide demographic data, which was analyzed in full. The data can be found in Table 5.3.

Participants were given sufficient time to review the AgileDSA Usage Guide (EPG) and its associated templates. Subsequently, demographic data and usability questionnaires were administered. In the usability questionnaire, participants were asked to evaluate five usability metrics—usefulness, ease of use, compatibility, value, and attitude toward potential use—for both the AgileDSA SDLC and any alternative BDAS SDLCs currently or previously used by the evaluators.

Statistical analysis was conducted using the Partial Least Squares (PLS) method (Barclay et al., 1995; Chin, 1998; Russo & Stol, 2021). PLS is a second-generation multivariate analysis technique that is particularly useful for: 1) simultaneously assessing reliability, discriminant and convergent validity of constructs, regression coefficients between hypothetical construct associations (known as path analysis), and the explained variance (R2) of dependent constructs; 2) small sample sizes; and 3) datasets that do not conform to normal distribution assumptions for each construct indicator.

Table 5.6 Demografic Data of the Panel of Expert.

VARIABLE	FREQUENCY	PERCENTAGE
Academic background level:		
• Bachelor	3	15.0
• Master level	14	70.0
• Doctorate	3	15.0
Main work setting:		
• Government Unit	6	30.0
• University/Research Unit	6	30.0
• Business enterprise	8	40.0
Years in work settings:		
• 1-5 years	3	15.0
• 6-10 years	8	40.0
• 11-15 years	2	10.0
• 16-20 years	3	15.0
• 20 or more years	4	20.0

Main Work Position:		
- IT Project Manager / IT Consultant	5	25.0
- Academic/Researcher	6	30.0
- IT Senior Developer	9	45.0
Working Region:		
- Latin America	20	100
Scope of work setting:		
- Nationwide	7	35.0
- Worldwide	8	40.0
- Regional	5	25.0

Tables 5.4 and 5.5 present, respectively, the evaluation of the AgileDSA methodology and the alternative SDLC for BDAS projects, including descriptive statistics, reliability measures, and discriminant validity of the evaluation dataset. Descriptive statistics (median, mean, and standard deviation) were calculated using the free software JASP (JASP, 2025), while reliability (Cronbach's alpha and composite reliability index) and discriminant validity statistics (Average Variance Extracted [AVE]) were computed using the free academic version of SmartPLS v4 (SmartPLS, 2025).

The results in Tables 5.4 and 5.5 support the evidence for retaining four final constructs—usefulness, ease of use, value, and attitude toward potential use—each measured with satisfactory levels of reliability and discriminant validity (Barclay et al., 1995; Chin, 1998; Russo & Stol, 2021). The construct compatibility was excluded from the final analysis in both tables due to unsatisfactory reliability and validity metrics. The PLS models generated using SmartPLS v4 are shown in Figure 5.1 for the AgileDSA methodology and Figure 5.2 for the alternative BDAS SDLC.

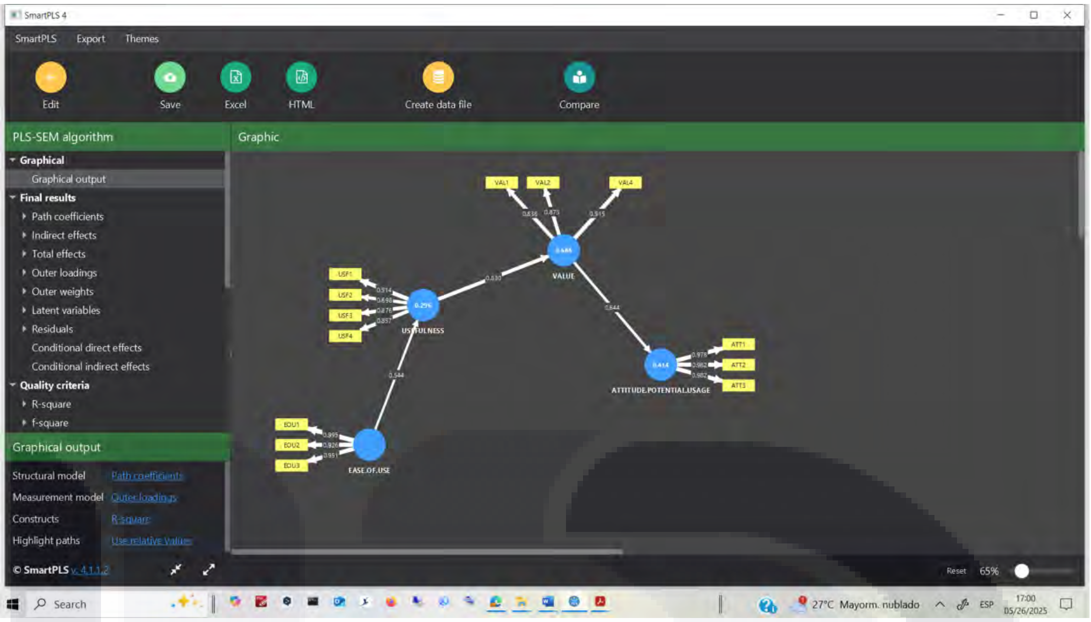


Figure 5.1 PLS model AgileDSA SDLC.

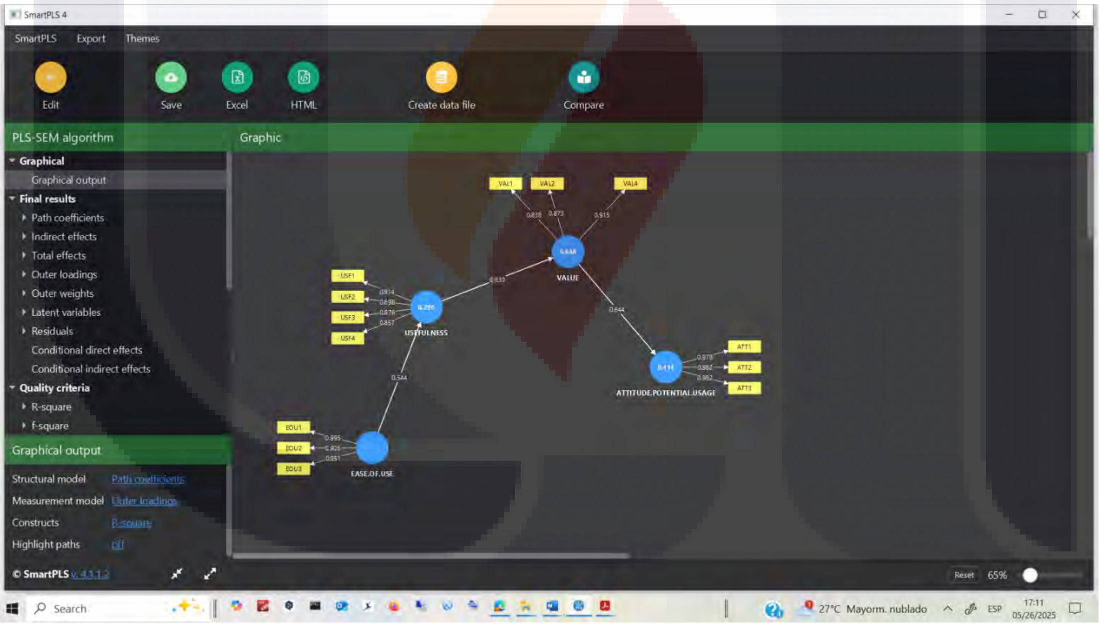


Figure 5.2 PLS model alternative SDLC.

Table 5.7 Descriptive, Reliability and Discriminant Validity of the Usability Constructs for AgileDSA SDLC.

Construct	Median	Mean	Standard Dev.	Cronbach´s Alpha >= 0.70	Composite Reliability Index >= 0.70	Average Variance Extracted (AVE) >= 0.500
USEFULNESS	4.125	4.100	0.656	0.864	0.918	0.707
EASE OF USE	4.665	4.417	0.674	0.954	0.964	0.917
VALUE	4.165	4.200	0.565	0.848	0.847	0.767
ATTITUDE OF POTENTIAL USAGE	2.000	1.466	1.040	0.980	0.984	0.962

Table 5.8 Descriptive, Reliability and Discriminant Validity of the Usability Constructs for the alternative BDAS SDLC.

Construct	Median	Mean	Standard Dev.	Cronbach´s Alpha >= 0.70	Composite Reliability Index >= 0.70	Average Variance Extracted (AVE) >= 0.500
USEFULNESS	3.000	3.413	0.832	0.904	0.919	0.778
EASE OF USE	3.165	3.533	0.964	0.947	0.992	0.903
VALUE	3.165	3.417	0.815	0.946	0.952	0.903
ATTITUDE OF POTENTIAL USAGE	0.000	0.433	1.382	0.991	0.992	0.983

Tables 5.6 and 5.7 present, respectively, the complementary discriminant validity statistics for the AgileDSA methodology and the alternative BDAS SDLC, based on the evaluation dataset. These statistics were calculated using the free SmartPLS v4 software (SmartPLS, 2025). The results from both tables provide supporting evidence for the assessment of the four final constructs—usefulness, ease of use, value, and attitude toward potential use—with satisfactory discriminant validity (Barclay et al., 1995; Chin, 1998; Russo & Stol, 2021). These tables show that the diagonal values (the square root of the AVE for each construct) are greater than the off-diagonal values, indicating that each construct shares more variance with its indicators than with those of other constructs (Barclay et al., 1995).

Table 5.9 Discriminant Validity of the Usability Constructs for the AgileDSA SDLC.

	ATTITUDE OF POTENTIAL USAGE	EASE OF USE	USEFULNESS	VALUE
ATTITUDE OF POTENTIAL USAGE	**0.981**	0.261	0.724	0.644
EASE OF USE	0.261	**0.958**	0.544	0.507
USEFULNESS	0.724	0.544	**0.841**	0.830
VALUE	0.644	0.507	0.830	**0.876**

Table 5.10 Discriminant Validity of the Usability Constructs for the alternative BDAS SDLC.

	ATTITUDE OF POTENTIAL USAGE	EASE OF USE	USEFULNESS	VALUE
ATTITUDE OF POTENTIAL USAGE	**0.991**	0.337	0.704	0.750
EASE OF USE	0.377	**0.950**	0.620	0.683
USEFULNESS	0.704	0.620	**0.882**	0.777
VALUE	0.750	0.683	0.777	**0.950**

Tables 5.8 and 5.9 present, respectively, the convergent validity statistics for the evaluation dataset corresponding to the AgileDSA methodology and the alternative BDAS SDLC. These statistics were also calculated using the free SmartPLS v4 software (SmartPLS, 2025). The results from both tables provide sufficient evidence to confirm adequate convergent validity for the four final constructs: usefulness, ease of use, value, and attitude toward potential use (Barclay et al., 1995; Chin, 1998; Russo & Stol, 2021). These tables show that the loadings (i.e., correlations) of each construct's items are above 0.700 and higher than the cross-loadings (i.e., correlations with items of other constructs), which supports the presence of strong convergent validity (Barclay et al., 1995).

Table 5.11 Convergent Validity of the Usability Constructs for the AgileDSA SDLC.

Discriminant validity – Cross loadings				
	ATTITUDE.POTENTIAL.USAGE	EASE.OF.USE	USEFULNESS	VALUE
ATT1	0.978	0.234	0.677	0.597
ATT2	0.982	0.273	0.750	0.670
ATT3	0.982	0.259	0.701	0.622
EOU1	0.269	0.995	0.569	0.512
EOU2	0.294	0.926	0.480	0.493
EOU3	0.189	0.951	0.510	0.453
USF1	0.533	0.699	0.914	0.791
USF2	0.632	0.023	0.698	0.493
USF3	0.723	0.460	0.876	0.747
USF4	0.627	0.413	0.857	0.693
VAL1	0.605	0.323	0.698	0.838
VAL2	0.558	0.593	0.746	0.873
VAL4	0.526	0.414	0.734	0.915

Table 5.12 Convergent Validity of the Usability Constructs for the alternative BDAS SDLC.

Discriminant validity – Cross loadings				
	ATTITUDE.POTENTIAL.USAGE	EASE.OF.USE	USEFULNESS	VALUE
ATT1	0.986	0.341	0.682	0.727
ATT2	0.991	0.402	0.691	0.740
ATT3	0.996	0.377	0.718	0.763
EOU1	0.359	0.977	0.595	0.576
EOU2	0.406	0.963	0.688	0.744
EOU3	0.285	0.909	0.433	0.608
USF1	0.623	0.570	0.856	0.700
USF2	0.559	0.391	0.794	0.557
USF3	0.646	0.584	0.931	0.745
USF4	0.649	.0609	0.938	0.719
VAL1	0.768	0.672	0.794	0.973
VAL2	0.710	0.645	0.640	0.938
VAL4	0.658	0.628	0.773	0.938

Finally, we conducted four hypothesis tests to gather evidence supporting a more favorable perception of the four usability constructs for the AgileDSA methodology compared to the alternative BDAS SDLC. Due to the lack of satisfactory normality test results, the non-parametric Wilcoxon Matched-Pairs Signed-Rank Test was used (Sheskin, 2000). Table 5.10 presents the results obtained. These four tests were calculated using the free JASP software (JASP, 2025). The results indicate that the evaluators perceived the new AgileDSA methodology as having better usability metrics than the alternative BDAS SDLC.

Table 5.13 Wilcoxon Signed-Rank Tests for the Usability Constructs in AgileDSA SDLC vs alternative BDAS SDLC.

Null Hypothesis	AgileDSA SDLC Median (med.1)	Alternative BDSA SDLC Median (med.2)	P-value	Implication
H0.1 For USEFULNESS construct (med.1<= med.2)	4.125	3.000	0.002	H0.1 is rejected, and thus the USEFULNESS of AgileDSA SDLC is better.
H0.2 For EASE OF USE construct (med.1<= med.2)	4.665	3.165	< 0.001	H0.2 is rejected, and thus the EASE OF USE of AgileDSA SDLC is better.
H0.3 For VALUE construct (med.1<= med.2)	4.165	3.165	< 0.001	H0.3 is rejected, and thus the VALUE of AgileDSA SDLC is better.
H0.4 For ATTITUDE OF POTENTIAL USAGE construct (med.1<= med.2)	2.000	0.000	0.001	H0.4 is rejected, and thus the ATTITUTE OF POTENTIAL USAGE of AgileDSA SDLC is better.

6. CONCLUSIONS
6.1 SUMMARY OF RESULTS

Section 1.3 of this document defined the research questions (RQ) and the null hypotheses (H0). The tables below present the results obtained for each research question and its associated hypothesis. It is important to note that journal and conference articles related to the topic were analyzed up to December 2023. These references were used to provide theoretical grounding and to reinforce the scientific methodological validity of this research.

Table 6.1 Summary of Results of this Ph.D. research for Research Question RQ.1

Research Question	Hypotheses	Results
RQ.1 What is the state of the art – contributions and limitations- on agile and non-agile development methodologies for Big Data-Data Science-Analytics Software Systems?	**H0.1** There is no need for an agile development methodology for Big Data-Data Science-Analytics Software Systems.	**The null hypothesis H0.1 is REJECTED.** The rejection of hypothesis H0.1 is based on the results of a specific literature review on agile development methodologies for BDAS (Big Data Analytics Systems) projects. The review involved a targeted search across 27 leading journals in Big Data Analytics Systems and 19 prominent journals in Software Engineering. Over 2,000 articles were analyzed to identify existing agile methodologies adapted to BDAS projects. From this review, only one relevant study was identified: “The Design of a Software Engineering Lifecycle Process for Big Data Projects” (Lin & Huang, 2018). Due to the lack of reported methodologies in the academic literature, six additional proprietary methodologies were included—identified through gray literature sources—to enrich the analysis.

Table 6.2 Summary of Results of this Ph.D. research for Research Question RQ.2.

Research Question	Hypotheses	Results
RQ.2 What is the state of the art – capabilities and limitations – of open-source development platforms for Big Data-Data Science-Analytics Software Systems?	**H0.2** There are no available open-source development platforms for Big Data-Data Science-Analytics Software Systems that can be satisfactorily evaluated in the technical, end-user, and organizational dimensions.	**The null hypothesis H0.2 is REJECTED.** The analysis of the methodologies mentioned in hypothesis H0.1 demonstrates that there are currently various open-source software alternatives capable of successfully supporting BDAS (Big Data Analytics Systems) projects. It was identified that it is possible to generate value within organizations without necessarily applying the "V" criteria typically required for a project to be considered Big Data. This finding broadens the scope of adoption, allowing smaller organizations, research groups, and startups to access the benefits of Big Data technologies. The literature review confirmed that BDAS projects exhibit specific characteristics described in this study, thereby supporting the need for a dedicated methodology. Although several methodologies have been proposed, most are reported as incomplete. One of the most significant findings regarding BDAS projects is the widespread use of Python and R programming languages. These open-source languages are extensively used in the development of BDAS projects and are among the most well-supported and well-documented languages in the community. Both have essential plugins and libraries critical for the development of BDAS solutions.

Table 6.3 Summary of Results of this Ph.D. research for Research Question RQ.3.

Research Question	Hypotheses	Results
RQ.3 What elements of Agile Development and Big Data-Data Science-Analytics Development Methodologies can be used to elaborate an Agile Development Methodology for Big Data-Data Science-Analytics Software Systems that can be evaluated theoretically valid from a Panel of Experts?	**H0.3** There are no elements of Agile Development and Big Data-Data Science-Analytics Development Methodologies that can be used to elaborate an Agile Development Methodology for Big Data-Data Science-Analytics Software Systems that can be evaluated theoretically valid from a Panel of Experts.	**The null hypothesis H0.3 is REJECTED.** In the search for elements to develop a new agile methodology that is easy to use, useful, compatible, and valuable for BDAS (Big Data Analytics Systems) projects, several existing methodologies were identified that include key elements such as roles, phases, activities, and artifacts. Initially, seven methodologies were identified; however, after a detailed analysis and comparison with the SCRUM-XP methodology, three were selected and approved: • **CRISP-DM**, recognized as the most widely used methodology. • **TDSP**, due to its agile nature. • **DDS**, selected for its close relationship with SCRUM. The decision to select these three methodologies was made following a thorough analysis of each, focusing on their roles, phases, activities, and artifacts. After evaluating these methodologies, the research team heuristically selected the design components to generate a new methodology: AgileDSA. The iterative process carried out to develop the new methodology is documented in Appendix 10.1. The methodology was evaluated by a panel of experts composed of 8 academics, researchers, and professionals who have worked with agile methodologies or with methodologies designed for BDAS projects. The evaluation of the AgileDSA methodology by this panel was satisfactory.

Table 6.4 Summary of Results of this Ph.D. research for Research Question RQ.4.

Research Question	Hypotheses	Results
RQ.4 Can the new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems be documented in an Electronic Process Guide (EPG), and be evaluated as agile, useful, ease of use, compatible and valuable from a pilot group of Big Data-Data Sciences-Analytics academics and practitioners?	**H0.4.1** The new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems cannot be documented in an Electronic Process Guide (EPG).	**The null hypothesis H0.4.1 is REJECTED.** The methodology was successfully documented, culminating in the development of a formalized Electronic Process Guide (EPG). The final artifact, titled AgileDSA (Agile Data Science Analytics Methodology), was implemented using web development technologies (HTML, CSS, and JavaScript) within the Visual Studio Code development environment. This EPG provides a clear, navigable, and accessible structure, offering users a practical guide for implementing the methodology. Furthermore, the developed EPG is publicly available for consultation at the following link: https://agile-data-science-analytics-development-methodology-gss.on.drv.tw/DCAT.RESEARCH.GSS/ Therefore, it is demonstrated that the documentation and structuring of the proposed methodology within an EPG is entirely feasible, robust, and functional.
	H0.4.2 The new elaborated Agile Development Methodology for Big Data-Data Science-Analytics Software Systems is not considered agile, useful, ease of use, compatible and valuable from a pilot group of Big Data-Data Sciences-Analytics academics and practitioners.	**The null hypothesis H0.4.2 is REJECTED.** The collected data reveal a positive perception of the proposed methodology across all evaluated dimensions. Notably, the new methodology received favorable ratings in terms of agility, usefulness, ease of use, compatibility, and value, and it also outperformed the methodologies traditionally used by the respondents. These results validate the favorable reception of the developed methodology by potential users and confirm its relevance, applicability, and comparative advantage for Big Data Science projects.

6.2 GENERAL AND SPECIFIC RESEARCH OBJECTIVES

Based on the research context presented, the main problem identified was the lack of specialized development methodologies for Big Data Science Analytics (BDAS) projects that are perceived by software developers as agile. That is, methodologies that are not overly rigid but are also easy to use, useful, compatible, and valuable in practical application.

In response to this problem, the present research focused on confirming this gap and proposing an appropriate solution from the perspective of software engineering. The results obtained confirm the need for well-documented, comprehensive, and agile BDAS methodologies. As demonstrated in this study, agility is compatible with BDAS projects. Most existing options are either proprietary or highly rigorous methodologies that are overly burdensome and have been in the market for a long time, which limits their adoption and adaptation across different contexts. Another important point to highlight is that most of these methodologies are designed for large-scale projects or organizations that require a specific architecture for project development.

To address this issue, a new methodology was designed and developed based on one of the most widely used agile methodologies—SCRUM-XP—while also incorporating design components from three of the most prominent methodologies for BDAS project development: CRISP-DM, TDSP, and DDS. This proposal includes the definition of specific roles, phases, activities, and artifacts tailored for Big Data Science Analytics projects, and is complemented by the development of an Electronic Process Guide (EPG) that systematizes its application.

Subsequently, the methodology was published and evaluated through surveys conducted with professionals and academics in the field of BDAS. They positively assessed its usefulness, ease of use, compatibility, and value compared to existing BDAS methodologies. The results not only validate the relevance of the proposed methodology but also demonstrate a higher level of acceptance compared to other methodologies previously used by the respondents, thereby supporting the significance and contribution of this research.

In conclusion, this thesis not only confirms the initially identified need but also provides a concrete contribution to the field of software engineering applied to Big Data Science Analytics projects, by offering an open, specialized methodology that has been empirically validated for its quality and practical usefulness.

6.3 CONTRIBUTIONS AND DELIVERABLES

The following outcomes were obtained from this research:

1. For the Theory of Software Engineering

   - A chapter published in a Springer International Publishing journal under the title "**A Selective Comparative Review of CRISP-DM and TDSP Development Methodologies for Big Data Analytics Systems**".
   - A research article for an IAJIT-indexed journal on theoretical analysis, entitled "**REVIEW OF AGILE SDLC FOR BIG DATA ANALYTICS SYSTEMS IN THE CONTEXT OF SMALL ORGANIZATIONS USING SCRUM-XP**".
   - A research article submitted to an indexed journal, presenting a theoretical analysis, entitled "**A COMPARATIVE REVIEW OF THE MAIN HEAVYWEIGHT AND AGILE SDLC DEVELOPMENT LIFE CYCLES FOR BI DATA ANALYTICS SYSTEMS (BDAS): 2000-2023 PERIOD**" (submitted).
   - A research article submitted to an indexed journal, presenting the AgileDSA methodology proposal and its empirical evaluation, entitled "**DESIGN AND USABILITY EVALUATION OF AGILEDSA: A SCRUM-XP ALIGNED SDLC FOR BIG DATA ANALYTICS SYSTEMS IN SMALL BUSINESS**" (submitted).

2. For the Software Engineering Practice

   - A new lightweight DS methodology: An agile methodology for Big Data Science Analytics (BDAS) projects, made available through a free online Electronic Process Guide (EPG): https://agile-data-science-analytics-development-methodology-gss.on.drv.tw/DCAT.RESEARCH.GSS/
   - A new Ph.D. graduate in Software Engineering.

6.4 CONCLUSIONS

Following the design, development, and empirical validation of the new Agile Development Methodology for BDAS projects—AgileDSA—it can be concluded, based on the results detailed in this research, that the design, construction, and evaluation of this methodology were both justified and significant. The methodology was successfully evaluated by 20 international reviewers, including both academics and professionals. This methodological proposal, based on one of the most widely used agile methodologies, SCRUM-XP, and enriched with elements from CRISP-DM, TDSP, and DDS—three of the most important methodologies for BDAS project development—demonstrates that it is indeed possible to systematize and adapt software engineering practices to meet the specific needs of Big Data projects, particularly those developed within small enterprises.

This doctoral research aimed to design a theoretically grounded and practically viable methodology with the following characteristics:

- An agile methodology that avoids the excessive documentation and rigor currently present in BDAS projects.
- An open-access methodology that is adaptable to different contexts.
- A hybrid framework that combines the most effective elements of recognized methodologies for data science project development.
- A formalized Electronic Process Guide (EPG) is designed to promote understanding, accessibility, and practical applicability for both academics and professionals.

The resulting product —an AgileDSA Electronic Process Guide (EPG)— is openly available and has been positively evaluated by a pilot group of professionals and researchers in terms of agility, usefulness, ease of use, compatibility, and overall value. Therefore, this research recommends its practical application in professional environments and its academic adoption for teaching development methodologies in BDAS projects.

The theoretical robustness and empirical validation of the methodology position it as a significant contribution to the field of software engineering. It addresses a previously unmet gap and provides a valuable tool to improve the quality and structure of Big Data project execution across various organizational contexts, particularly in small enterprises.

7. DISCUSSION OF RESULTS

7.1 DISCUSSION ON THEORETICAL FRAMEWORK

To develop the theoretical framework, a literature review was conducted on three main topics. The study focused on Data Science / Big Data / Analytics, as well as on Software Engineering and Agile Methodologies. Additionally, development methodologies for BDAS (Big Data Analytics Systems) projects were also examined. These topics served as the foundation for constructing the theoretical framework and guiding the remainder of the research, as they revealed the lack of specialized, accessible, and standardized methodologies tailored to the specific needs of BDAS projects.

Thanks to the development of the theoretical framework and its associated literature review, it was determined that several methodologies exist for the development of Big Data Analytics Science (BDAS) projects. However, many authors highlight the lack of methodologies perceived as comprehensive and well-documented. Most of the existing approaches exhibit limitations in terms of scalability, documentation, and the definition of roles, phases, activities, and artifacts. For instance, although CRISP-DM is one of the most widely used and best-documented methodologies for BDAS projects, it lacks clearly defined roles. Similarly, other methodologies, such as ASUM, are considered proprietary and insufficiently documented. These findings, as presented in the theoretical framework, underscore the existing gap in agile, clear, complete, and well-documented methodologies tailored to the specific needs of BDAS projects.

The theoretical framework also focused on the topic of Data Science / Big Data / Analytics, aiming to understand their differences and the main characteristics that distinguish a BDAS (Big Data Analytics Science) project from a traditional one. This section revealed that value can be generated by applying BDAS tools and techniques even in small data projects. Several authors point out that the value derived from using BDAS techniques is comparable regardless of whether the data is large or small. Numerous studies demonstrate the value created within organizations using these techniques. Additionally, the necessary architecture for the development of such projects was analyzed, along with the identification of the best free tools available for implementing BDAS projects.

Finally, the topic of software engineering and agile methodologies was analyzed. This allowed us to understand the components required for a methodology to be considered complete, namely, the inclusion of roles, phases, activities, and artifacts. Additionally, two of the most widely used methodologies worldwide, SCRUM and XP, were examined. Several authors highlight that the use of these methodologies enhances the final quality of the software developed and that they can be effectively applied to data science projects.

Thus, the theoretical framework supported the justification for designing and developing a new methodology for BDAS project development that is agile, easy to use, compatible, useful, and valuable, documented in an Electronic Process Guide (EPG).

7.2 DISCUSSION ON RESEARCH METHODOLOGY

The research strategy followed was structured into six consecutive stages: 1) Identification and justification of the design problem; 2) Establishment of the objectives and constraints for the design of the expected artifact; 3) Creation and development of the artifact; 4) Initial validation through a proof of concept; 5) Formal evaluation of the artifact; and 6) Dissemination of the findings obtained. This methodology, centered on the design-based research approach, facilitated a continuous improvement process of the solution, supported by both theoretical foundations and empirical validations.

The methodology proved effective by integrating theoretical analysis with practical development. The selection of SCRUM and XP methodologies as the foundational reference ensured the agility and control required for BDAS projects, while the adaptation of elements from existing methodologies (CRISP-DM, TDSP, and DDS) enabled the formulation of a flexible and realistic solution that meets the specific needs of BDAS projects. The use of expert judgment and a pilot survey with academics and professionals contributed to the triangulation of results, enhancing the reliability and validity of the findings.

However, this approach presents certain limitations, notably the small sample size used during the validation phase and the potential for bias in the selection of expert participants. Nevertheless, the adopted methodological framework ensured that each stage was aligned with the study's objectives and hypotheses, allowing for a comprehensive and well-founded outcome.

7.3 DISCUSSION ON RESULTS – SOLUTION AND EVALUATIONS

The research findings confirm the existence of a methodological gap in the development of BDAS systems. The rejection of the four null hypotheses (H0.1 to H0.4) underscores the relevance of the proposed solution and its empirical validity.

The design and development of the new AgileDSA (Agile Data Science Analytics Methodology) addresses the main limitations of existing models. It provides an agile, structured, and flexible framework that includes clearly defined roles, phases, activities, and artifacts. Furthermore, the developed Electronic Process Guide (EPG) facilitates the use and implementation of the methodology across various types of organizations, particularly those classified as small or medium-sized enterprises.

The proposed methodology was designed to support small teams and medium-sized organizations aiming to leverage the benefits of Data Science. Additionally, this research highlights the remarkable flexibility available in terms of technologies, architectures, and data volumes, which enables the maximization of value generated through data analysis projects.

The results obtained through the survey reveal that the methodology was positively evaluated in terms of agility, ease of use, compatibility, and added value by both academics and professionals. It is important to highlight that the proposed approach outperformed existing methodologies, reflecting its potential for broader adoption. These findings support not only its theoretical soundness but also its practical applicability in the development of Data Science Analytics systems, particularly in resource-constrained environments.

7.4 DICUSSION ON FUTURE WORK

This research presents several opportunities for future work. First, further validation is needed through longitudinal case studies conducted in both industrial and academic settings. Implementing the methodology in these contexts will provide a more detailed understanding of its adaptability and performance across diverse scenarios.

Another opportunity for future work would be to conduct tests in real production environments, where the methodology is applied to a real-world case study, both in academic and industrial domains. The goal is to confirm that the methodology effectively adapts to the development of BDAS projects within small teams or organizations, and that it can be implemented by experienced teams who can provide feedback based on their comparison with existing methodologies.

Similarly, quality metrics could be developed that align with the methodology proposed in this research and enable the evaluation of various aspects, for example: value delivered per sprint, model interpretability, user satisfaction, among others.

Finally, exploring the integration of ethical and governance considerations related to AI and Big Data (e.g., transparency, data bias, accountability) into the methodology could enhance its relevance within the discourse of contemporary software engineering.

7.5 DISCUSSION ON RESTRICTIONS AND LIMITATIONS

Although this research offers valuable contributions, it is important to acknowledge certain limitations. The evaluation of the methodology was conducted through a pilot survey with a limited number of participants, which may restrict the generalizability of the findings. Despite the participants being selected for their expertise, a broader and more diverse sample would enable a more robust validation.

Another identified limitation pertains to the range of technological platforms considered. The evaluation of the methodology focused primarily on environments utilizing Python and R, which, although widely adopted, do not encompass the full spectrum of technologies used in BDAS projects (such as Scala, Julia, Jupyter, Power BI, Orange, or Tableau).

In summary, this research establishes a solid foundation for the development of lightweight, standards-aligned methodologies within the context of Big Data projects. However, its true potential will be realized through ongoing evaluation and an iterative process of continuous improvement.

8. GLOSSARY

- **Agile Models:** It is not a complete process or an agile methodology, but rather a set of principles and practices to model and perform requirements analysis, complementing most iterative methodologies. Ambler recommends its use with XP, RUP, or any other methodology (ISO/IEC/IEEE 24765:2017, 2017).
- **Agile Software Development:** Software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback (ISO/IEC/IEEE 24765:2017, 2017).
- **Software Development:** Is a programmer or a business company engaged in one or more aspects of the software development process. It is a broader scope of algorithmic programming (ISO/IEC/IEEE 24765:2017, 2017).
- **Software Life Cycle:** Project-specific sequence of activities that is created by mapping the activities of a standard onto a selected software life cycle model (SLCM) (ISO/IEC/IEEE 24765:2017, 2017).**Software Engineering:** Application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software (ISO/IEC/IEEE 24765:2017, 2017).
- **Software Engineering Process:** It is a set of interrelated activities that transform one or more inputs into outputs while consuming resources to achieve the transformation (Bourque et al., 2014).
- **Software:** Computer programs, procedures, and possibly associated documentation and data about the operation of a computer system. (ISO/IEC/IEEE 24765:2017, 2017).
- **Scrum:** Scrum is defined by the Scrum guide itself as: "A lightweight framework that helps people, teams, and organizations to generate value through adaptive solutions to complex problems" (Sutherland & Schwaber, 2020).
- **Product Owner:** *He is responsible for maximizing the value of the product resulting from the Scrum Team's work, that is, defining, prioritizing, and communicating the product requirements. He is the only person responsible for managing the Product Backlog, clearly expressing the elements of the Product Backlog, prioritizing user stories to achieve the objectives and missions in the best way" (Sutherland & Schwaber, 2020).*
- **Scrum Master:** "He is responsible for establishing compliance with the rules and principles of Scrum-based development. The Scrum Master is responsible for the effectiveness of the Scrum Team, helping to eliminate development impediments and improving processes, helping the Scrum Team to improve its practices, within the framework of Scrum. This helps the Product Owner, the Scrum Team,

and the organization by guiding them on iterations that they have with each other, maximizing the value created between them" (Sutherland & Schwaber, 2020).

- **Scrum Team:** *"It consists of professionals who carry out the work of delivering a finished product increment that can potentially be put into production at the end of each sprint. The development team follows the user stories established by the Product Owner to deliver an increment within the established time. The specific skills that developers need are broad and vary by scope of work" (Sutherland & Schwaber, 2020).*
- **Sprint:** *"Defined as the heart of Scrum, it is a block of time of one month or less during which a usable and potentially deployable increment of finished product is created. This event is a container for the rest of the events, this means that the sprint consists of the Sprint Planning, the Daily Scrums, the Sprint Review, and the Sprint Retrospective. Each Sprint has a definition of what will be built, a design, and a flexible plan that will guide its construction, the team's work, and the resulting product" (Sutherland & Schwaber, 2020).*
- **Sprint Planning:** "It is all the work that will be done during the Sprint. This plan is created through the collaborative work of the Scrum Team. Planning a Sprint is a maximum of 8 hours in length for a one-month Sprint. This section answers questions such as: What can be delivered in the resulting increase in the Sprint that begins? And how will you get the work necessary to deliver the increase?" (Sutherland & Schwaber, 2020).
- **Daily Scrum:** *"It is an event that is repeated every day with an approximate duration of 15 minutes, and is aimed at the team's developers, in which the development progress status is communicated and evaluated, improving communication, identifying impediments, promoting streamlining decisions and consequently eliminates the need for other meetings" (Sutherland & Schwaber, 2020).*
- **Data Sciences:** "Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data." (Turkey, 1962).
- **Business Intelligence:** "BI is a broad category of applications, technologies, and processes for collecting, storing, accessing, and analyzing data to help business users make better decisions" (Watson, 2009).
- **Analytics:** *"By analytics we mean the extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based human analysis, ability to drive decisions and actions". (Davenport & Harris, 2007).*
- **Descriptive Analytics:** They are reports like dashboards, data visualization, they have been widely used for some time and are the core applications of traditional BI. Descriptive analyses look back and reveal what happened.

However, one tendency is to include predictive analytics findings, such as future sales forecasts, in dashboards (Watson, 2014).

- **Predictive Analytics:** Suggests what will happen in the future. Methods and algorithms for predictive analytics, such as regression analysis, machine learning, and neural networks, have been around for some time. The ability to analyze new data sources, Big Data, creates additional opportunities for insight and is especially important for companies with large amounts of data. Golden Path analysis is an exciting new technique for predictive analytics. It involves analyzing large amounts of behavioral data (that is, data associated with people's activities or actions) to identify patterns of events or activities that predict customer actions (Watson, 2014).
- **Prescriptive Analytics:** Predict what will happen, prescriptive analysis suggests what to do. Prescriptive analytics can identify optimal solutions, often for scarce resource allocation. It has also been researched in academia for a long time, but now being used more in revenue management, it is becoming more common for organizations that have "perishable" assets such as rental cars, hotel rooms, and airplane seats. For example, Harrah's Entertainment, a leader in the use of analytics, has been using revenue management for hotel room rates for many years (Watson, 2014).
- **Big Data:** *"Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes" (Mills et al., 2012).*
- **Small Data:** "Small data connects people with timely, meaningful insights (derived from big data and/or "local" sources), organized and packaged – often visually – to be accessible, understandable, and actionable for everyday tasks".
- **Volume:** *"Large volume of data that either consumes huge storage or consists of a large number of records" (Russom, 2011).*
- **Variety:** The word ***'Variety'*** denotes the fact that Big Data originates from numerous sources that can be structured, semi-structured, or unstructured (Schroeck et al., 2012).
- **Velocity:** High data quality is an important Big Data requirement for better predictability in the trading environment (Schroeck et al., 2012).
- **Veracity:** High data quality is an important Big Data requirement for better predictability in the trading environment (Schroeck et al., 2012). Therefore, verification is necessary to generate authentic and relevant data and to have the ability to filter incorrect data (Beulke, 2011).
- **Value:** It is the added value obtained by organizations. Value is created only when data is analyzed and acted upon correctly. To do this, we must identify all the data that will help us in the best way to generate value.

- **Python:** Python is a general-purpose object-oriented programming language due to its extensive library that primarily enables the development of Big Data, Artificial Intelligence (AI), Data Science, Test Frameworks, and Web Development applications. Released in 1989, Python is easy to learn and a favorite with programmers and developers. Python is one of the most popular programming languages in the world, second only to Java and C (IBM, 2021).
- **R Language:** R is an Open-Source programming language that is optimized for statistical analysis and data visualization. Developed in 1992, R has a rich ecosystem with complex data models and elegant data reporting tools (IBM, 2021).
- **Java:** Java is an object-oriented programming language specifically designed to allow developers a continuity platform. It is an extremely popular language that runs on a virtual machine, allowing it to be run on any type of device without having to compile it repeatedly. Java was created by Sun Microsystems in 1991, as a programming tool and an object-oriented language, allowing programmers to generate autonomous code fragments, which interact with other objects to solve a problem, offering support for different technologies.
- **Open-Source:** Originally, the expression open source (or open source) referred to open-source software (OSS). Open-source software is code designed in a way that is accessible to the public: everyone can view, modify, and distribute the code in any way they see fit. Open-source software is developed in a decentralized and collaborative manner, so it relies on peer review and community production. In addition, it is usually more economical, flexible, and durable than its proprietary alternatives, since those in charge of its development are the communities and not a single author or a single company (Red Hat, 2021).
- **Architectural Design:** process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system (ISO/IEC/IEEE 24765:2017, 2017).

9. REFERENCES

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile software development methods: A comparative review. *Agile software development*, 31-59.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003, May). New directions on agile methods: a comparative analysis. In *25th International Conference on Software Engineering, 2003. Proceedings.* (pp. 244-254).

Ackoff, R. L. (1962). Scientific method: Optimizing applied research decisions.

Adibuzzaman, M., DeLaurentis, P., Hill, J., & Benneyworth, B. D. (2017). Big data in healthcare–the promises, challenges and opportunities from a research perspective: A case study with a model database. In AMIA Annual Symposium Proceedings (Vol. 2017, p. 384). American Medical Informatics Association.

Agrawal, R., Mehta, M., Shafer, J. C., Srikant, R., Arning, A., & Bollinger, T. (1996, August). The Quest Data Mining System. In *KDD* (Vol. 96, pp. 244-249).

Akhilomen, J. (2013). Data mining application for cyber credit-card fraud detection system. In Advances in Data Mining. Applications and Theoretical Aspects: 13th Industrial Conference, ICDM 2013, New York, NY, USA, July 16-21, 2013. Proceedings 13 (pp. 218-228). Springer Berlin Heidelberg.

Akter, S., Wamba, S. F., Gunasekaran, A., Dubey, R., & Childe, S. J. (2016). How to improve firm performance using big data analytics capability and business strategy alignment?. *International Journal of Production Economics*, *182*, 113-131.

Ambler, S. W., & Lines, M. (2016). The Disciplined Agile Process Decision Framework.

Analytics Solutions Unified Method for Data Mining. IBM, 2015. : https://developer.ibm.com/predictiveanalytics/2015/10/16/have-you-seen-asum-dm/ Dec. 2022.

Banker, S. (2014). Warehouse control in the age of the Internet of Things: if warehouses are to utilize new sensors and intelligence to optimize performance and connect to the enterprise, warehouse management systems and warehouse control systems architectures need to be re-conceptualized. Supply Chain Management Review, 18(5).

Barclay, D. W., C. Higgins and R. Thompson (1995). `The partial least squares (PLS) approach to causal modeling: Personal computer adaptation and use as an illustration', Technology Studies, 2(2), pp. 285± 309.

Bichler, M., Heinzl, A., & van der Aalst, W. M. P. (2016). Business analytics and data science.

Beecham s., T. Hall, C. Britton, M. Cottee, and A. Rainer. Using an expert panel to validate a requirements process improvement model. Journal of Systems and Software, 76(3): 251–275, 2005.

Best, M. (2015). Small Data and Sustainable Development. In International Conference on Communication/Culture and Sustainable Development Goals: Challenges for a new generation (pp. 1-6).

Beulke, D. (2011). Big data impacts data management: The 5 vs of big data. *Available from: Big Data Impacts Data Management: The 5Vs of Big Data, accessed*, *21*.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64-69.

Bourque, P., Fairley, R. E., & IEEE Computer Society. (2014). *Guide to the software engineering body of knowledge*.

Boehm, B., & Turner, R. (2003, June). Observations on balancing discipline and agility. In *Proceedings of the Agile Development Conference, 2003. ADC 2003* (pp. 32-39). IEEE.

Boehm, B., & Turner, R. (2004, May). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Proceedings. 26th International Conference on Software Engineering* (pp. 718-719). IEEE.

Bonde, A. (2013). Defining small data. *Small Data Group*.

Brachman, R. J., & Anand, T. (1996). The process of knowledge discovery in databases. In Advances in knowledge discovery and data mining (pp. 37-57).

Brodie, M. L. (2015, June). Understanding Data Science: An Emerging Discipline for Data Intensive Discovery. In *DAMDID/RCDL* (pp. 238-245).

Burton, S. (2021). Data governance: The path to a data-driven culture. *Applied Marketing Analytics*, *6*(4), 298-308.

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring–A systematic literature review. *Journal of Systems and Software*, *110*, 85-100.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc*, *9*, 13.

Chang, H. C., Wang, C. Y., & Hawamdeh, S. (2019). Emerging trends in data analytics and knowledge management job market: extending KSA framework. *Journal of Knowledge Management*.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 1165-1188.

Chen, Q., Feng, Y., Liu, L., & Tian, X. (2019). Understanding consumers' reactance of online personalized advertising: A new scheme of rational choice from a perspective of negative effects. International Journal of Information Management, 44, 53-64.

Chin, A. G., Harris, M. A., & Brookshire, R. (2018). A bidirectional perspective of trust and risk in determining factors that influence mobile app installation. International Journal of Information Management, 39, 49-59.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. Modern methods for business research, 295(2), 295-336.

Chin, W. W. (2009). How to write up and report PLS analyses. In Handbook of partial least squares: Concepts, methods and applications (pp. 655-690). Berlin, Heidelberg: Springer Berlin Heidelberg.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. Information systems research, 20(3), 329-354.

Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge in society, 1(1), 104.

Costa, C. J., & Aparicio, J. T. (2020, June). POST-DS: A Methodology to Boost Data Science. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6). IEEE.

Cox, M., & Ellsworth, D. (1997, August). Managing big data for scientific visualization. In *ACM siggraph* (Vol. 97, pp. 21-38).

Dåderman, A., & Rosander, S. (2018). Evaluating frameworks for implementing machine learning in signal processing: A comparative study of CRISP-DM, semma and kdd.

Das, M., Cui, R., Campbell, D. R., Agrawal, G., & Ramnath, R. (2015, October). Towards methods for systematic research on big data. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 2072-2081). IEEE.

Davenport, T. (2014). Big data at work: dispelling the myths, uncovering the opportunities. Harvard Business Review Press.

Davenport, T. H. (2006). Competing on analytics. *Harvard business review*, *84*(1), 98.

Davenport, T. H., & Dyché, J. (2013). Big data in big companies. International Institute for Analytics, 3(1-31).

Davenport, T., & Harris, J. (2017). *Competing on analytics: Updated, with a new introduction: The new science of winning*. Harvard Business Press.

Davenport, T. H., Barth, P., & Bean, R. (2012). How'big data'is different.

Davenport, T. H., & Harris, J. G. (2007). The architecture of business intelligence. *Competing on analytics: The new science of winning*.

Davoudian, A., & Liu, M. (2020). Big data systems: a software engineering perspective. *ACM Computing Surveys (CSUR)*, *53*(5), 1-39.

Delen, D., & Ram, S. (2018). Research challenges and opportunities in business analytics. *Journal of Business Analytics*, *1*(1), 2-12.

Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, *55*(1), 412-421.

Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, *56*(12), 64-73.

Dijkman, R. M., Sprenkels, B., Peeters, T., & Janssen, A. (2015). Business models for the Internet of Things. International Journal of Information Management, 35(6), 672-678.

Dudziak, T. (1999). Extreme programming an overview. *Methoden und Werkzeuge der Software: produktion WS*, *2000*, 2000.

Dybå. T. An instrument for measuring the key factors of success in software process improvement. Empirical software engineering, 5(4):357–390, 2000.

F. Tripp, J., & Armstrong, D. J. (2018). Agile methodologies: organizational adoption motives, tailoring, and performance. *Journal of Computer Information Systems*, *58*(2), 170-179.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*(8), 28-35.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1991). KDD: An Overview. *AI Magazine*, *14*(3), 5.

Galvan, S., Mora, M., & Laporte, C. Y. (2021). Reconciliation of scrum and the project management process of the ISO/IEC 29110 standard-Entry profile—An experimental evaluation through usability measures | SpringerLink. https://link.springer.com/article/10.1007/s11219-021-09552-3

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, *35*(2), 137-144.

Gartner, 2014. Gartner survey reveals that 73 percent of organizations have invested or plan to invest in Big Data in the next two years. Press Release, September 17th http://www.gartner.com/newsroom/id/2848718.

Gartner, Gartner Top 10 Data and Analytics Trends for 2021, Retrived from: https://www.gartner.com/smarterwithgartner/gartner-top-10-data-and-analytics-trends-for-2021

Gentile, B. (2012). Top 5 myths about big data.

Grady, N. W., Payne, J. A., & Parker, H. (2017, December). Agile big data analytics: AnalyticsOps for data science. In *2017 IEEE international conference on big data (big data)* (pp. 2331-2339). IEEE.

Greeno, J. G., & Simon, H. A. (1988). Problem solving and reasoning. John Wiley & Sons.

Halper, F. (2015). Next-Generation Analytics and Platforms for Business Success. TDWI Research Report. Retrieved from: www.tdwi.org

Hayashi, C. (1998). What is data science? Fundamental concepts and a heuristic example. In *Data science, classification, and related methods* (pp. 40-51). Springer, Tokyo.

Heller, B., & Röthlisberger, M. (2015). Big data on trial: Researching syntactic alternations in GloWbE and ICE. In *From Data to Evidence (d2e), Date: 2015/10/19-2015/10/22, Location: Helsinki*.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, *19*(2), 4.

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology*, *85*, 60-70.

IBM, 2021 Python vs R: What´s the Difference? Retrieved from: https://www.ibm.com/cloud/blog/python-vs-r

"ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary," in ISO/IEC/IEEE 24765:2017(E), vol., no., pp.1-541, 28 Aug. 2017, doi: 10.1109/IEEESTD.2017.8016712.

Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM*, *52*(8), 36-44.

Jarr, S. (2015). Fast Data and the New Enterprise Data Architecture. O'Reilly Publishing.

Jyothi, V. E., & Rao, K. N. (2011). Effective implementation of agile practices. *International Journal of Advanced Computer Science and Applications*, *2*(3).

Karahanna E., D. W. Straub, and N. L. Chervany. Information technology adoption across time: a cross-sectional comparison of pre-adoption and post-adoption beliefs. MIS quar-terly, pages 183–213, 1999.

Katsis, Y., Balac, N., Chapman, D., Kapoor, M., Block, J., Griswold, W. G., ... & Patrick, K. (2017, July). Big data techniques for public health: a case study. In 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE) (pp. 222-231). IEEE.

Kdnuggets, 2019 Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis Retrived from: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html

Keen, P. G., & Morton, S. (1978). MS (1978). *Decision support systems: An organizational perspective*, 264.

KENDALL, J. E., & KENDALL, K. E. (2004). Agile methodologies and the lone systems analyst: When individual creativity and organizational goals collide in the global IT environment. *Journal of Individual Employment Rights*, *11*(4).

Kim, S., & Park, H. (2013). Effects of various characteristics of social commerce (s-commerce) on consumers' trust and trust performance. International journal of information management, 33(2), 318-332.

Kitchin, R. (2013). Big data and human geography: Opportunities, challenges and risks. *Dialogues in human geography*, *3*(3), 262-267.

Kitchin, R. (2014). *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage.

Kitchin, R., & Lauriault, T. P. (2015). Small data in the era of big data. *GeoJournal*, *80*(4), 463-475.

Kröckel, J., & Bodendorf, F. (2012). Visual customer behavior analysis at the point of sale. International Journal on Advances in Systems and Measurements, 5(3).

Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. International journal of information management, 34(3), 387-394.

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. META group research note, 6(70), 1.

Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics, and data science. *International Journal of Information Management*, *36*(5), 700-710.

Lee, T., Lee, H., Rhee, K. H., & Shin, U. S. (2014). The efficient implementation of distributed indexing with Hadoop for digital investigations on Big Data. *Computer Science and Information Systems*, *11*(3), 1037-1054.

Lee D., J. Park, and J.-H. Ahn. On the explanation of factors affecting e-commerce adoption. ICIS 2001 Proceedings, page 14, 2001.

Leybourn, E. (2013). *Directing The Agile Organisation: A lean approach to business management*. IT Governance Ltd.

Li, H., Jiang, J., & Wu, M. (2014). The effects of trust assurances on consumers' initial online trust: A two-stage decision-making process perspective. International Journal of Information Management, 34(3), 395-405.

Lin, Y. T., & Huang, S. J. (2018). The design of a software engineering lifecycle process for big data projects. IT Professional, 20(1), 45-52.

Liu, L., Lee, M. K., Liu, R., & Chen, J. (2018). Trust transfer in social media brand communities: The role of consumer engagement. International Journal of Information Management, 41, 1-13.

Mariscal, G., Marban, O., & Fernandez, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review, 25(2), 137-166.

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Orallo, J. H., Kull, M., Lachiche, N., ... & Flach, P. A. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*.

Marx, V. (2013). The big challenges of big data. Nature, 498(7453), 255-260.

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, *40*(1), 1-6.

McClure, R. M. (1968). *NATO SOFTWARE ENGINEERING CONFERENCE 1968*. 136.

Microsoft, 2021 Azure Application Architecture Guide Retrived from: https://docs.microsoft.com/en-us/azure/architecture/browse/

What is the team data science process?. Microsoft, 2023. https://docs.microsoft.com/enus/azure/machinelearning/teamdatascienceprocess/overview Nov. 2023.

M. Mora (2009). Conceptual Research Method: a Description (in Spanish language). Technical Report UAA-DSI-01, Autonomous University of Aguascalientes, Mexico.

Miller, H. J. (2010). The data avalanche is here. Shouldn't we be digging?. Journal of Regional Science, 50(1), 181-201.

Mills, S., Lucas, S., Irakliotis, L., Rappa, M., Carlson, T., & Perlowitz, B. (2012). Demystifying big data: a practical guide to transforming the business of government. *TechAmerica Foundation, Washington*.

Moine, J. M., Gordillo, S. E., & Haedo, A. S. (2011). Análisis comparativo de metodologías para la gestión de proyectos de minería de datos. In *Congreso Argentino de Ciencias de la Computación* (Vol. 17).

Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information systems research*, *2*(3), 192-222.

Mora, M., Adelakun, O., Reyes-Delgado, P. Y., & Diaz, O. (2023). AVS_FD_MVITS: an agile IT service design workflow for small data centers. The Journal of Supercomputing, 79(15), 17519-17561.

Mora, M., Gómez, J. M., O'Connor, R. V., & Gelman, O. (2016). An MADM risk-based evaluation-selection model of free-libre open source software tools. International Journal of Technology, Policy and Management, 16(4), 326-354.

Mora, M., Steenkamp, A. L., Gelman, O., & Raisinghani, M. S. (2012). On IT and SwE Research Methodologies and Paradigms: A Systemic Landscape Review. In *Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems* (pp. 149-164). IGI Global.

Naur, P. (1974). *Concise survey of computer methods*. Petrocelli Books.

New Vantage (2019), NewVantage Partners 2019 Big Data and AI Executive Survey (2019).

Newell, A., & Simon, H. A. (1972). Human problem solving (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-hall.

Niazi M., D. Wilson, and D. Zowghi. A framework for assisting the design of effective software process improvement implementation strategies. Journal of Systems and Software, 78(2):204–222, 2005.

Núñez, A., Hendriks, J., Li, Z., De Schutter, B., & Dollevoet, R. (2014, October). Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study. In *2014 ieee international conference on big data (big data)* (pp. 48-53). IEEE

O'Neil, C., & Schutt, R. (2013). *Doing data science: Straight talk from the frontline*. " O'Reilly Media, Inc.".

Oestereich, B., & Schröder, C. (2017). *Das kollegial geführte Unternehmen: Ideen und Praktiken für die agile Organisation von morgen*. Vahlen.

Olson, D., & Delen, D. (2008). Schematic of SEMMA. *Data mining Techniques*, *19*.

Oktaba, H., & Ibargüengoitia González, G. (1998). Software process modeled with objects: Static view. *Computación y Sistemas, 1*(4), 228-238.

Palfreyman, J. (2013). Big Data–Vexed by Veracity?.

Palomo, E. J., Elizondo, D., Domínguez, E., Luque, R. M., & Watson, T. (2012). SOM-based techniques towards hierarchical visualisation of network forensics traffic data. Computational intelligence for privacy and security, 75-95.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of management information systems, 24(3), 45-77.

Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects. *KDD News*.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, 229-238.

Pressman, R. S. (2015). *Software engineering: A practitioner's approach* (Eighth edition). McGraw-Hill Education.

Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big data*, *1*(1), 51-59.

Purtova, N. (2011). Property in personal data: second life of an old idea in the age of cloud computing, chain informatisation, and ambient intelligence. In Computers, privacy and data protection: an element of choice (pp. 39-64). Dordrecht: Springer Netherlands.

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and Software Technology, 50(4), 280–295. https://doi.org/10.1016/j.infsof.2007.02.002

Qumer, A., & Henderson-Sellers, B. (2006, December). Crystallization of agility: back to basics. In *ICSOFT 2006-1st International Conference on Software and Data Technologies, Proceedings*.

Ravizza, S. M., Hambrick, D. Z., & Fenn, K. M. (2014). Non-academic internet use in the classroom is negatively related to classroom learning regardless of intellectual ability. Computers & Education, 78, 109-114.

Reyes-Delgado, P. Y., Mora, M., Duran-Limon, H. A., Rodríguez-Martínez, L. C., O'Connor, R. V., & Mendoza-Gonzalez, R. (2016). The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review. Computer Standards & Interfaces, 47, 24-41.

Rich S., 2012. Big Data Is a "New Natural Resource" Retrived from: http://www.govtech.com/policy-management/Big-Data-Is-a-New-Natural-Resource-IBM-Says.html

Robbes, R., Vidal, R., & Bastarrica, M. C. (2013). Are software analytics efforts worthwhile for small companies? The case of Amisoft. IEEE software, 30(5), 46-53.

Rodríguez, L. C., Mora, M., Martin, M. V., O'Connor, R., & Alvarez, F. (2009). Process models of SDLCs: comparison and evolution. In *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (pp. 76-89). IGI Global.

Runkler, T. A. (2020). *Data analytics*. Springer Fachmedien Wiesbaden.

Runkler, T. A. (2020). Data Visualization. In *Data Analytics* (pp. 37-59). Springer Vieweg, Wiesbaden.

Russo, D., & Stol, K. J. (2021). PLS-SEM for software engineering research: An introduction and survey. ACM Computing Surveys (CSUR), 54(4), 1-38.

Russom, P. (2011). Big data analytics. *TDWI best practices report, fourth quarter*, *19*(4), 1-34.

Saltz J., Data Driven Scrum. 2022. https://www.datasciencepm.com/datadrivenscrum/

Saltz, J., Hotz, N., Wild, D., & Stirling, K. (2018). Exploring project management methodologies used within data science teams. In *24th Americas Conference on Information Systems 2018: Digital Disruption, AMCIS 2018*. Association for Information Systems.

Saltz, J. S. (2015, October). The need for new processes, methodologies and tools to support big d data project effectiveness. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 20

Saponara, S., & Bacchillone, T. (2012). Network architecture, security issues, and hardware implementation of a home area network for smart grid. Journal of Computer Networks and Communications, 2012(1), 534512.

Sargent. R. G. An introduction to verification and validation of simulation models. In Simulation Conference (WSC), 2013 Winter, pages 321–327. IEEE, 2013.

Sargent. R. G. Verification, validation, and accreditation: verification, validation, and accreditation of simulation models. In Proceedings of the 32nd conference on Winter simulation, pages 50–59. Society for Computer Simulation International, 2000.

SAS Institute Inc. 2017. SAS® Enterprise Miner™ 14.3: Reference Help. Cary, NC: SAS Institute Inc.

Sawant, N., & Shah, H. (2013). Big data application architecture. In *Big data Application Architecture Q & A* (pp. 9-28). Apress, Berkeley, CA.

Shea, Rick. 2014. Big Data, Small Data. http://www.optiv8.com/approach/big-data-small-data/ (Access 27 October 2015)

Sheskin, D. J. (2000). Handbook of parametric and nonparametric statistical procedures. Chapman and hall/CRC.

Song, I. Y., & Zhu, Y. (2016). Big data and data science: what should we teach?. *Expert Systems*, *33*(4), 364-373.

Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufano, P. (2012). Analytics: The real-world use of big data. *IBM Global Business Services*, *12*(2012), 1-20.

Schwaber, K. (1997). Scrum development process. In *Business object design and implementation* (pp. 117-134). Springer, London.

Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217-222.

Shaw, M. (2003, May). Writing good software engineering research papers. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 726-736). IEEE.

Song, Y., Schreier, P. J., Ramírez, D., & Hasija, T. (2016). Canonical correlation analysis of high-dimensional data with very small sample support. *Signal Processing*, *128*, 449-458.

Song, I. Y., & Zhu, Y. (2016). Big data and data science: what should we teach?. *Expert Systems*, *33*(4), 364-373.

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, *94*, 87-97.

Stobierski, T. (2021). How to Structure Your Data Analytics Team. Harvard Business School Online.

Sutherland, J., & Schwaber, K. (2020). The scrum guide. *The definitive guide to scrum: The rules of the game. Scrum. org*, *268*.

Thinyane, M. (2017, June). Investigating an Architectural Framework for Small Data Platforms. In *Proceedings of the 17th European Conference on Digital Government (ECDG 2017), Lisbon, Portugal* (pp. 220-227).

Tsai, C. W., Lai, C. F., & Vasilakos, A. V. (2014). Future internet of things: open issues and challenges. Wireless Networks, 20, 2201-2217.

Tukey, J. W. (1962). The future of data analysis. *The annals of mathematical statistics*, *33*(1), 1-67.

VentureBeat (2019), Why do 87% of data science projects never make it into production? 2019.

Viswanathan, V. (2014). *Data Analytics with R: A Hands-on Approach*. Infivista Incorporated.

Vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to Design Science Research. In Design Science Research. Cases (pp. 1-13). Springer, Cham.

Wang, Y., & Yu, C. (2017). Social interaction-based consumer decision-making model in social commerce: The role of word of mouth and observational learning. *International Journal of Information Management*, *37*(3), 179-189.

Watson, H. J. (2014). Tutorial: Big data analytics: Concepts, technologies, and applications. *Communications of the Association for Information Systems*, *34*(1), 65.

Watson, H. J. (2009). Tutorial: business intelligence–past, present, and future. *Communications of the Association for Information Systems*, *25*(1), 39.

Watters, C. A., Keefer, K. V., Kloosterman, P. H., Summerfeldt, L. J., & Parker, J. D. (2013). Examining the structure of the Internet Addiction Test in adolescents: A bifactor approach. Computers in Human Behavior, 29(6), 2294-2302.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., ... & Wesslén, A. (2012). Empirical strategies. Experimentation in Software Engineering, 9-36.

Wysocki, R. K. (2009). *Effective project management: traditional, agile, extreme*. John Wiley & Sons.

Yan, Y., Huang, C., Wang, Q., & Hu, B. (2020). Data mining of customer choice behavior in internet of things within relationship network. International Journal of Information Management, 50, 566-574.

Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: a survey. Journal of Big Data, 2, 1-41.

10. APPENDICES

10.1 SELECTIVE SEARCH.

Table 10.1 Set of 7 studies on BDAS Development Life Cycles.

Type of PAIS/|BPMS Life Cycle	Publication Domain	Publication Name	Type of Publication	Publication IF	Publication Year	Name of SDLC	Citations
Heavyweight	Analytics Data Science	AI Magazine	JCR journal	2.524	1996	KDD: Knowledge Discovery in Databases	12666
Heavyweight	Analytics Data Science	SAS institute Web Site	Grey Literature	-	1996	SEMMA: Sample, Explore, Modify, Model, and Assess	8
Heavyweight	Analytics Data Science	SPSS Inc. Web Site	Grey Literature	-	2000	CRISP-DM: Cross Industry Standard Process for Data Mining	2017
Heavyweight	Software Engineering	IEEE IT PROF	JCR journal	2.590	2018	BDPL: Big Data Project Lifecycle	12
Agile	Analytics Data Science	IBM Analytics Web Site	Grey Literature	-	2015	ASUM-DM: Analytics Solutions Unified Method	3
Agile	Analytics Data Science	Microsoft Azure Web Site	Grey Literature	-	2016	TDSP: The Team Data Science Process	19
Agile	Analytics Data Science	Data Driven Scrum Web Site	Grey literature	-	2022	DSS: Data Driven Scrum	-

10.2 A PRO FORMA OF AN AGILE SDLC FOR BDAS (FROM SCRUM AND XP)

Table 10.2 Pro forma of the agile Scrum-XP SDLC for BDAS.

SDLC element	SDLC element description
Roles (3)	**User roles:** {**R.1** Scrum-XP product owner}. **Management roles:** {**R.2** Scrum-XP master}. **Technical roles**:{ **R.3** Scrum-XP development team}.
Phases-Activities (6, 13)	**Pre-Game Phases:** **Phase.1 Product Exploration:** To obtain the user requirements through the initial (no prioritized) and final (already prioritized) full product backlog(user stories) work product. If required, to explore empirically a Spike. **Activities:** {ACT.1 Product vision declaration. ACT.2 Product backlog(user stories) elaboration and prioritization. ACT.3 Spikes exploration (if required).} **Phase.2 Product Release Planning:** To elaborate an agreed product backlog(user stories) development plan. **Activities:** {ACT.4 Product backlog(user stories) development planning.}.
	Game Phases: **Phase.3 Sprint-Iteration Planning:** To elaborate an agreed Sprint-Iteration backlog(user stories) development plan. **Activities:** {ACT.5 Sprint-Iteration backlog(user stories) development planning.}. **Phase.4 Sprint-Iteration Development:** To sketch a simple architectural design supported by the current Sprint-Iteration backlog(user stories), build the Sprint-Iteration backlog(user stories), and elaborate and apply the user acceptance and functional tests. **Activities**: {ACT.6 Simple architectural design. ACT.7 Daily Scrum-XP meeting. ACT.8 User acceptance tests elaboration. ACT.9 Technical tests elaboration. ACT.10 Increment building, testing and integration.}. **Phase.5 Sprint-Iteration Review and Retrospective:** To conduct the Sprint-Iteration review and retrospective. **Activities:** {ACT.11 Sprint-Iteration review. ACT.12 Sprint-Iteration retrospective.}.
	Post-Game Phase: **Phase.6 Product Release:** To deliver the final WP.14 Software product release. **Activities:** {ACT.13 Product release delivery.}.
Artifacts (15)	**Pre-Game Phases:** **Phase.1 Product Exploration:** {WP.1 Product vision statement. WP.2 Product backlog(user stories). WP.3 Spike records (if used).}. **Phase.2 Product Release Planning:** {WP.4 Product backlog(user stories) development plan.}.
	Game Phases: **Phase.3 Sprint-Iteration Planning:** {WP.5 Sprint-Iteration backlog(user stories) development plan.}. **Phase.4 Sprint-Iteration Development:** {WP.6 Simple architectural design. WP.7 Daily Scrum-XP 3-question record. WP.8 Kanban board. WP.9 Burndown chart. WP.10 User acceptance tests. WP.11 Technical functional tests. WP.12 Sprint-Iteration software increment. WP.13 Sprint-Iteration software build.}. **Phase.5 Sprint-Iteration Review and Retrospective:** {WP.14 Sprint-Iteration review record.}.
	Post-Game phase: **Phase.6 Product Release:** {WP.15 Software product release.}.

10.3 DESIGN OF THE ARTIFACT METHODOLOGY.

Once the theoretical design sources were selected, design components were chosen among roles, activities, and artifacts that could aid in the design of the BDAS methodology. Tables 10.1, 10.2, and 10.3 show the first and second iterations conducted to generate the selected design components for the BDAS-type methodology. The third iteration is represented in Chapter 4.4 of this research, which corresponds to the proposal of the BDAS methodology.

The first iteration, shown in Table 10.1, displays all the design components that the working team considered heuristically and based on their experience from 4 of the methodologies evaluated in this research. This was done to select suitable and necessary design components for creating a new BDAS methodology based on Scrum-XP.

Once the design components from the 4 evaluated methodologies are represented, the second iteration thoroughly reviews each component individually. The working team studies, analyzes, evaluates, and questions the importance of each evaluated component to be subsequently implemented in the proposed BDAS methodology. The design components considered most relevant and best suited for the BDAS methodology will be implemented in the second iteration, resulting in Table 10.2.

The third and final iteration re-examines, analyzes, and evaluates the design components proposed in the second iteration, selecting the minimum essential design components for the BDAS methodology. This aims to create an agile methodology for BDAS projects that is easy to use, useful, compatible, and adds value to small and medium-sized enterprises.

The selection and evaluation of the design components from each of the evaluated methodologies were conducted heuristically and at the discretion of the team members based on the selective review mentioned in Chapter 3 of this research. It is also important to mention that many of the design components for the BDAS methodology were based on the proposal from DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).

Table 10.3 Roles for Desing Components first and second iteration.

Roles										
Design Component	**Source**	**Name**	**Why this could be helpful**	**SDLC that is also using it**				**Iteration**		
				DTS.1	**DTS.2**	**DTS.3**	**DTS.4**	**1**	**2**	**3**
DC.4 Scrum-XP Roles	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	• Customer-Product Owner • Coach-Master • Development Team	**Customer-Product Owner**: The closest role to the stakeholders, this is the person who knows how to provide value to the project.		X		X	X	X	X
			Coach-Master: The person who is in charged to remove all the obstacles, coaching the team, ensuring the transparency, and promoting the self-organization.		X	X	X	X	X	X
			Development Team: The cross-functionality team who is able to build the increment every sprint. It is self-organized.		X	X	X	X	X	
DC.8 TDSP Roles	DTS.3 TDSP (Microsoft, 2107).	• Group manager, • Team lead, • Project lead, • Project individual contributors	**Group manager:** Manages the entire data science unit in an enterprise.			X		X		
			Team lead: Manages a team in the data science unit of an enterprise.		X	X	X	X	X	
			Project lead: Manages the daily activities of individual data scientists on a specific data science project.		X	X	X	X	X	
			Project individual contributors: Data scientists, business analysts, data engineers, architects, and others who execute a data science project.		X	X	X	X	X	
DC.12 DDS Roles	DTS.4 DDS (Saltz, 2022).	• Product Owner • Process Expert • DDS Team Members	**Product Owner:** The Product Owner in DDS is the empowered central point of product leadership ("voice of the client")		X		X	X	X	
			Process Expert: The Process Expert acts as a coach, facilitator, and impediment remover.		X	X	X	X	X	
			DDS Team Members: The DDS team is comprised of a cross-functional collection of DDS Team Members.		X	X	X	X	X	X

Table 10.4 Phases and Activities for Desing Components first and second iteration.

Phases and Activities										
Design Component	Source	Name	Why this could be helpful	SDLC that is also using it				Iteration		
				DTS.1	DTS.2	DTS.3	DTS.4	1	2	3
DC.1 CRISP-DM Phases	DTS.1 CRISP-DM (Chapman et al., 2000).	• Business Understanding • Data Understanding • Data Preparation • Modeling • Evaluation • Deployment	**Business Understanding:** In the initial stage, we focus on understanding the project's objectives and requirements.	X	X	X		X		
			Data Understanding: This stage begins with information gathering and continues with actions to delve deeper into the data.	X		X		X	X	X
			Data Preparation: This phase encompasses all actions aimed at creating the final dataset from the raw dataset.	X		X		X	X	X
			Modeling: During this phase, various modeling techniques are chosen and applied. Typically, there are several methods to address the same type of data science problem.	X		X		X	X	
			Evaluation: Before proceeding with the final implementation of the previously created model, it is crucial to perform comprehensive evaluations of the developed model.	X				X		
			Deployment: This stage varies according to the requirements of the data science project and can range from generating reports to implementing a repeatable data mining process.	X	X	X		X		
DC.2 CRISP-DM Activities	DTS.1 CRISP-DM (Chapman et al., 2000).	• Determine Data Mining Goals • Collect initial Data • Describe Data • Explore Data • Verify Data Quality • Select Data • Clean Data	**Business Understanding - Determine Data Mining Goals:** This activity establishes the project's objectives in technical terms.	X				X	X	
			Data Understanding - Collect initial Data: This process involves acquiring datasets, the location where they are stored, and the methods used to acquire them.	X		X		X	X	X
			Data Uderstanding - Describe Data: Its objective is to examine the "raw" or	X		X		X		

		- Construct Data - Integrate Data - Format Data - Select Modeling Techniques - Build Model - Assess Model	"superficial" properties of the acquired data and report the results.							
			Data Understanding - Explore Data: Data exploration helps address data extraction issues considering assumptions and their impact on the rest of the project.	X		X		X		
			Data Understanding - Verify Data Quality: In this phase, questions such as "Are the data complete (covering all necessary cases)?" "Are they correct or do they contain errors, and if so, how often?" "Are there missing values in the data? If so, how are they represented, where do they occur, and how often?" are addressed.	X		X		X	X	
			Data Preparation - Select Data: In this phase, the data to be used for analysis will be decided.	X				X	X	
			Data Preparation - Clean Data: The main objective of this activity is to improve data quality, representativeness, and impartiality.	X		X		X	X	X
			Data Preparation - Construct Data: Data construction is the process of developing new records or producing derived attributes.	X				X	X	
			Data Preparation - Integrate Data: This stage provides methods by which information from various tables or records is combined to create new records or value scores.	X				X		
			Data Preparation - Format Data: It focuses on syntactic modifications made to the data without changing its meaning.	X				X		
			Modeling - Select Modeling Techniques: Specific modeling techniques are selected to be applied to the datasets. Different modeling techniques can be applied to the same dataset.	X		X		X	X	X
			Modeling - Build Model: Selected models are implemented and parameterized on the prepared dataset.	X		X		X	X	

			Modeling - Assess Model: Model evaluation focuses on interpreting the model based on quality metrics, project success criteria, desired test design, and data science results in the business context.	X		X		X	X	X
DC.5 Scrum-XP Phases	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	• Exploration • Product Planning • Iteration-Sprin • Planning • Iteratio • Sprint, Produc • Release	**Exploration**: Plan all the project and identify the projects needs.	X	X	X		X	X	X
			Product Planning: Plan the product according the needs.		X		X	X	X	
			Iteration-Sprint Planning: Select the activities the provide more value to the project as priority to be developed during a fixed time.		X			X	X	
			Iteration-Sprint: Build the increment in a Iterative process.		X		X	X	X	X
			Product Release: Release the increment with the most important features choosed by the Owner.	X	X	X		X	X	X
DC.6 Scrum-XP Activities	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999)	• Product vision definition • Product backlog (user story set) definition • Product backlog (user story set) prioritization • Spike testing • Product backlog (user story set) effort estimation • Product backlog (user story set) negotiation • Style codifying standard definition • Iteration-sprint user story selection	**Exploration - Product vision definition**: To Have a clear vision of the product and what need to be developed.	X	X	X		X	X	X
			Exploration - Product backlog definition: Create the user stories or tasks that need to be developed.		X		X	X	X	X
			Exploration - Product backlog prioritization: Set the user stories to prioritize the tasks for the one that provide more value.		X		X	X	X	X
			Exploration - Spike testing: Define the spikes that need some effort to have a better knowledge to close the spike and create the needed user stories.		X			X		
			Product Planning - Product backlog effort estimation: Estimate every single user stories by the developer, it is possible to use fixed time or user stories points (recommended).		X		X	X	X	X
			Product Planning - Product backlog negotiation: Negotiate as needed in some		X			X	X	

		• Iteration sprint user story task planning • Iteration-sprint user story plan negotiation • Stand-up meeting • Customer functional tests elaboration • Simple design • Codification and unit testing • Increment integration and customer functional testing • Iteration-sprint review and retrospective • Product releasing	user stories. Negotiations with the product owner can avoid conflicts during the sprint.							
			Product Planning - Style codifying standard definition: Define standards in the code could help to create a better product and more maintainable in the feature.		X			X		
			Iteration Sprint Planning - User story selection: Select the most valuable user stories to be developed during the sprint by the Product Owner. The development team choose the task according to their skills.		X		X	X	X	
			Iteration Sprint Planning - User story task planning: Planning the user story selected in terms what would be the best approach for done this task.		X		X	X		
			Iteration Sprint Planning - User story plan negotiation: Negotiate with product owner some items for the Sprint Planning		X			X	X	
			Iteration Sprint - Stand-up meeting: Meet with the team to talk about the progress, the upcoming work and any block that can have.		X		X	X	X	X
			Iteration Sprint - Customer functional tests elaboration: Elaborate test cases for every single user story that is developed.		X			X		
			Iteration Sprint - Simple design: Create a simple design of how to develop the story.		X			X	X	
			Iteration Sprint - Codification and unit testing: Code and test the selected user story.		X			X	X	
			Iteration Sprint - Increment integration and customer functional testing: Merge the finished users stories with increment that is a working version of the product with the functionality described in the developed user stories.		X		X	X	X	
			Iteration Sprint - Review and retrospective: Conduct a retrospective by all the team to know how what is working,		X		X	X	X	X

			what is not. and how to be better in the next sprints.							
			Product Release - Product releasing: Release the increment.	X	X	X		X	X	X
DC.9 TDSP Phases	DTS.3 TDSP (Microsoft, 2107).	• Business Understanding • Data Acquisition and Understanding • Modeling • Deployment • Customer Acceptance	**Business Understanding:** The objective of this phase is to identify the main variables that will serve as model objectives.	X	X	X		X		
			Data Acquisition and Understanding: In this phase, a clean and high-quality dataset is generated, and the data architecture solution is developed.	X		X		X	X	X
			Modeling: The data for the learning model is determined, and a machine learning model is created.	X		X		X	X	
			Deployment: In this phase, the models with data pipelines are implemented in a production environment.	X		X		X		
			Customer Acceptance: The aim of this phase is to ensure the model and its implementation meet all customer requirements.	X	X	X		X	X	
DC.10 TDSP Activities	DTS.3 TDSP (Microsoft, 2107).	• Define Objective • Identify Data Source • Ingest Data • Explore the Data • Set up a Data Pipeline • Feature Engineering • Model Training • Model Evaluation • Operationalize a Model • System Validation • Project hand-off	**Business Understanding - Define Objective:** The main objective is to identify the project's goals by interacting with the client and formulating core questions that data science can address.	X	X	X		X	X	X
			Business Understanding - Identify Data Source: The required datasets for the BEDS that can help answer the Client's queries are defined	X		X		X	X	X
			Data Acquisition and Understanding - Ingest Data: Data is moved from source locations to destination locations where analysis operations are performed.	X		X		X	X	X
			Data Acquisition and Understanding - Explore the Data: Datasets are explored and processed to remove noise, discrepancies, or missing data.	X		X		X	X	X
			Data Acquisition and Understanding - Set up a Data Pipeline: The data ingestion architecture is specified based on business	X		X		X	X	X

			needs and constraints (batch mode, streaming, real-time, or hybrid).							
			Modeling - Feature Engineering: TDSP provides a methodological guide for selecting the most appropriate model (referred to as the Machine Learning Algorithm Reference Sheet).	X		X		X	X	
			Modeling - Model Training: In this part, machine learning models are trained and calibrated.	X		X		X	X	X
			Modeling - Model Evaluation: This activity determines whether the trained and calibrated statistical/machine learning model produces results with a level of validity suitable for use in production.	X		X		X	X	X
			Deployment - Operationalize a Model: The main objective of this activity is the implementation of the model and the pipeline in a production or similar environment for application consumption.	X		X		X		
			Customer Acceptance - System Validation: The aim of this phase is to ensure the model and its implementation meet all customer requirements.	X	X	X		X	X	
			Deployment - Project hand-off: Handing over the project to the entity that will execute the system in production.	X	X	X		X	X	X
DC.13 DDS Phases	DTS.4 DDS (Saltz, 2022).	• Brainstorm • Prioritize • Create / Refine • Observe & analyze	**Brainstorm:** Teams exchange ideas about potential questions to answer or experiments to conduct.				X	X		
			Prioritize: The team prioritizes these questions and selects the highest-priority item to work on, involving the identification of data to be used and the models to be created.				X	X	X	
			Create / Refine: Involves the team collectively interpreting their work's results.		X		X	X	X	
			Observe & analyze: The team implementing the results and prioritizing future work.				X	X		

DC.14 DDS Activities	DTS.4 DDS (Saltz, 2022).	• Backlog Refinement • Prioritization of the Backlog • Iterations • Iteration Duration • Product Increments • Backlog Item Selection • Daily Meeting • Iteration Review • Retrospective	**Brainstorm - Backlog Refinement:** In addition to the DDS Team working on one or more iterations, the team also spends time evaluating the Backlog Items so they can be prioritized.		X		X	X	X	
			Prioritize - Prioritization of the Backlog: The team explores the Items in their Backlog by providing high level estimates of: (1) the value of the work, (2) the amount of work (team effort), and (3) the probability of success of that work.		X		X	X	X	X
			Create / Refine - Iterations: An Iteration is a collection of one or more backlog items.		X		X	X	X	
			Create / Refine - Iteration Duration: Each iteration is capability-based (not time-boxed calendar events). Furthermore, each iteration should aim to be a minimally viable set of work that can deliver value.				X	X	X	X
			Create / Refine - Product Increments: A high-level goal for the team to achieve in a fixed amount of time (ex. 3 months) using multiple iterations is known as a Product Increment.		X		X	X	X	X
			Observe & analyze - Backlog Item Selection: Occurs when the team has capacity to start a new iteration (e.g., when a previous iteration has completed).		X		X	X	X	
			Observe & analyze - Daily Meeting: Occurs each workday, when the team meets for a 15-minute inspect-and-adapt activity.		X		X	X	X	X
			Observe & analyze - Iteration Review: Reviews might be weekly and are calendar based to account for the fact that there might be several iterations per week, and there would be diminishing returns if iteration reviews occurred on a daily.		X		X	X	X	X
			Observe & analyze - Retrospective: Occurs at regular intervals (ex. once a month) and is a time to inspect and adapt the process.		X		X	X	X	X

Table 10.5 Artifacts for Desing Components first and second iteration.

Processes Artifacts										
Design Component	Source	Name	Why this could be helpful	SDLC that is also using it				Iteration		
				DTS.1	DTS.2	DTS.3	DTS.4	1	2	3
DC.3 CRISP-DM Artifacts	DTS.1 CRISP-DM (Chapman et al., 2000).	• Data mining goals • Data Mining Success Criterial • Initial Data Collection Report • Data Description Report • Data Exploration Report • Data Quality Report • Data Cleaning Report • Merged Data • Reformatted Data • Dataset • Dataset Description • Modeling Technique • Models • Model Assessment • Assessment of Data Mining Results	**Business Understanding - Data mining goals:** Describe the intended outputs of the project that enables the achievement of the business objectives.	X				X	X	X
			Business Understanding - Data Mining Success Criterial: Define the criteria for a successful outcome to the project in technical terms.	X		X		X		
			Data Understanding - Initial Data Collection Report: List the dataset (or datasets) acquired, together with their locations within the project, the methods used to acquire them and any problems encountered.	X		X		X	X	
			Data Understanding - Data Description Report: Describe the data which has been acquired, including: the format of the data, the quantity of data.	X		X		X		X
			Data Understanding - Data Exploration Report: Describe results of this task including first findings or initial hypothesis and their impact on the remainder of the project.	X		X		X		
			Data Understanding - Data Quality Report: List the results of the data quality verification; if quality problems exist, list possible solutions.	X		X		X	X	X
			Data Preparation - Data Cleaning Report: Describe what decisions and actions were taken to address the data quality problems reported during the verify data quality task of the data understanding phase.	X		X		X	X	
			Data Preparation - Merged Data: Merging tables refers to joining together two or more tables that have different information about the same objects.	X				X		

			Data Preparation - Reformatted Data: Some tools have requirements on the order of the attributes, such as the first field being a unique identifier for each record or the last field being the outcome field the model is to predict.	X				X		
			Data Preparation - Dataset: This is the dataset (or datasets) produced by the data preparation phase, which will be used for modeling or the major analysis work of the project.	X		X		X	X	
			Data Preparation - Dataset Description: Describe the dataset (or datasets) that will be used for the modeling or the major analysis work of the project.	X				X		
			Modeling - Modeling Technique: Document the actual modeling technique that is to be used	X		X		X	X	X
			Modeling - Models: These are the actual models produced by the modeling tool, not a report.	X		X		X	X	
			Modeling - Model Assessment: Summarize results of this task, list qualities of generated models (e.g., in terms of accuracy) and rank their quality in relation to each other.	X		X		X	X	X
			Evaluation - Assessment of Data Mining Results: Summarize assessment results in terms of business success criteria including a final statement whether the project already meets the initial business objectives.	X				X	X	
DC.7 Scrum-XP Artifacts	DTS.2 Scrum-XP (Schwaber & Sutherland, 2020) (Dudziak, 1999).	• Product vision • Product backlog • Iteration-sprint plan • Iteration-sprint Kanban board • Iteration-sprint burndown chart	**Exploration - Product vision:** Describes the overarching long-term mission of your product.	X	X	X		X	X	X
			Exploration - Product backlog: A prioritized list of work for the development team that is derived from the product roadmap and its requirements.		X		X	X	X	X
			Product Planning - Product backlog plan: No reported.		X		X	X		

		- Customer functional tests - Simple architecture design - Unit tests - Unit codes - Build increment - Iteration-sprint agreements - Product done	**Interaction Sprint Planning - Iteration-sprint plan: I**nvolves a planning meeting at the beginning of each sprint where the team analyzes the backlog items and divides them into tasks and tests.		X		X	X	X	X
			Iteration Sprint - Kanban board: Agile project management tool designed to help visualize work, limit work in progress and maximize efficiency.		X		X	X	X	
			Iteration Sprint - Burndown chart: Is a graphical representation of the work remaining for a project and the time remaining to complete it.		X		X	X		
			Iteration Sprint - Customer functional tests: Is a type of software testing that validates web or mobile applications against pre-determined specifications and requirements.		X			X		
			Iteration Sprint - Simple architecture design: It is the process of simply defining the structure, organization and planning of the hardware and software components of a computer system.		X			X		
			Iteration Sprint - Unit tests: It is an effective way to check the correct functioning of the smallest individual units of computer programs.		X			X	X	
			Iteration Sprint - Unit codes: No reported.		X			X		
			Iteration Sprint - Build increment: A product increment is whatever you previously built, plus anything new you just finished in the latest sprint, all integrated, tested, and ready to be delivered or deployed.		X		X	X	X	X
			Iteration Sprint - Iteration-sprint agreements: No reported.		X			X		
			Product Release- Product done: The final release with the final increment.	X	X	X		X	X	X

DC.11 TDSP Artifacts	DTS.3 TDSP (Microsoft, 2107).	• Charter Document • Data Sources • Data Dictionaries • Data Quality Report • Solution Architecture • Checkpoint Decision • A status Dashboard • A final modeling report • A final solution architecture document • Exit report	**Business Understanding - Charter Document:** You update the document throughout the project as you make new discoveries and as business requirements change.	X		X		X	X	
			Business Understanding - Data Sources: You can use Azure Machine Learning to handle data source management.	X		X		X	X	X
			Business Understanding - Data Dictionaries: This document provides descriptions of the data that the client provides.	X		X		X		
			Data Acquisition and Understanding - Data Quality Report: That includes data summaries, the relationships between each attribute and target, the variable ranking, and more.	X		X		X	X	
			Data Acquisition and Understanding - Solution Architecture: Such as a diagram or description of your data pipeline that your team uses to run predictions on new data.	X		X		X	X	X
			Data Acquisition and Understanding - Checkpoint Decision: Before you begin full-feature engineering and model building, you can reevaluate the project to determine whether the value expected is sufficient to continue pursuing it.	X		X		X		
			Deployment - A status Dashboard: That displays the system health and key metrics.		X	X	X	X	X	
			Deployment - A final modeling report: With deployment details.	X		X		X	X	
			Deployment - A final solution architecture document: No reported.	X		X		X		
			Customer acceptance - Exit report: This technical report contains details about the project that the customer can use to learn how to operate the system.	X	X	X		X	X	X
DC.15 DDS Artifacts	DTS.4 DDS (Saltz, 2022).	• Item • Backlog	**Brainstorm - Item:** An Item may take a variety of forms such as "user stories", "experiments", or "testable hypotheses".		X		X	X	X	

		- Item Breakdown Board - Task Board	**Prioritize - Backlog:** The Backlog is a prioritized list of Items (i.e., work to be prioritized).		X		X	X	X	X
			Observe & analyze - Item Breakdown Board: The Item Breakdown Board (IBB) is the place where each Item (in the Backlog) is broken down into tasks.		X		X	X		
			Create / Refine - Task Board: The Task Board is a visual representation of the Item(s) currently in progress.		X		X	X	X	

10.4 INSTRUCTIONS FOR THE PANEL OF EXPERTS

INSTRUCTIONS FOR THE PANEL OF EXPERTS

“Small Business DSD (Data Sprint Development).v1 – an Agile Development Methodology for Big Data Analytics Systems”

You have been kindly contacted as a potential academic expert or professional expert on Data Science Systems to evaluate the Conceptual Validity of Small Business DSD (Data Sprint Development).v1 – an Agile Development Methodology for Big Data Analytics Systems.

For this aim, please use the next documents:

- Document.1: Description of the DSD.v1 (PDF format)
- Document.2: Conceptual Validity Questionnaire (Word format)
- Document.3: Demographic Data Questionnaire (Word format)

We ask you kindly to perform the following evaluative tasks:

- Analyze Document.1 (min-max period of 15-20 minutes)
- Answer statements from Document.2 (about 15 minutes)
- Answer statements from Document.3 (about 15 minutes)

Please return the two questionnaires to gss.kw.13@gmail.com on or before October 30, 2024.

We thank you very much in advance for your academic-professional collaboration.
Sincerely,

Main Design Science Research Team
PhD(c) Gerardo Salazar Salazar, Autonomous University of Aguascalientes, Mexico
Dr. Manuel Mora, Autonomous University of Aguascalientes, Mexico
Dr. Hector Alejandro Duran Limon, University of Guadalajara, Mexico

10.5 DEMOGRAPHIC DATA OF THE PANEL OF EXPERTS

DEMOGRAPHIC DATA OF THE PANEL OF EXPERTS

(15 minutes)

“Small Business DSD (Data Sprint Development).v1 – an Agile Development Methodology for Big Data Analytics Systems”

INSTRUCTIONS. Please answer the following statements regarding your demographic data:

1. Age range: ( ) <=30 years ( ) 31-40 years ( ) 41-50 years ( ) > 50 years	2. Academic highest gained level: ( ) Bachelor level ( ) Bachelor enhanced with Professional Certifications ( ) Master level ( )Doctorate level	3. Main area of formal studies: ( ) Computer Engineering ( ) Business Informatics ( ) Business Management ( ) Other
4. Main work setting: ( ) Business enterprise ( ) University/Research Unit ( ) Government Unit	5. Scope of work setting: ( ) Regional ( ) Nationwide ( ) Worldwide	6. Region of working setting: ( ) USA/CAN ( ) Europe ( ) Asia ( ) Latin America
7. Years in work settings: ( ) 1-5 years ( ) 6-10 years ( ) 11-15 years ( ) 16-20 years ( ) 20 or more years	8. Main Work Position: ( )Academic/Researcher ( ) IT Project Manager / IT Consultant ( ) Business Manager / Business Consultant ( ) IT Senior Developer	

9A. Years involved (i.e. knowing, using, teaching, investigating or giving consulting) on AGILE PROCESS (Scrum, XP): ( ) <1 year ( ) 1-3 years ( ) 4-6 years ( ) 7-9 years ( ) 10 or more years	9B. Years involved (i.e. knowing, using, teaching, investigating or giving consulting) on Data Science Analytics Systems: ( ) <=5 years ( ) 6-10 years ( ) 11-15 years ( ) 16-20 years ( ) >20 years
10A. Number of projects (academic, training or consulting ones) involved with on AGILE PROCESS (Scrum, XP) ( ) 1-3 ( ) 4-6 ( ) 7-9 ( ) 10 or more	10B. Number of projects (academic, training or consulting ones) involved on Data Science Analytics Systems: ( ) 1-3 ( ) 4-6 ( ) 7-9 ( ) 10 or more
11A. Self-evaluation on the expertise level AGILE PROCESS (Scrum, XP) ( ) very high level of expertise ( ) high level of expertise ( ) moderate level of expertise ( ) low level of expertise ( ) very low level of expertise	11B. Self-evaluation on the expertise level on Data Science Analytics Systems: ( ) very high level of expertise ( ) high level of expertise ( ) moderate level of expertise ( ) low level of expertise ( ) very low level of expertise

Thanks very much for your valuable participation!

Main Design Science Research Team

PhD(c) Gerardo Salazar Salazar, Autonomous University of Aguascalientes, Mexico
Dr. Manuel Mora, Autonomous University of Aguascalientes, Mexico
Dr. Hector Alejandro Duran Limon, University of Guadalajara, Mexico

10.6 CONCEPTUAL EVALUATION BY PANEL OF EXPERTS

CONCEPTUAL EVALUATION BY PANEL OF EXPERTS

(15 minutes)

“Small Business DSD (Data Sprint Development).v1 – an Agile Development Methodology for Big Data Analytics Systems”

INSTRUCTIONS. Please respond the following statements regarding the conceptual validity of the Small Business DSD (Data Sprint Development).v1 – an Agile Development Methodology for Big Data Analytics Systems. You must respond to each one of the following 7 statements marking the score (1..5) that you consider as valid. Please answer all 7 statements. No answered statement will be counted as neutral (score 3).

V1.	The conceptual product (DSD.v1) is supported by robust theoretical knowledge (e.g. based on scientific literature).						
	Strongly disagree	1	2	3	4	5	Strongly agree
V2.	The theoretical knowledge used for elaborating this conceptual product (DSDv1) is relevant for the addressed topic.						
	Strongly disagree	1	2	3	4	5	Strongly agree
V3.	The scientific literature considered for elaborating this conceptual product (DSD.v1) does not present important omissions for the topic.						
	Strongly disagree	1	2	3	4	5	Strongly agree
V4.	The conceptual product (DSD.v1) is logically coherent.						
	Strongly disagree	1	2	3	4	5	Strongly agree
V5.	The conceptual product (DSD.v1) is adequate for achieving the purpose of its utilization.						
	Strongly disagree	1	2	3	4	5	Strongly agree
V6.	The conceptual product (DSD.v1) provides new scientific-based knowledge (e.g. it is not a just a duplication of an existent conceptual product).						
	Strongly disagree	1	2	3	4	5	Strongly agree
V7.	The presentation style of the conceptual product (DSD.v1) is adequate for a scientific report.						
	Strongly disagree	1	2	3	4	5	Strongly agree

Open Comments

Please feel free to add comments (if any) to improve the conceptual product DSD.v1

Thanks very much for your valuable participation as an academic or professional expert !

Main Design Science Research Team

PhD(c) Gerardo Salazar Salazar, Autonomous University of Aguascalientes, Mexico
Dr. Manuel Mora, Autonomous University of Aguascalientes, Mexico
Dr. Hector Alejandro Duran Limon, University of Guadalajara,

10.7 EVALUATION BY PANEL OF EXPERTS

PILOT EVALUATION

(30 minutes)

“Agile Data Science Analytics Development Methodology (AgileDSA-DevMet).v1 – an Agile Development Methodology for Big Data Analytics Systems (BDAS)”

INSTRUCTIONS. Please respond the following statements regarding the 7 usability metrics for the AgileDSA-DevMet.v1 – an Agile Development Methodology for Big Data Analytics Systems (BDAS)”. You must respond all items marking the score (1..5) that you consider as valid. Please answer all items. No answered statement will be counted as neutral (score 3).

USEFULNESS – *is the degree to which using the new TOOL is perceived as being better than using the current used TOOL.*	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE
	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1					RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.				
1. If I were to use the TOOL (X|Y), it would enable me to accomplish the agile development of a BDAS more quickly.	1	2	3	4	5	1	2	3	4	5
2. If I were to use the TOOL (X|Y), the quality of my work (agile development of a BDAS) would improve.	1	2	3	4	5	1	2	3	4	5
3. If I were to use the TOOL (X|Y), it would enhance my effectiveness on the job (related with the agile development of a BDAS).	1	2	3	4	5	1	2	3	4	5
4. If I were to use the TOOL (X|Y), it would make my job easier (related with the agile development of a BDAS).	1	2	3	4	5	1	2	3	4	5

EASE OF USE - *is the degree to which using the new TOOL is perceived as being free of effort.*	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE
	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1					RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.				
1. Learning to use the TOOL (X|Y), would be easy for me.	1	2	3	4	5	1	2	3	4	5
2. If I were to use the TOOL (X|Y), it would be easy to operate.	1	2	3	4	5	1	2	3	4	5
3. If I were to use the TOOL (X|Y), it would be difficult to use.	1	2	3	4	5	1	2	3	4	5

COMPATIBILITY - *is the degree to which using new the TOOL is perceived as compatible with what people do.*	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE	STRONGLY DISAGREE	DISAGREE	NETURAL	AGREE	STRONGLY AGREE
	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1					RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.				
1. If I were to use the TOOL (X|Y), it would be compatible with most aspects of my work (related with the agile development of a BDAS).	1	2	3	4	5	1	2	3	4	5
2. If I were to use the TOOL (X|Y), it would fit my work style (related with the agile development of a BDAS).	1	2	3	4	5	1	2	3	4	5
3. If I were to use the TOOL (X|Y), it would fit well with the way I like to work (related with the agile development of a BDAS).	1	2	3	4	5	1	2	3	4	5

VALUE - *the degree to which using the new TOOL is perceived as a value delivery entity for users by savings on money, time, and the provision of a variety of valuable resources, and by an overall value.*	VERY LOW	LOW	MODERATE	HIGH	VERY HIGH	VERY LOW	LOW	MODERATE	HIGH	VERY HIGH
	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1					RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.				
1. The value for saving money by using the TOOL (X|Y), for the agile development of a BDAS is:	1	2	3	4	5	1	2	3	4	5
2. The value for saving valuable time by using the TOOL (X|Y), for the agile development of a BDAS is:	1	2	3	4	5	1	2	3	4	5
3. The value for finding the information on roles-actions, phases-activities and artifacts-templates for the agile development of a BDAS by using the TOOL (X|Y) is:	1	2	3	4	5	1	2	3	4	5
4. In overall, the value of using the TOOL (X|Y), for the agile development of a BDAS is:	1	2	3	4	5	1	2	3	4	5

NOTE: please answer the 3 following questions. They have the same inquiry but their scales are different:

ATTITUDE.01	EXTREMELY NEGATIVE						EXTREMELY POSITIVE	EXTREMELY NEGATIVE						EXTREMELY POSITIVE
All considered things, using TOOL (X Y) in my job within next six months would be:	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1							RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.						
	-3	-2	-1	0	+1	+2	+3	-3	-2	-1	0	+1	+2	+3

ATTITUDE.02	EXTREMELY BAD						EXTREMELY GOOD	EXTREMELY BAD						EXTREMELY GOOD
All considered things, using TOOL (X|Y) in my job within next six months would be:	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1							RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.						
	-3	-2	-1	0	+1	+2	+3	-3	-2	-1	0	+1	+2	+3

ATTITUDE.03	EXTREMELY HARMFUL						EXTREMELY BENEFICIAL	EXTREMELY HARMFUL						EXTREMELY BENEFICIAL
All considered things, using TOOL (X|Y) in my job within next six months would be:	RESPONSES FOR TOOL X = AgileDSA-DevMet.v1							RESPONSES FOR TOOL Y = Any other BDAS Methodology you use.						
	-3	-2	-1	0	+1	+2	+3	-3	-2	-1	0	+1	+2	+3

OPEN COMMENTS:

Please feel free to add any open comment on benefits of using the AgileDSA-DevMet.v1 vs your current tool (methodology) for the agile development of a BDAS:

Benefits from using AgileDSA-DevMet.v1:

Benefits from using my current TOOL (methodology):

Please feel free to add any open comment on limitations of using the AgileDSA-DevMet.v1 vs your current tool for the agile development of a BDAS:

Limitations from using AgileDSA-DevMet.v1:

Limitations from using my current TOOL (methodology):

Thanks very much for your valuable participation!