CENTRO DE CIENCIAS BÁSICAS

MAESTRÍA EN INFORMÁTICA Y TECNOLOGIAS COMPUTACIONALES

"Factores que Determinan la Aceptación o Rechazo del Uso de Metodologías de Desarrollo de Software en la Dirección de Desarrollo de Sistemas de Información en el INEGI"

PRESENTA

L.I. Héctor Uriel Ruiz Guerrero

DIRECTOR DE TESIS

M.I.T.C. Carlos Argelio Arévalo Mercado

SINODALES

Dra. Laura A. Garza González

M.I.T.C. José Eduardo Macías Luévano

Aguascalientes, Ags., **Junio** 2010

Agradecimientos

En primer lugar quiero agradecer a Dios por permitirme llegar hasta donde he llegado. Quiero agradecer a mi papá que se encuentra en algún lugar cerca de Dios y que siempre me ha ayudado y acompañado a donde quiera que vaya.

A mi asesor el Dr. Carlos Arévalo por su acertada dirección en mi proyecto de tesis, ya que él fue el principal impulsor de que yo desarrollara este tema que en lo personal me parece bastante interesante y que sin su ayuda no hubiera podido llevar a buen término este proyecto; así mismo gracias a mis sinodales la Dra. Laura A. Garza y el M. en C. Jorge Macías por contribuir a que esta investigación tuviera la calidad necesaria para obtener el grado de maestría.

Agradezco profunda y enormemente a mi esposa Yolanda quien siempre y en todo momento me tuvo la paciencia, dedicación, sacrificio, consejos y de quien recibí un apoyo definitivo en todos los sentidos para terminar esta tesis.

A toda mi familia, mi madre quién siempre me apoya en todo lo que hago y quien fue la persona que me inculco valores y principios morales que me han llevado hasta donde estoy; a mis hermanos Adrian, Rosy, Lourdes y César por brindarme su respaldo en todo momento.

A Magda con quién recientemente he hecho una amistad muy especial y de quien recibí apoyo, una enorme confianza y consejos para no cejar en concluir este trabajo en momentos difíciles.

A mis amigos, que siempre me dieron palabras de aliento, me brindaron su ayuda y por todos los momentos vividos durante el transcurso de la maestría y que siempre los consideraré mi equipo, a Rubén por su ayuda para finalizar esta tesis, a Sergio quien siempre nos aconsejo para mejorar en muchos aspectos, a Israel por la entrega y compañerismo y a Sticker quien nos demostró serenidad, experiencia y profesionalismo.

Gracias a todos!

Dedicatorias

Esta tesis la dedico a la personita más importante para mí en este momento, a Deni, mi hija, a quien le robe muchas horas de dedicación y cuidado, de quien me prive de tener momentos de su niñez que nunca regresarán; espero que no sea demasiado tarde para recuperar todo este tiempo perdido y que el poco tiempo que le dediqué durante el transcurso de la maestría haya sido suficiente para inculcarle valores y principios que forjarán su carácter e ideología a lo largo de su vida.

Te quiero mucho Deni!

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES
Conmemoración del Bicentenario del inicio de la Independencia de México y del Centenario de la Revolución Mexicana

Centro de Ciencias Básicas

L.I. HÉCTOR URIEL RUIZ GUERRERO
PASANTE DE LA MAESTRÍA EN INFORMÁTICA
Y TECNOLOGÍAS COMPUTACIONALES
P R E S E N T E .

Estimado (a) Alumno (a) Ruiz:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o trabajo práctico titulado: **"Factores que Determinan la Aceptación o Rechazo del Uso de Metodologías de Desarrollo de Software en la Dirección de Desarrollo de Sistemas de Información en el INEGI",** hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

A T E N T A M E N T E
Aguascalientes, Ags., 9 de junio de 2010
"LUMEN PROFERRE"
EL DECANO

DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES
CENTRO DE CIENCIAS BÁSICAS

c.c.p.- Archivo

Por este conducto autorizamos al tesista:

L.I. Héctor Uriel Ruiz Guerrero

La impresión de su documento final de Tesis, ya que cumple con los requisitos de contenido y forma exigidos en la Universidad Autónoma de Aguascalientes.

Director

M.I.T.C Carlos Argelio Arévalo Mercado

Sinodal

Dra. Laura A. Garza González

Sinodal

M.I.T.C. Jorge Eduardo Macías Luévano

Abstract

Muchas organizaciones intentan implementar metodologías destinadas a mejorar los procesos de desarrollo de software. Sin embargo la resistencia de los desarrolladores hacia dichas metodologías a menudo desalienta su adopción. A pesar de que las inversiones en desarrollo de software que hacen las empresas se ha incrementado enormemente, evidencias sugieren que el desarrollo de software no está mejorando como debiera, la calidad se está mejorando a un ritmo menor de lo que se esperaba, la productividad de hecho se ha declinado en un 13 por ciento en años recientes, y hasta un 75 por ciento de todos los esfuerzos de desarrollo son considerados como fracasos. Mejorar los procesos de desarrollo de software ha sido y sigue siendo una de las principales preocupaciones de la administración de las Tecnologías de Información.

El presente estudio tiene por objetivo tratar de identificar y describir los factores que influyen en los desarrolladores de sistemas de software en la Dirección de Desarrollo de Sistemas de Información en el INEGI, para aceptar o rechazar el uso de una metodología de desarrollo de software. Este estudio pretende generar datos e hipótesis que constituyan la materia prima para investigaciones más precisas. Aplicando el método de análisis de factores se lograron obtener seis factores que determinan la aceptación de las metodologías de desarrollo de software por parte de los desarrolladores. Estos factores coinciden con estudios en los cuales está basada esta tesis, de los cuales los principales se pueden resumir en utilidad, compatibilidad y facilidad de uso.

Contenido

Título

Factores que Determinan la Aceptación o Rechazo del Uso de Metodologías de Desarrollo de Software en la Dirección de Desarrollo de Sistemas de Información en el INEGI.

Introducción

En una economía digital impulsada por las tecnologías de la información, se ha demostrado que los sistemas computacionales contribuyen enormemente a acelerar la productividad de cualquier negocio o empresa. Anualmente se invierten importantes cantidades de dinero en el desarrollo de sistemas de software, mismos que según un estudio publicado por Standish Group, empresa dedicada a la asesoría en el desarrollo de sistemas de información, más de la mitad de estos desarrollos fracasarán. (Standish Group, 1994) Una manera de enfrentar este problema es establecer metodologías de desarrollo de software proporcionadas por la ingeniería de software. Sin embargo, en ocasiones las empresas que intentan implementar alguna de estas metodologías se enfrentan a un rechazo por parte de los desarrolladores de software, lo cual es un obstáculo para alcanzar el éxito al establecer dichas metodologías. (Riemenschneider, B.C. Hardgrave, & F.D. Davis, 2002)

Existen modelos teóricos basados en variables psicológicas para explicar los factores que influyen en el rechazo de las personas a utilizar tecnologías de información, pero están orientados a explicar el uso y aceptación de tecnologías, más no la adopción de metodologías de desarrollo de software, que es un fenómeno más específico, que se relaciona con usuarios más especializados. Por lo que los estudios que intentan explicar este fenómeno generalmente combinan ciertas variables de los modelos teóricos de adopción de tecnologías.

1.1 Importancia del uso de metodologías.

El software afecta la vida cotidiana, nuestros sistemas financieros están administrados por sistemas bancarios, nuestro transporte esta manejado por varios dispositivos que

contienen software embebido, el cuidado de nuestra salud depende de sofisticados sistemas médicos, nuestra manufactura se lleva con enormes sistemas de procesamientos de recursos empresariales e incluso nuestro suministro de comida es administrado por sistemas de software de distribución y almacenaje. (Tsui & Karam, 2006)

Con nuestro mundo dependiendo fuertemente del software, es natural que la ingeniería de software, la cual incluye el desarrollo y soporte de esos sistemas de software, esté ganando la atención de la industria del software, las academias y el gobierno. (Tsui & Karam, 2006)

La ingeniería de software es una disciplina de la ingeniería cuya meta es el desarrollo costeable de sistemas de software. Éste es abstracto e intangible. No está restringido por materiales, o gobernado por leyes físicas o por procesos de manufactura. De alguna forma esto simplifica la ingeniería del software ya que no existen limitaciones físicas del potencial del software. Sin embargo, esta falta de restricciones naturales significa que el software puede llegar a ser extremadamente complejo y, por lo tanto, muy difícil de entender. La ingeniería de software comprende todos los aspectos de la producción de software. (Sommerville, 2005)

A pesar de los adelantos en las técnicas de desarrollo de software durante los últimos treinta años, tanto la calidad del software como la productividad de su proceso de elaboración todavía no han alcanzado niveles comparables con los de otras tecnologías más antiguas. Todo esto, combinado con un aumento espectacular de la demanda de software, ha provocado lo que se ha denominado *la crisis del software*. (Falgueras, 2003)

Las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. (Abud, 2005)

El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, por lo que suelen establecerse métricas que permiten evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar. Existen estándares para evaluar la calidad del software, tomando uno como

ejemplo: ISO/IEC 9126 de la familia ISO. Esta norma ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology – Software product evaluation: Quality characteristics and guidelines for their use” en el cual se establecen las características de calidad para productos de software. Este estándar, establece que cualquier componente de la calidad de software puede ser descrito en términos de una a seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. Las siguientes son preguntas centrales que atiende cada una de estas características:

Funcionalidad: ¿Las funciones y propiedades satisfacen las necesidades explicitas e implícitas; esto es, el qué …?

Confiablidad: ¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?

Usabilidad: ¿El software es fácil de usar y de aprender?

Eficiencia: ¿Es rápido y minimalista en cuanto al uso de recursos?

Mantenibilidad: ¿Es fácil de modificar y verificar?

Portabilidad: ¿Es fácil de transferir de un ambiente a otro?

La causa principal de dificultades en cuanto a la calidad, es la gran complejidad del software comparado con otros tipos de productos, que provoca, por ejemplo, que no sea posible, ni mucho menos, probar el funcionamiento de un software en todas las combinaciones de condiciones que se pueden dar. Y eso ocurre en una época en la que se da cada vez más importancia a la calidad en todos los ámbitos, al considerarla un factor de competitividad dentro de unos mercados cada vez más saturados y, por lo tanto, más exigentes. No es extraño, pues, que el tema de la calidad adquiera una importancia creciente dentro de la ingeniería de software. (Falgueras, 2003)

Por lo que respecta a la productividad, cabe decir para empezar que cualquier fabricación en serie tiene necesariamente una productividad mucho más elevada que la fabricación

de un producto singular; pero, incluso, si la comparamos con otras ingenierías la productividad es claramente inferior. La complejidad del producto también puede ser una causa de este hecho, pero ciertamente no es la única. Un factor que tiene un peso importante en la baja productividad es el hecho de que, a diferencia de las otras tecnologías, en un proyecto de software el desarrollo empieza tradicionalmente de cero (solo recientemente se utilizan fragmentos de software "prefabricados"). (Falgueras, 2003)

Una cuestión importante que enfrentan los administradores de sistemas de información es mejorar la efectividad del desarrollo de sistemas. Muchas organizaciones están intentando implementar metodologías de desarrollo de sistemas para apoyar en este esfuerzo. (Roberts, Gibson, & Fields, 1998)

1.2 Metodologías de desarrollo de software.

Las metodologías de administración de proyectos de software que se han desarrollado en las pasadas dos décadas se han hecho para enfrentar el endémico problema de los fracasos de los proyectos de software causados, en gran medida, por la falta de una planeación y pobre ejecución. (Brewer, Dittman, & Ghatge, 2005)

Las metodologías de desarrollo de sistemas pueden mejorar el proceso de desarrollo una vez que el entendimiento de los principios fundamentales se alcanza. En ambientes modernos de sistemas de información, una metodología de desarrollo de sistemas comprende una estrategia global para el desarrollo sistemas de información basados en computadoras que incluya un marco flexible de la secuencia de las tareas de desarrollo, junto con las técnicas utilizadas para realizar cada tarea. Esta definición no es la única que describe una metodología de desarrollo de sistemas. La literatura de sistemas de información contiene tantas definiciones de metodologías de desarrollo de sistemas como tantas metodologías hay, de las cuales existen cientos. (Roberts et al., 1998)

Una metodología, impone un proceso disciplinado sobre el desarrollo de software con el objetivo de hacer este desarrollo más predecible y eficiente. Todas las actividades basadas en conocimiento utilizan métodos que varían en sofisticación y formalidad, una metodología describe como modelar y construir un sistema de software de una forma confiable y reproducible. (Gacitúa, 2003)

El uso de metodologías estándar de alguna manera ha ayudado en asegurar que los proyectos emprendidos tengan algún grado de éxito. (Brewer et al., 2005)

Las metodologías de desarrollo de sistemas proveen el potencial para ayudar a aliviar los problemas de desarrollo y mantenimiento de software que plagan muchos departamentos de sistemas de información. Las primeras metodologías de desarrollo de sistemas se desarrollaron para intentar responder a la complejidad de los problemas asociados con el desarrollo de sistemas. Un estudio muestra la necesidad de desarrollar métodos rigurosos para desarrollar requerimientos de sistemas para incrementar la confiabilidad y reducir los costos de los sistemas para las organizaciones. Otros estudios muestran que errores tardíos en el proceso de desarrollo son más costosos que los errores corregidos de manera temprana en el proceso. Las metodologías de desarrollo de sistemas ofrecen una alternativa a los métodos de desarrollo intuitivos, informales y riesgosos, los cuales son usados frecuentemente en la práctica. El uso de metodologías de desarrollo de sistemas formales es importante por varias razones: (Roberts et al., 1998)

1. Se pueden enseñar
2. Proveen consistencia debido a que todos usan las mismas técnicas
3. Estas requieren entregables de manera explícita que se pueden corroborar para la calidad
4. Proveen una disciplina de desarrollo tipo ingeniería

La metodología de desarrollo cobra gran importancia en proyectos empresariales pues al no utilizarla adecuadamente se puede desembocar en la frustración del equipo de desarrollo y en la insatisfacción de los clientes. Por tanto el uso de una metodología es crítico para controlar el ciclo de vida de un proyecto. (Torres & Alférez, 2008)

Los modelos de desarrollo de sistemas como cascada, Sashimi, Espiral y metodologías ágiles se han convertido en herramientas clave para los administradores de proyectos y desarrolladores de software para apoyarlos en la entrega de proyectos a tiempo, dentro del presupuesto y que cumplan con los requerimientos del cliente. Desafortunadamente, muchos no aplicaron estos modelos adecuadamente ni utilizaron la metodología correcta que encajara en el proyecto. (Brewer et al., 2005)

Los desarrolladores no siempre creen que el enfoque de desarrollo elegido es necesariamente el enfoque apropiado. En sus investigaciones (Verner & Cerpa, 1997) encontraron que algunos profesionales se habían involucrado en desarrollos en el que deberían haber usado el enfoque de "prototipos", pero en cambio se uso el método de "cascada"; otros desarrolladores se involucraron en desarrollos donde se utilizó "prototipos", a pesar de que los desarrolladores pensaron que el modelo de "cascada" hubiera sido más apropiado. (Khalifa & Verner, 2000)

El Standish Group, una compañía de consultoría, la cual publica anualmente sus resultados desde 1994 en un reporte llamado "El Caos", es líder en evaluación de riesgos, retorno de inversión y costo en inversiones de tecnologías de información. (Brewer et al., 2005)

Según el reporte de "El Caos" en 1994, en los Estados Unidos se gastan $250 billones de dólares cada año en el desarrollo de aplicaciones de TI, de aproximadamente 175,000 proyectos. Una gran parte de estos proyectos, fracasará. El Standish Group muestra que un 31.1% de los proyectos serán cancelados antes de que se completen, 52.7% de los proyectos costarán 189% más de su costo estimado o se terminarán después de la fecha de termino pactada. El costo de estos fracasos y excesos de tiempo y presupuesto son solo la punta de un iceberg. El costo de las oportunidades perdidas no es medible, pero puede ser fácil en trillones de dólares. Uno solo tiene que ver la ciudad de Denver para darse cuenta del alcance del problema. El fracaso para producir software confiable para manejar el equipaje en el nuevo aeropuerto le está costando a la ciudad $1.1 millones de dólares por día. (Standish Group, 1994)

Por el lado de los éxitos, el promedio de los proyectos que se completan a tiempo y en presupuesto es del 16.2% y en las compañías grandes es de solo el 9%. Aún cuando estos proyectos son finalizados muchos no son ni la sombra de las especificaciones de los requerimientos originales. Estos proyectos completos por las compañías más grandes americanas, tienen solo aproximadamente el 42% de las características y funciones propuestas originalmente. Las compañías pequeñas lo hacen mucho mejor con un 74.2% de sus funciones y características originales. (Standish Group, 1994). En la figura 1.2.a siguiente podemos observar mejor la situación antes expuesta.

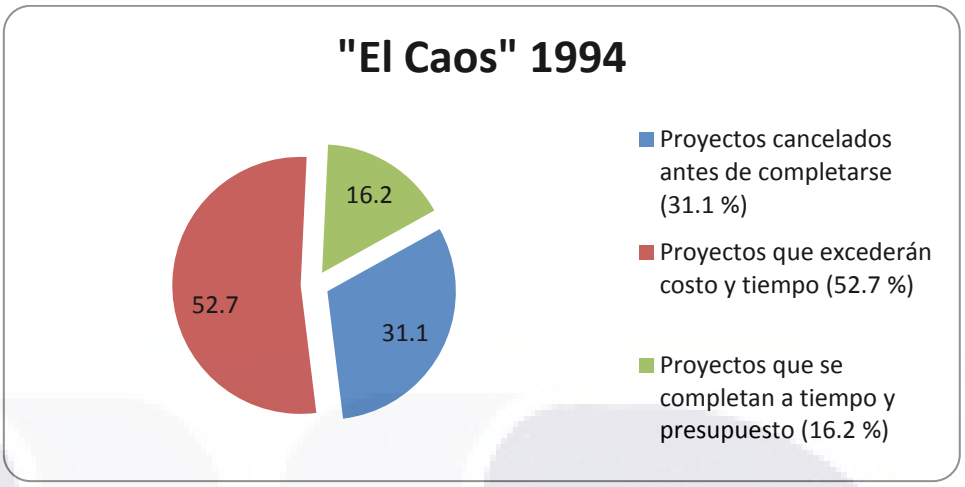

Figura 1.2.a Porcentajes de éxito en los proyectos de software (1994)

Cerca de tres cuartas partes de todos los proyectos de TI en la era de internet que fueron concebidos en los últimos siete años han sufrido de uno o más de los siguientes: fracaso total, exceso en costo, exceso de tiempo o un despliegue con menos características o funciones de las que se prometieron. (Brewer et al., 2005) coraje frustración

Ahora en 2009, Standish Group presenta que, como se muestra en la figura 1.2.b los resultados de este año muestran un marcado decremento en los porcentajes de proyectos exitosos con respecto a años anteriores, con un 32% de todos los proyectos que tienen éxito con entrega a tiempo, presupuesto estimado y las características y funciones requeridas. Un 44% de los proyectos se terminaron con retraso en tiempo, mayor presupuesto y con menos de las funciones y características requeridas y un 24% de los proyectos fracasaron, ya sea porque se cancelaron antes de terminarse o proyectos que se entregaron pero nunca se usaron. (Standish Group, 2009)

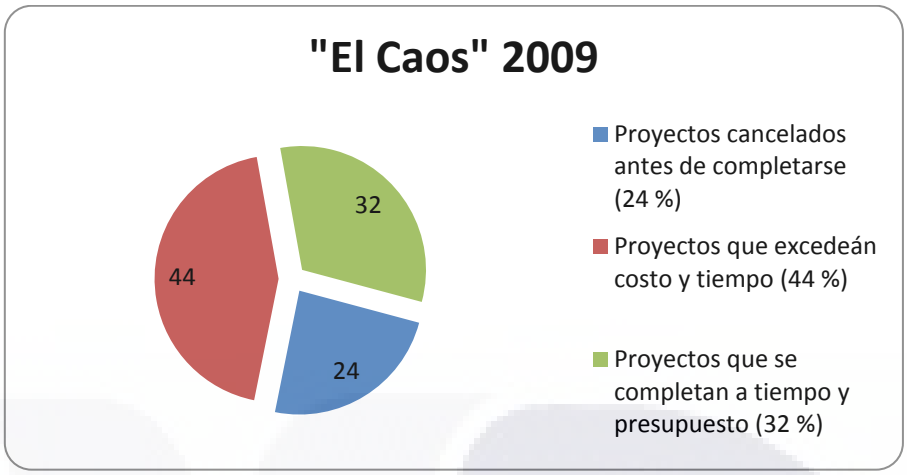

Figura 1.2.b Porcentajes de éxito en los proyectos de software (2009)

Estos números representan un decremento en las tasas de éxitos con respecto a previos estudios, de la misma manera que hay un significante incremento en el número de fracasos, según Jim Crear, director de Informática en Standish Group, estos puntos están bajos con respecto a los últimos cinco periodos de estudio. Los resultados de este año representan la tasa de fracasos más alta en esta década. (Standish Group, 2009)

Por otro lado el ciclo de vida de un proyecto no es la solución a todos los problemas concernientes al desarrollo de un proyecto. Esto significa que no liberará al administrador del proyecto de la difícil responsabilidad de tomar decisiones, comparar alternativas, pelear batallas políticas, negociar con los usuarios, levantar la moral de los programadores o cualquier otra prueba relacionada al proyecto. El administrador del proyecto, permanecerá administrando en todo el sentido de la palabra. La única ayuda que el ciclo de vida puede proveer, es que puede organizar las actividades del administrador, haciendo más probable que los problemas, serán enfrentados en el momento preciso. (Yourdon, 2006)

1.3 Resistencia al uso de metodologías

Muchas organizaciones tratan de implantar metodologías dirigidas a mejorar los procesos de desarrollo de software. Pero, la resistencia individual de los desarrolladores de

software contra el uso de tales metodologías a menudo obstruye su adopción. (Riemenschneider et al., 2002)

Existen estudios relacionados a la aceptación o rechazo de las nuevas tecnologías, que se apoyan en modelos de comportamiento tales como *TAM (Technology Acceptance Model), TAM2 (Una extensión del modelo TAM), TPB (Theory of Planned Behavior), PCI (Perceived Characteristics of Innovating), MPCU (Model of Personal Computer Utilization), DOI (Diffusion of Innovation)*; todos estos como los más importantes.

Estos modelos están enfocados a explicar las causas que determinan la aceptación o rechazo de las herramientas de tecnología de información, haciendo referencia a sistemas de software principalmente, pero no fueron diseñados para explicar la adopción de metodologías, por lo que los investigadores han tenido que adaptarlas a tal contexto.

Así (Riemenschneider et al., 2002) identificaron cuatro determinantes de intención de uso significativos para la adopción de metodologías: ***utilidad*** (en los cinco modelos), ***norma subjetiva*** (TAM2, TPB y MPCU), ***voluntariedad*** (TAM2 y PCI) y ***compatibilidad*** (PCI). Como se puede observar, ninguno de los cinco modelos originales contiene los cuatro determinantes de uso significativos. El presente documento abordará el modelo TAM2 y PCI, dado que estos modelos abarcan los cuatro determinantes significativos de intención de uso de las cinco metodologías. Aunque también se explicarán otros modelos para apoyar la documentación de estos dos últimos que se tomarán como base.

Khalifa y Verner (2000) desarrollaron y probaron un modelo que explica y por lo tanto predice, el grado de uso de los métodos de desarrollo. Encontraron dos principales constructos, “calidad del proceso” y “condiciones de facilidad” como los conductores hacia el uso de metodologías.

La adopción inicial de un método o enfoque de desarrollo de software (ej. Cascada, prototipos, espiral, iterativo u orientado a objetos, etc.) normalmente es una decisión organizacional. A menudo múltiples métodos son usados y adoptados juntos para el mismo proyecto, pero en diferentes fases del ciclo de vida de desarrollo. El grado de uso de un método particular, sin embargo es una decisión individual hecha por los desarrolladores o con una aportación significativa hecha por estos mismos.

(Patrick Y K. Chau & HU, 2001) En el contexto de la telemedicina, examinaron y compararon los modelos “Modelo de Aceptación de Tecnología (TAM)”, la “Teoría de la Conducta Planificada (TPB)”, y un modelo de descomposición TPB. Los resultados del estudio destacan las posibles limitaciones de TAM y TPB en explicar o predecir la aceptación de la tecnología por parte de profesionales. Además, las conclusiones del estudio también indican que los instrumentos que se han desarrollado y probado en varias ocasiones en anteriores estudios con usuarios finales y los directores de las empresas en condiciones normales de una empresa no pueden ser igualmente válidos en un entorno profesional.

(Pfleeger, 1999) sugirió que los investigadores deberían buscar ejemplos de transferencia de tecnología y posteriormente evaluarlos para entender las variables principales que hacen la adopción rápida y efectiva. En un estudio exploratorio, (Roberts, Gibson, & Fields, 1999) señaló los factores potenciales que pueden afectar la implementación de metodologías: transición organizacional (la preparación y la ejecución), soporte de la gerencia, transición del proceso (entendimiento del nuevo proceso), uso de modelos por el proceso y soporte externo. Este estudio no estuvo fundamentado teóricamente, empleo medidas psicométricas no validas y no evaluó la relación entre los factores derivados y las intensiones de uso de cada desarrollador o comportamiento. En el contexto del desarrollo de sistemas orientados a objetos, (Johnson, B. Hardgrave, & Doke, 1999) utilizó el procedimiento de (Ajzen, 1988) de pruebas previas para obtener creencias para identificar principales creencias específicas sobresaliendo la actitud, norma subjetiva, control del comportamiento percibido, constructos de la Teoría del Comportamiento Planificado. No probaron empíricamente la relación entre estas creencias obtenidas y los constructos de TPB, intensiones de los desarrolladores o el uso del desarrollo de sistemas orientados a objetos. (Khalifa & Verner, 2000) desarrollaron un modelo de las determinantes de uso de los desarrolladores de software de las metodologías de cascada y prototipos, basadas en parte al modelo de (Triandis, 1979) de la conducta humana. Aunque en su modelo no emplearon escalas validadas psicométricamente de constructos teóricos, este es el estudio más valido teóricamente de los estudios previos de la adopción de metodologías por los desarrolladores.

1.4 Adopción de herramientas CASE

En general, hay una escasez de investigación en la literatura de la ingeniería de software sobre las determinantes de la aceptación de metodologías. Los únicos estudios previos que directamente pertenecen a la adopción a nivel individual de metodologías, son los estudios exploratorios preliminares de (Roberts et al., 1998), (Roberts et al., 1999), (Johnson et al., 1999) y (Khalifa & Verner, 2000). Otras investigaciones relacionadas abordan la adopción de herramientas o técnicas específicas que pueden ser usadas dentro de una metodología, en lugar de la adopción de una metodología. Otros tipos de estudios que se relacionan en algún punto, tales como los enfoques descriptivos o de contingencia han abordado los patrones de uso de las metodologías y las consecuencias de desempeño de relacionar las metodologías en un contexto específico, pero no han estudiado las determinantes de la intención del desarrollador para adoptar una metodología (Riemenschneider et al., 2002). Un ejemplo de lo anterior es (Conference & Khosrowpour, 1990) el cual contiene un artículo que reporta las características de dos principales metodologías de desarrollo (Ciclo de vida del desarrollo de sistemas ("ASDM" por sus siglas en inglés) y el de prototipos). Los resultados se presentan en el uso/aplicabilidad durante varias fases del desarrollo de sistemas, diferentes tipos de sistemas y diferentes tipos de problemas. La principal contribución de este artículo es su ayuda para un enfoque de contingencia en la selección de una metodología, más no aborda el tema de los factores de aceptación de las mismas.

Se han hecho algunas investigaciones a nivel individual que no se relacionan directamente con la adopción de metodologías, sino con la adopción de herramientas específicas (como CASE) y técnicas (como programación orientada a objetos) comúnmente usadas dentro de las metodologías. Esta es una distinción clave, ya que la adopción de metodologías representa un cambio mucho más radical que adoptar herramientas y técnicas. Además, las herramientas y técnicas pueden ser y a menudo son, usadas fuera del contexto de las metodologías. (Riemenschneider et al., 2002)

Capítulo 2

Problemática particular

En el INEGI existen numerosas áreas dedicadas al desarrollo de sistemas, existen desde las áreas llamadas de tratamiento de información en donde se desarrollan sistemas de software de tamaño pequeño principalmente para autoconsumo, en estas áreas se realizan operaciones de tratamiento, verificación y corrección de información; y por otro lado se encuentran las áreas dedicadas completamente al desarrollo de sistemas de tamaños diversos, desde pequeños hasta grandes. También se puede hacer distinción del tipo de sistemas; para el desarrollo del presente documento tomaremos como referencia áreas donde se desarrollan para la captación, tratamiento y difusión de información estadística así como también sistemas de automatización de procesos de trabajo administrativo. Las áreas mencionadas pertenecen a la Dirección de Desarrollo de Sistemas de Información, cuyo personal de desarrollo es el que tomaremos como población para la presente tesis.

El tipo de herramientas que se utilizan para el desarrollo de sistemas son muy variados, como lenguajes de programación podemos mencionar Delphi y Java con entornos de desarrollo como JDeveloper y NetBeans, junto con JSF-ADF, Hibernate e ibatis como tecnologías y frameworks de Java; para consultar bases de datos se utilizan herramientas y lenguajes como PL/SQL de Oracle, AQUA, Toad; para documentación de los sistemas se utiliza software como la suite de Microsoft Office, Flow, StarUML, plugin de JDev, FrontPage para diseño de páginas Web, Subversion y SmartSDN para control de versiones, LoadRunner para testeo de aplicaciones entre muchos otros programas y aplicaciones.

El número de personas que están involucradas en la Dirección de Desarrollo de Sistemas de Información, asciende aproximadamente a los 50 desarrolladores[*1], el personal en esta área no corresponde a todo el personal de desarrollo de sistemas en todo el INEGI, es solo una área que se tiene detectada que utiliza la metodología de desarrollo propia del instituto. Entre los cuales ya se encuentran líderes de proyecto, analistas, diseñadores, desarrolladores y testers; generalmente ocurre que una persona juega diferentes roles. Esta fue una limitante importante para la aplicación del instrumento de medición, ya que era fundamental que el personal a encuestar tuviera conocimientos lo más homogéneos

posible y que conocieran la diferencia de utilizar una metodología de desarrollo y no usarla.

A pesar de que los profesionales en el área de desarrollo de sistemas saben que las probabilidades de concluir el desarrollo de un sistema de software de mejor forma, es siguiendo una metodología de desarrollo y de que el INEGI posee su propia metodología basada en Proceso Unificado (UP); evidencia indirecta muestra que en esta y en otras áreas de desarrollo de sistemas no se están aplicando metodologías o bien no se aplican de manera correcta, debido a que no existe una documentación adecuada sobre los sistemas, no se realiza la debida calendarización de un desarrollo, no se establecen bien los objetivos, alcances y limitaciones del sistema, etc.

Los efectos de esta falta de metodologías o correcta aplicación de ellas, incide en una baja calidad en los sistemas, retrasos en las fechas de entrega, requerimientos no cubiertos o sistemas no utilizados por los usuarios.

Sería deseable conocer los factores que afectan la adopción o rechazo a las metodologías por parte de los desarrolladores, ya que con esta información se puede hacer un estudio de cual o cuales metodologías pueden mejorar este problema y con ello contribuir a una actitud positiva del personal para usar la metodología, e incluso se puede elaborar una metodología a la medida, tomando en cuenta estos factores.

*1: Es muy difícil determinar el número de áreas de desarrollo y desarrolladores que hay en todo INEGI debido a que las áreas tienen diferentes nombres, mismos que en ocasiones no nos dicen el propósito de esa área y las plazas de los desarrolladores a menudo son tomadas de otros proyectos que no son de desarrollo de sistemas.

Capítulo 3

Objetivo general

Identificar los factores de la aceptación o rechazo de las metodologías de desarrollo de software por parte de los profesionales en esta área laboral en la Dirección de Desarrollo de Sistemas de Información en INEGI.

Objetivos específicos:

- Diseñar un instrumento de medición de factores de aceptación y rechazo al uso de metodologías de desarrollo de software, tomando como base modelos de comportamiento originalmente diseñados para medir la aceptación de herramientas de tecnologías de información.
- Probar el instrumento de medición para verificar su funcionalidad y fiabilidad de los resultados mediante métodos estadísticos.
- Aplicar el instrumento de medición en áreas de desarrollo de la Dirección de Desarrollo de Sistemas de Información.
- Identificar y describir los factores que determinan la aceptación o el rechazo al uso de las metodologías de desarrollo de software, mediante análisis estadístico de factores.

Capítulo 4. Marco Teórico

4.1 Modelos de Intención de la Psicología Social.

La presencia de computadoras y tecnologías de información en las organizaciones actuales se ha extendido dramáticamente. Algunas estimaciones indican, que desde los 80's, cerca del 50 por ciento de todo el nuevo capital que se invierte en organizaciones han sido en tecnologías de información. Todavía para que las tecnologías incrementen la productividad, estas deben ser aceptadas por los empleados de las organizaciones. (Venkatesh, Smith, Morris, G. Davis, & F. Davis, 2003)

Los sistemas computacionales no pueden mejorar el desempeño de una organización si no se usan. Desafortunadamente, la resistencia a usar los sistemas por parte de los administradores y profesionales, es un problema ampliamente extendido. Los desarrolladores de software requieren un mejor entendimiento del por qué las personas se resisten a usar computadoras, a fin de elaborar métodos prácticos para evaluar los sistemas, predecir cómo responderán los usuarios y mejorar la aceptación de estos (F. Davis, 1989).

Entender por qué las personas aceptan o rechazan las computadoras, ha probado ser uno de los mayores retos en la investigación de los sistemas de información. Los investigadores han estudiado el impacto de las creencias y actitudes internas de los usuarios en su comportamiento y cómo éstas son influenciadas por factores externos, incluyendo: las características de diseño técnico del sistema, involucramiento del usuario en el desarrollo del sistema, el tipo de proceso de desarrollo de sistema usado, la naturaleza del proceso de implementación y el estilo cognitivo. En general, sin embargo, estas investigaciones han sido mezcladas e inconclusas. En parte, esto es debido al amplio arreglo de diferentes creencias, actitudes y medidas de satisfacción que han sido empleadas, a menudo sin una justificación teórica o psicométrica adecuada. (F. Davis, 1989)

Investigadores en sistemas de información han sugerido modelos de intención de psicología social como un potencial fundamento teórico para investigar en las determinantes de conducta del usuario (F. Davis, 1989). Explicar la aceptación del usuario de la nueva tecnología es a menudo descrito como una de las áreas de

investigación más maduras en la literatura de los sistemas de información contemporánea. (Venkatesh et al., 2003)

La psicología social es una disciplina que estudia cómo los fenómenos psicológicos están determinados y conformados por procesos sociales y culturales. En su larga historia, son muchas las temáticas tratadas con esta finalidad: mientras que en sus inicios los temas fundacionales tenían que ver básicamente con los instintos, la imitación, la sugestión y los fenómenos colectivos, en su posterior institucionalización destacan temáticas como el análisis de la formación de la identidad social, los procesos de normalización y socialización, la formación y cambio de las actitudes, la violencia y la agresión social, y los procesos de influencia (mayoritaria y minoritaria), conformidad y obediencia. De la misma manera, y paralelamente a todo este conjunto de investigaciones y modelos teóricos, encontramos en la disciplina una permanente reflexión sobre su aplicabilidad y la posibilidad de intervenir en los problemas sociales. (Gracia, 2004)

Muchos modelos teóricos han sido introducidos en la literatura de Sistemas de Información en relación a las determinantes de las intenciones individuales voluntarias para adoptar herramientas de Tecnologías de Información en el trabajo (Riemenschneider et al., 2002). Las investigaciones en esta área han resultado en varios modelos teóricos, con raíces en sistemas de información, psicología y sociología, que rutinariamente explican más del 40 por ciento de la varianza en la intención individual para usar tecnologías. Los investigadores se enfrentan con una selección de entre una multitud de modelos y encuentran que solo deben “seleccionar y elegir” constructos entre los modelos, o elegir el modelo favorito e ignorar las contribuciones de los modelos alternativos. (Venkatesh et al., 2003)

4.2 Teoría de la Acción Razonada (TRA)

La TRA (Ajzen & Fishbein, 1980) en un modelo ampliamente estudiado de la psicología social el cual se ocupa de las determinantes del comportamiento intencionado conscientemente. De acuerdo con TRA el rendimiento de una persona de un comportamiento específico es determinado por su *intención de comportamiento* (BI) para realizar el comportamiento, y BI esta conjuntamente determinado por la *actitud* de la

persona (A) y la *norma subjetiva* (SN) concerniente al comportamiento en cuestión, con pesos relativos típicamente estimados por regresión: BI = A + SN

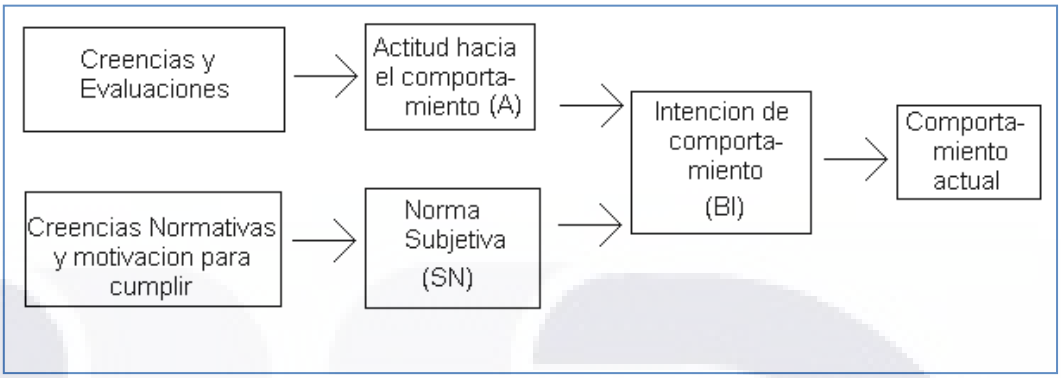

Figura 4.2.a Modelo TRA. Figura tomada de (Ajzen & Fishbein, 1980)

"BI" es una medida de la fuerza de la intención de una persona a realizar un comportamiento específico. "A" es definida como los sentimientos positivos o negativos (afecto evaluativo) sobre realizar el comportamiento objetivo. La norma subjetiva se refiere a "la percepción de una persona de que la mayoría de las personas quienes son importantes para ella piensan que deberían realizar el comportamiento en cuestión".

De acuerdo con TRA, la actitud de una persona hacia un comportamiento está determinada por sus creencias principales (b_i) sobre las consecuencias de realizar el comportamiento, multiplicado por la evaluación (e_i) de esas consecuencias:

$$A = \sum b_i e_i$$

Las creencias (b_i) son definidas como la probabilidad subjetiva individual que al realizar el comportamiento objetivo resultará en la consecuencia *i*. El termino de evaluación (e_i) se refiere a "una respuesta evaluativa implícita" a la consecuencia. Esta última ecuación postula que los estímulos externos influyen en las actitudes solo indirectamente a través de cambios en la estructura de la creencia de la persona.

TRA teoriza que una norma subjetiva de un individuo (SN) es determinada por una función de multiplicación de sus creencias normativas (nb_i), por ejemplo, las expectativas percibidas de determinados individuos o de un grupo referente, y su motivación para cumplir (mc_i) con esas expectativas:

$$SN = \sum nb_i mc_i$$

TRA es un modelo general, y como tal, no especifica las creencias que son operativas para un comportamiento en particular. Los investigadores que usan TRA deben primero identificar las creencias que son principales para los temas en relación al comportamiento bajo investigación. (Ajzen & Fishbein, 1980) sugieren obtener de cinco a nueve creencias sobresalientes usando entrevistas de respuesta libre con miembros representativos de la población objetivo.

Un aspecto particularmente útil de TRA desde una perspectiva de sistemas de información, es su afirmación de que cualquier otro factor que influye en el comportamiento lo hace solo indirectamente influenciando "A", "SN" o sus pesos relativos. Así, las variables tales como características de diseño de sistemas, características de usuario (incluyendo estilo cognitivo y otras variables de personalidad), características de las tareas, naturaleza del desarrollo o procesos de implementación, influencias políticas, estructura organizacional y así sucesivamente las que caigan en esta categoría, las cuales son referidas como "variables externas". Esto implica que TRA interviene en el impacto de variables ambientales incontrolables e intervenciones controlables en el comportamiento del usuario. De ser así entonces TRA captura las variables psicológicas internas a través de numerosas variables externas estudiadas en la investigación de sistemas de información logrando su influencia sobre la aceptación de los usuarios, y puede proporcionar un marco común de referencia dentro del cual integra diferentes líneas de investigación.

Un cuerpo sustancial de datos empíricos en soporte de TRA se ha acumulado. TRA se ha utilizado ampliamente en investigaciones aplicadas que abarcan una gran variedad de áreas, mientras que al mismo tiempo estimula una gran cantidad de investigaciones teóricas orientadas a la comprensión de las limitaciones teóricas, probando supuestos principales y analizando varias extensiones y refinamientos. (Fred, Richard, & Paul, 1989)

4.3 Modelo de Aceptación de Tecnología (TAM)

Fred Davis en sus artículos *"User Acceptance of Computer Technology: A Comparison of Two Theoretical Models (1989)"* y en "*Perceived Usefulness, Perceived Ease of Use, and*

User Acceptance of Information Technology (1989)" introdujo el *Modelo de Aceptación de Tecnología* (TAM) para predecir y explicar el conocimiento en la adopción de los trabajadores sobre la aceptación de la aplicaciones de tecnologías de información en el área laboral de una organización. (Riemenschneider et al., 2002)

El objetivo de TAM es proveer una explicación de las determinantes de la aceptación de computadoras que es general, capaz de explicar la conducta del usuario a través de un amplio rango de tecnologías computacionales para el usuario final y la población de usuarios, mientras que al mismo tiempo están siendo ponderados y teóricamente justificados. (Fred et al., 1989)

Idealmente a uno le gustaría un modelo que sea útil no solo para predecir sino también para explicar, así que los investigadores y profesionales puedan identificar porque un sistema en particular puede ser inaceptable y perseguir los pasos correctivos apropiados. Un propósito clave de TAM, por lo tanto, es proveer una base para rastrear el impacto de factores externos en creencias internas, actitudes e intensiones. (Fred et al., 1989)

TAM postula que las intenciones de usar un sistema están conjuntamente determinadas por la *utilidad* percibida por el individuo, que es la medida en que la persona piensa que usando el sistema mejorará su desempeño y la *facilidad de uso* percibida, que es la medida en la que el usuario percibe que usar el sistema estará libre de esfuerzo. A través de numerosos estudios, la *utilidad* percibida típicamente tiene una fuerte influencia en las intenciones de uso, y la *facilidad de uso* tiene un efecto secundario significativo. (Riemenschneider et al., 2002)

El *Modelo de Aceptación de Tecnología* se basa en la *Teoría de la Acción Razonada* (TRA), para relacionar dos conjuntos de constructos: *Utilidad* percibida (UP) y *Facilidad de Uso* percibida (FUP); y la actitud del usuario (A), intenciones conductuales (IC), y el comportamiento real de uso de la computadora. La UP y FUP definen la actitud hacia el sistema o sea el deseo de usarlo. A y UP influyen la IC de un usuario de usar un sistema. La predicción de usar un sistema es dado por IC. (Palacios, 2004)

La teoría ha emergido como uno de los modelos más influyentes en el ámbito de la investigación. Sin embargo, varios autores hacen notar que ésta teoría es incompleta en un aspecto muy importante, y es el hecho de que no toma en cuenta la influencia social en la adopción de un nuevo sistema de información. (Palacios, 2004)

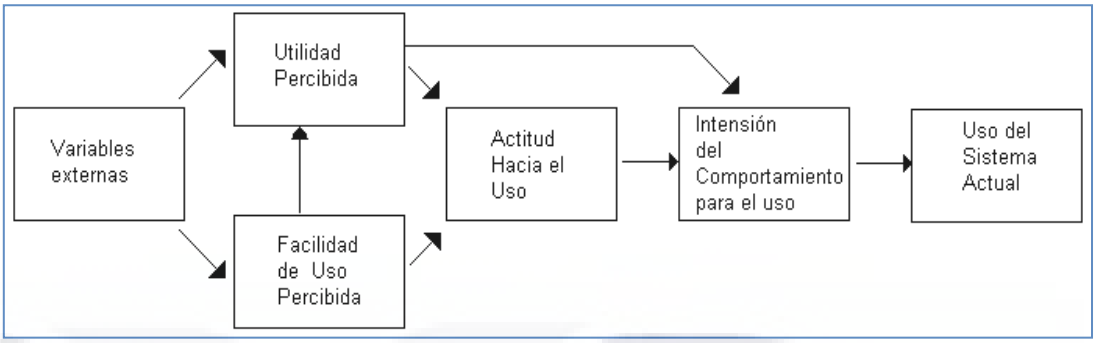

Figura 4.3.a Modelo TAM. Figura tomada de (Fred et al., 1989)

4.3.1 Utilidad y Facilidad de Uso Percibida

¿Qué es lo que ocasiona que las personas acepten o rechacen las tecnologías de información? De entre muchas variables que pueden influenciar el uso de sistemas, investigaciones previas sugieren dos determinantes que son especialmente importantes. En primer lugar, las personas tienden a usar o no usar una aplicación en la medida que ellos creen que les ayudará a realizar sus trabajos de una mejor manera. Nos referimos a esta primera variable como *utilidad percibida*. En segundo lugar, aún si los usuarios potenciales creen que una aplicación determinada es útil, ellos pueden, al mismo tiempo creer que el sistema es demasiado difícil de usar y de ahí que los beneficios de desempeño por el uso están compensados por el esfuerzo de usar la aplicación. Esto es, adicionalmente a la utilidad, el uso es teóricamente influenciado por *la facilidad de uso percibida*. (F. Davis, 1989)

4.4 Extensión del Modelo de Aceptación de Tecnología (TAM2)

TAM2 es una extensión del modelo TAM visto anteriormente, diseñado para abarcar las situaciones de voluntariedad y obligación de uso incluyendo dos constructores adicionales. *Norma subjetiva*, define como el grado en que la gente piensa que otros que son importantes para ellos piensan que deben de proceder, fue un determinante significativo de intención para el uso de herramientas en el contexto de uso por obligación. Venkatesh y Davis muestran que la *voluntariedad* percibida define el grado en el cual los

adoptantes potenciales perciben la decisión de adopción de ser obligatorio, significativamente modera el efecto directo de la norma subjetiva en la intención de uso. (Riemenschneider et al., 2002)

4.5 Características de la Innovación Percibidas (PCI)

Moore and Benbasat, basándose en el extensivo trabajo de la *difusión de la innovación* de Rogers, publicaron un instrumento para medir lo que ellos denominaron "Perceived Characteristics of Innovating" (PCI) en relación a la adopción de las *workstations* personales. Siete constructores fueron representados en el instrumento de PCI. Estos incluyen la *ventaja relativa*, que es el grado en que una innovación se percibe como mejor que su precursor, y la *complejidad*, que es el grado en que una innovación se percibe como algo difícil de usar. Notando la equivalencia conceptual, Moore y Benbasat usaron la escala de utilidad percibida de Davis para medir la ventaja relativa y la escala de facilidad de uso para medir la complejidad (en dirección inversa). El instrumento PCI también incluye medidas de compatibilidad, demostrabilidad resultado, la imagen y la visibilidad. La *compatibilidad* se refiere al grado en que una innovación se percibe como coherente con los valores existentes, las necesidades y experiencias previas de los potenciales adoptantes; demostrabilidad resultado se refiere al grado en que una innovación se percibe como susceptible de demostración de las ventajas tangibles. *Imagen* se refiere al grado en el que el uso de una innovación es percibida como para mejorar la propia imagen o el estatus en un entorno social. Y la *visibilidad* se refiere al grado en el cual el resultado de una innovación es observable por otros. Moore y Benbasat también incluyeron una medida de *voluntariedad*, la cual es definida y medida de la misma manera que Venkatesh y Davis. A pesar de que Moore y Benbasat no reportaron la relación entre los siete constructores y su uso o intenciones de uso, Agarwal y Prasad reportaron soporte empírico para PCI como modelo de intención de uso de la World Wide Web. (Riemenschneider et al., 2002)

4.6 Teoría del Comportamiento Planeado (TPB)

Esta teoría introducida por Ajzen y sus colegas en la psicología social, postula tres principales determinantes de intención: control conductual percibido, norma subjetiva y actitud. El *Control conductual percibido* se refiere a la percepción propia de las limitaciones internas y externas sobre la realización de la conducta. De acuerdo con hallazgos de Chau, Taylor y Todd, se representan distintas subdimensiones internas y externas de control conductual percibido como constructos independientes. La *norma subjetiva* es definida y medida de la misma manera que en TAM2. La *actitud* se refiere al grado propio de favorabilidad o no favorabilidad hacia un comportamiento objetivo, y es conceptualmente equivalente al constructo de usabilidad de TAM. (Riemenschneider et al., 2002)

4.7 Modelo de Utilización de la Computadora Personal (MPCU)

Thompson introdujo un modelo de utilización de la computadora personal (MPCU) basado en el modelo de conducta interpersonal de Triandis. El modelo incluye el constructo del *factor social* el cual se refiere a la internalización de la persona de la cultura subjetiva del grupo de referencia, y los acuerdos interpersonales que el individuo ha hecho con los demás. El constructo de *afecto* de MPCU se refiere a la reacción emocional positiva o negativa que un individuo asocia con un acto en particular. El modelo de Triandis incluye consecuencia percibida y Thompson delinea tres tipos: dos a corto plazo y uno a largo plazo. Las consecuencias a corto plazo incluyen *complejidad*, definida y medida esencialmente en orden inverso a la facilidad de uso percibida en el modelo TAM, y *ajuste al trabajo* definida y medida equivalentemente a la utilidad percibida en el modelo TAM. En las consecuencias de *carrera o profesión*, la dimensión de las consecuencias de uso a largo plazo se refiere a los resultados que tienen un beneficio en el futuro, tales como incrementar la flexibilidad para cambiar de trabajo o incrementar las oportunidades de un trabajo más significativo. Finalmente Thompson, incluye *condiciones facilitadoras*, los cuales son factores en el ambiente que hacen un acto fácil de hacer. Este constructo es paralelo al constructo de control de conducta percibido externo de TPB y se equipararon los dos. (Riemenschneider et al., 2002)

4.8 Ingeniería de Software

El Software de computadora es el producto que diseñan y construyen los ingenieros de software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números y texto y también incluyen representaciones de información de audio, video e imágenes. (Pressman, 1998) Son los programas de computadora y la documentación asociada. Los productos de software se pueden desarrollar para un cliente en particular o para un mercado en general. (Sommerville, 2005)

La ingeniería de software es una disciplina de ingeniería que comprende todos los aspectos de la producción de software, la ingeniería de software comprende las prácticas para desarrollar y entregar un software útil. (Sommerville, 2005) Los ingenieros de software lo construyen, y virtualmente cualquier persona en el mundo industrializado lo utiliza directa o indirectamente. (Pressman, 1998)

4.9 El Proceso de desarrollo de software.

Un *proceso* está definido como una serie de acciones u operaciones que conducen a un fin. En general, una empresa u organización requiere de uno o más procesos para lograr sus objetivos, los cuales por lo general involucran la utilización de software. En el caso de una empresa que se dedica al desarrollo de software, se requieren procesos que abarquen desde la creación del software hasta su mantenimiento.

Todo esto es conocido como el *ciclo de vida* del software. Un aspecto básico para manejar la complejidad inherente en el software es contar con un modelo de proceso a seguir. El modelo de proceso define un orden para llevar a cabo los distintos aspectos del proceso. El modelo se puede definir como un grupo de estrategias, actividades, métodos y tareas, que se organizan para lograr un conjunto de metas y objetivos. (Santana Tapia, 2003)

En la profesión de desarrollo de sistemas, los términos *métodos, metodologías, ciclo de vida del proyecto y ciclo de vida de desarrollo de sistemas* son intercambiables. (Yourdon, 2006)

Al revisar la literatura de la ingeniería de software, se observa que se usan como sinónimos *procesos de software*, *metodologías o métodos de desarrollo de software, modelos de desarrollo de software, ciclo de vida del proyecto* y *ciclo de vida de desarrollo de sistemas* y aunque hay ciertas diferencias entre algunos de estos términos, todos se relacionan con la producción de software de computadoras. Para efectos de esta tesis se utilizará el término *metodología de desarrollo*, entendiendo que se pueden utilizar textos originales con los otros términos dependiendo del autor.

Hacia los años sesenta se inicio el desarrollo de aplicaciones informáticas mediante un conjunto de procedimientos que constituyeron finalmente el llamado ciclo de desarrollo lineal o en cascada. Este método ha ido perfeccionándose con la inclusión de nuevas técnicas de análisis de diseño. Nuevas herramientas del tipo CASE y sobre todo con la aportación de nuevas tecnologías de la información como lo han sido la estructuración del software de presentación, aplicación y datos, o la orientación a objetos y el desarrollo de aplicaciones de internet.

Debido a la adopción de estas nuevas tecnologías, han ido surgiendo necesariamente nuevos paradigmas de desarrollo más acordes con el tipo de software a obtener, por lo que implantar una metodología de desarrollo en una organización que fabrique software, no es tarea fácil y, menos aún, rápida. Se requiere una firme decisión por parte de la dirección y un equipo técnico cualificado que esté dispuesto a investigar qué métodos, técnicas y herramientas son las más adecuadas de utilizar para el entorno empresarial en el que se encuentran. (Barranco, 2001)

4.10 Características de las metodologías de desarrollo.

Una metodología de desarrollo de software se fundamenta sobre tres pilares básicos: que hay que hacer y en qué orden, cómo deben realizarse las tareas y con que pueden llevarse a cabo. Esto es, qué *etapas, actividades y tareas* se deben acometer, qué

técnicas deben emplearse para realizar estas actividades y cuáles son las *herramientas software* a utilizar en cada caso.

La metodología a aplicar será más completa y robusta cuantos más elementos añadamos a esta terna. Así por ejemplo, pueden incorporarse a ella todas las actividades de control de calidad a lo largo del ciclo de desarrollo, también pueden incorporarse todas aquellas actividades propias de la gestión del proyecto como la planificación, el seguimiento y el control del proyecto, e incluso aquellas actividades propias de la gestión de versiones, los cambios producidos durante el desarrollo y la migración de componentes entre diferentes entornos de construcción, pruebas y explotación final. Una metodología que recoja todo esto constituye en sí un entorno de ingeniería de software. Sin embargo, no existe en el mercado algo tan completo y a la vez tan ajustado a nuestra manera de trabajar en equipo, por lo que las organizaciones tienden a adquirir un conjunto de herramientas de ayuda a la creación de software para después intentar integrarlas entre sí a base de complejos interfaces. (Barranco, 2001)

4.11 Metodologías de desarrollo.

Los procesos de software juegan un papel importante en la ingeniería de software. El proceso de desarrollo y soporte del software a menudo requiere de muchas tareas diferentes que se tienen que realizar por diferentes personas en alguna secuencia relacionada. Cuando se les deja a los ingenieros de software realizar una tarea basada en su propia experiencia, conocimientos y valores, ellos no necesariamente perciben y realizan las tareas de la misma manera o del mismo orden. Incluso ellos en algunas ocasiones no realizan las mismas tareas. Estas inconsistencias causan que los proyectos tomen más tiempo con un producto terminado más pobre y en situaciones peores, un fracaso total del proyecto. (Tsui & Karam, 2006)

El objetivo de los modelos de procesos de software es proveer una guía para coordinar y controlar sistemáticamente las tareas que deben ser realizadas en orden para alcanzar el producto final y los objetivos del proyecto. Un modelo de proceso define lo siguiente:

- Un conjunto de tareas que necesitan ser realizadas.
- Entradas y salidas de cada tarea

- Precondiciones y post condiciones para cada tarea
- La secuencia y flujo de esas tareas.

Debemos preguntarnos si un proceso de desarrollo de software es necesario si hay solo una persona desarrollando el software. La respuesta es que, depende. Si el proceso de desarrollo de software es visto como solo un agente de coordinación y control, entonces no hay necesidad debido a que solo es una persona. Sin embargo, si el proceso es visto como una perspectiva de guía para generar varios entregables intermedios adicionalmente al código ejecutable, por ejemplo, un documento de diseño, guía de usuario, casos de prueba, entonces aunque sea una persona quien sea el que desarrolle el software puede necesitar un proceso. (Tsui & Karam, 2006)

4.11.1 Métodos Estructurados

Un método estructurado es una forma sistemática de elaborar modelos de un sistema existente o de un sistema que tiene que ser construido. Fueron desarrollados por primera vez en la década de los 70 para soportar el análisis y el diseño del software y evolucionaron en las décadas de los 80 y de los 90 para soportar el desarrollo orientado a objetos. (Sommerville, 2005)

Las metodologías tradicionales o estructuradas se enfocan principalmente en la descomposición funcional de un sistema. El objetivo es lograr una definición completa del sistema en términos de funciones, estableciendo los datos de entrada y salida correspondientes. Se conocen a estas metodologías como *análisis y diseño estructurado (SA/SD, Structured Analysis and Structured Design)*. Durante las actividades de desarrollo se utilizan diferentes herramientas de modelado: (Weitzenfeld, 2004)

- Diagramas de flujo de datos.
- Diagramas de transición de estado.
- Diagramas entidad-relación.

Los métodos estructurados han sido aplicados con éxito en muchos proyectos grandes. Pueden suponer reducciones significativas de coste debido a que utilizan notaciones

estándar y aseguran que se produce una documentación de diseño estándar. Sin embargo, los métodos estructurados tienen una serie de inconvenientes:

1. No proporcionan un soporte efectivo para la comprensión o el modelado de requerimientos del sistema no funcionales.
2. No discriminan en tanto que normalmente no incluyen guías que ayuden a los usuarios a decidir si un método es adecuado para un problema concreto. Tampoco incluyen normalmente consejos sobre cómo pueden adaptarse para su uso en un entorno particular.
3. A menudo generan demasiada documentación. La esencia de los requerimientos del sistema puede quedar oculta por el volumen de detalle que se incluye.
4. Los modelos producidos son detallados, y los usuarios a menudo los encuentran difíciles de entender. Estos usuarios, por lo tanto, no pueden comprobar el realismo de estos modelos. (Sommerville, 2005)

4.11.1.1 Ejemplos de Modelos con Métodos Estructurados.

Modelo de cascada: El primer modelo de proceso de desarrollo de software que se publicó se derivó de procesos de ingeniería de sistemas más generales. Debido a la cascada de una fase a otra, dicho modelo se le conoce como *modelo en cascada*. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

1. Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.
2. Diseño del sistema y del software. El proceso de diseño divide los requerimientos en sistemas de hardware o software. Establece una arquitectura completa del sistema. Identifica y describe las abstracciones fundamentales del sistema de software y sus relaciones.
3. Implementación y prueba de unidades. El diseño del software se lleva a cabo como un conjunto de unidades o programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
4. Integración y prueba del sistema. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar

que se cumplan los requerimientos de software. Después de las pruebas, el sistema se entrega al cliente.

5. Funcionamiento y mantenimiento. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos. (Sommerville, 2005)

Desarrollo Evolutivo: Es una regla más que una excepción, que los requerimientos de los sistemas tanto técnicos como de información, no se conocen por completo desde el comienzo del proyecto. El conocimiento del sistema crece progresivamente como progresa el trabajo. Cuando la primera versión del sistema está en operación, aparecen nuevos requerimientos y los antiguos cambian. De esta forma, un sistema como un todo, no puede ser completamente desarrollado en la creencia de que la especificación de los requerimientos permanecerá constante durante el periodo de desarrollo, el cual puede ser de varios años en sistemas grandes. (Jacobson, 1992) El desarrollo evolutivo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. Existen dos tipos de desarrollo evolutivo:

1. *Desarrollo exploratorio*, donde el objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos propuestos por el cliente.
2. *Prototipos desechables*, donde el objetivo del proceso de desarrollo evolutivo es comprender los requerimientos del cliente y entonces desarrollar una definición mejorada de los requerimientos para el sistema. (Sommerville, 2005)

RUP (Rational Unified Process): El Proceso Unificado de Rational es un marco de proceso de software más que un simple proceso, desarrollado por Rational Software Corporation, la cual fue adquirida por IBM. El origen de RUP está enraizado en el original Objectory Process de 1987 y el Rational Objectory Process de 1997 así como también en el Lenguaje de Modelado Unificado (UML). De muchas maneras, ha incorporado muchas de las primeras experiencias de los

modelos *iterativo, incremental* y *espiral.* Este marco de proceso está impulsado por tres principales conceptos que forman las bases de RUP:

- Casos de uso y requerimientos.
- Centrada en la Arquitectura.
- Iterativo e incremental. (Tsui & Karam, 2006)

4.11.2 Métodos Orientados a Objetos.

Los métodos tradicionales tratan a las funciones y los datos por separado. Tal enfoque a menudo conduce a problemas durante el mantenimiento debido a que la estructura *datos/funciones* es muy sensible a los cambios. Los métodos orientados a objetos no separan las funciones y los datos, sino que los ven como un todo integrado. (Jacobson, 1992)

Las metodologías *orientadas a objetos* se enfocan principalmente en el modelado de un sistema en términos de objetos. A diferencia de las metodologías estructuradas, se identifican inicialmente los objetos del sistema para luego especificar su comportamiento. (Weitzenfeld, 2004)

El análisis orientado a objetos tiene por objetivo la comprensión del sistema a ser desarrollado y construir un modelo lógico del sistema. Este modelo es basado en objetos naturales encontrados en el dominio del problema. Los objetos mantienen los datos y tienen un comportamiento en términos de que el comportamiento total del sistema puede ser expresado. Ya que los objetos en el dominio del problema serán estables, la estructura general del sistema normalmente será bastante estable. Los cambios ocurrirán, pero ya que los cambios a menudo vienen del dominio, se espera que esos cambios debieran de ser locales y afectar tan pocos objetos como sea posible. (Jacobson, 1992)

Durante las actividades de desarrollo se utilizan diferentes herramientas de modelado: (Weitzenfeld, 2004)

- Diagrama de clases
- Diagrama de casos de uso
- Diagrama de transición de estado

- Diagrama de secuencia
- Diagrama de colaboración
- Diagrama de subsistema

4.11.2.1 Ejemplos de metodologías orientadas a objetos.

OMT (Object Modeling Technique): La metodología OMT fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric.

OMT es una de las metodologías de análisis y diseño orientadas a objetos más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

4.11.3 Métodos Ágiles.

Los procesos Ágiles son una familia de metodologías de desarrollo de software que producen software en iteraciones cortas y permiten cambios mayores en el diseño. Debería notarse desde el principio que no todas las características de los procesos Ágiles son nuevas y revolucionarias. Muchas son derivadas de años de experiencia y son similares a los procesos iterativos e incrementales. (Tsui & Karam, 2006)

Aunque no hay una definición absoluta sobre de que está constituida una metodología Ágil, hay varias características que son compartidas por la mayoría de los métodos ágiles. Los siguientes puntos son una lista de las características de representan a los métodos ágiles:

- Entregas e iteraciones cortas
- Diseño incremental
- Involucramiento del usuario

- Mínima documentación
- Comunicación informal
- Asumir que los requerimientos y entorno cambian

Cuando se sigue este enfoque interesante y flexible, es importante asegurar que nadie abuse de la metodología, en particular la relacionada con la documentación. Es evidente, que suficiente documentación debe de estar disponible si el software liberado necesita ser mantenido por un grupo diferente de los desarrolladores originales. (Tsui & Karam, 2006)

4.11.3.1 Ejemplos de metodologías ágiles.

Programación Extrema (XP): XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Canós et al., s.d.)

SCRUM: Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Canós et al., s.d.)

Crystal Methodologies: Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo

de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). (Canós et al., s.d.)

Dynamic Systems Development Method (DSDM): Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases. (Canós et al., s.d.)

4.11.4 Métodos Formales.

Los métodos de la ingeniería de software se pueden desglosar sobre un espectro de “formalidad”, que va unido ligeramente al grado de rigor matemático que se aplica durante los métodos del análisis y diseño. Para crear modelos de análisis y diseño, se utiliza una combinación de diagramas, texto, tablas y notaciones sencillas aplicando sin embargo poco rigor matemático.

A continuación se considera el extremo opuesto del espectro de formalidad. Aquí se describe una especificación y un diseño utilizando una sintaxis y una semántica formal que especifican el funcionamiento y el comportamiento del sistema. La especificación formal tiene una forma matemática. Estos métodos permiten al ingeniero de software crear una especificación sin ambigüedades que sea más completa y constante que las que se utilizan en métodos convencionales o los orientados a objetos. (Pressman, 1998)

Capítulo 5. Metodología

5.1 Población objetivo

Para seleccionar una muestra, lo primero es definir nuestra unidad de análisis (personas, organizaciones, periódicos, etc.), el 'quiénes van a ser medidos", depende de precisar claramente el problema a investigar y los objetivos de la investigación (Hernández, Fernández, & Baptista, 1991).

Una vez que se ha definido cuál será nuestra unidad de análisis, se procede a delimitar la población que va a ser estudiada y sobre la cual se pretende generalizar los resultados. Así, una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones. La muestra suele ser definida como un subgrupo de la población. Para seleccionar la muestra deben delimitarse las características de la población. Por otro lado las muestras no probabilísticas o también llamadas muestras dirigidas suponen un procedimiento de selección informal y un poco arbitraria. Aún así se utilizan en muchas investigaciones y a partir de ellas se hacen inferencias sobre la población. La muestra dirigida selecciona sujetos "típicos" con la vaga esperanza de que sean casos representativos de una población determinada. (Hernández et al., 1991)

La ventaja de una muestra no probabilística es su utilidad para un determinado diseño de estudio, que requiere no tanto de una "representatividad de elementos de una población, sino de una cuidadosa y controlada elección de sujetos con ciertas características especificadas previamente en el planteamiento del problema" (Hernández et al., 1991). Hay varias clases de muestras dirigidas y la que se usará en este estudio es la muestra de sujetos-tipo, ésta se utiliza en estudios exploratorios y en investigaciones de tipo cualitativo, donde el objetivo es la riqueza, profundidad y calidad de la información, y no la cantidad, y estandarización. En estudios de perspectiva fenomenológica donde el objetivo es analizar los valores, ritos y significados de un determinado grupo social, el uso de muestras de sujetos-tipo es frecuente. (Hernández et al., 1991)

En la aplicación de la encuesta se obtuvo la siguiente distribución de la muestra, que por motivos de confidencialidad solo se mostrarán roles o funciones dentro de la Dirección de Desarrollo de Sistemas de Información donde se aplicó el cuestionario, omitiendo el nivel jerárquico. Cabe hacer mención que operativamente los entrevistados tenían varios roles y que para fines de esta encuesta se tomo el rol principal, la característica en común es

que todos conocen la metodología de desarrollo que se utiliza en la Dirección de Desarrollo de Sistemas de Información o bien pueden establecer las diferencias entre usar o no una metodología, que es la principal característica que se persigue para efectos del presente estudio:

- 3 Líderes de proyecto
- 1 Administrador de base de datos
- 40 Analistas / Desarrolladores

La muestra se compone de unidades de análisis con un promedio de edad de 35 años, los cuales reportan en sus roles una experiencia de 9 años promedio y una antigüedad promedio dentro del instituto de 6 años, 15 de los encuestados son mujeres y 29 hombres; en el aspecto de tipo de contrato son 21 personas que tienen contrato tipo tradicional

5.2 Características de la población objetivo

Las características que se tomarán en cuenta para determinar la población objetivo son las siguientes:

5.2.1 Estabilidad Laboral

La estabilidad laboral nos ayudará en el estudio para determinar si éste aspecto determina la disponibilidad hacia la adopción de las tecnologías de la información, ya que de manera empírica se tiene la creencia que la gente con mayor riesgo laboral tiene una mejor disposición para adaptarse a los cambios tecnológicos con respecto de la que tiene una estabilidad laboral fija. (Ávila, 2010)

La anterior afirmación se generalizará hacia la adopción de las metodologías de desarrollo de software que es el motivo de la presente investigación.

Para esta investigación solo existirán dos categorías: contratación eventual, que significa que el trabajador tiene un tiempo definido dentro de la organización, y contratación federalizada, que son los trabajadores quienes cuentan con una estabilidad laboral mayor que los de contratación eventual por el hecho de que su contrato no tiene la fecha de

término y dependen directamente de un presupuesto fijo anual dentro de la organización. (Ávila, 2010)

5.2.2 Género

El género es definido como el sexo biológico. Hay en crecimiento investigaciones completas sobre las implicaciones del sexo en las tecnologías de información. Las diferencias entre hombres y mujeres han sido estudiadas en varios contextos incluyendo email y mensajería instantánea. (Ávila, 2010)

Existen numerosos estudios sobre las diferencias entre hombres y mujeres con respecto a la aceptación de las tecnologías de información; en la presente investigación también se tratará de llevar esta generalización a la aceptación de las metodologías de desarrollo.

5.2.3 Edad

La edad es un factor importante en las tecnologías de la información, ya que cada vez hay gente de mayor edad en la fuerza de trabajo. (Venkatesh, 2000) encontró que a corto plazo, normas subjetivas, actitud hacia usar la tecnología y el control de comportamiento percibido tienen un significante impacto sobre trabajadores más grandes mientras solo la actitud hacia usar la tecnología tiene impacto sobre trabajadores más jóvenes. La actitud hacia usar la tecnología es más importante para trabajadores jóvenes que para trabajadores más viejos. (Ávila, 2010)

Como resultado global (Venkatesh, 2000) encontraron que los trabajadores más jóvenes, cuando se comparan con los trabajadores más viejos, estuvieron más inclinados a usar las tecnologías. (Ávila, 2010)

El estudio de este factor también se generalizará hacia el uso de las metodologías de desarrollo de software.

5.3 Instrumento de medición

En (Riemenschneider et al., 2002) se construyó y validó un cuestionario mismo que se utilizará en este documento y el cual fue elaborado de la siguiente forma:

Se construyó usando escalas de medición de investigaciones previas. Los constructos de la *intención del comportamiento* fueron adaptados de escalas estándar como las usadas por (Ajzen & Fishbein, 1980), (Taylor & Todd, 1995), (Venkatesh & F. Davis, 2000). Los constructos de *facilidad de uso y utilidad* se tomaron del instrumento original de (F. Davis, 1989). Las escalas de *voluntariedad, compatibilidad, demostrabilidad resultante, imagen y visibilidad* fueron adaptadas de (Moore & Benbasat, 1991). Los constructos de *PBC-I, PBC-E y consecuencias de carrera* se incluyeron de (P. Chau, 1996) y (Thompson, Higgins, & Howell, 1991). Los constructos de *norma subjetiva* vienen de (Venkatesh & F. Davis, 2000) y (Ajzen & Fishbein, 1980). Eliminaron los constructos de afecto, los cuales no parecen ser apropiados en el contexto de las metodologías (el afecto pertenece a una respuesta emocional). La eliminación de los constructos de afecto es consistente con estudios previos de este tipo (Taylor & Todd, 1995).

El cuestionario se medirá en la escala de Likert de 7 puntos, los cuales se muestran a continuación:

- Totalmente en desacuerdo
- Fuertemente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Fuertemente de acuerdo
- Totalmente de acuerdo

El número de preguntas es de 45, divididas en 12 factores o constructos. La medición de los constructos con las preguntas del cuestionario se presenta con la siguiente distribución; en las preguntas se mencionarán las siglas MDS las cuales hacen referencia a la **M**etodología de **D**esarrollo de **S**oftware que se usa en la Dirección de Área encuestada, eso se hace para mantener la confidencialidad de los procesos del área y para dar la posibilidad de inclusión de otras metodologías, así mismo se mencionan las siglas de DDA para hacer referencia a la **D**irección **D**e **Á**rea:

- Intención de comportamiento
    1. Tengo la firme intención de utilizar la metodología MDS en mi trabajo para mi próximo proyecto

2. Si se presenta la oportunidad, usaría la metodología MDS

- Utilidad / Ventaja relativa / Actitud / Compatibilidad en el trabajo
    1. Si aplico la metodología MDS me permitirá mejorar mi trabajo
    2. Al usar la metodología MDS se incrementa mi productividad (hacer mas en el mismo tiempo)
    3. El usar la metodología MDS realza la calidad de mi trabajo
    4. Si uso la metodología MDS mi trabajo será más fácil
    5. Las ventajas de usar la metodología MDS superan a las desventajas
    6. La metodología MDS es útil en mi trabajo

- Facilidad de uso / Complejidad
    1. Aprender la metodología MDS fue fácil para mi
    2. Creo que la metodología MDS fue clara y entendible
    3. Usar la metodología MDS no requiere de mucho esfuerzo mental
    4. Encuentro a la metodología MDS fácil de usar
    5. La metodología MDS no es tediosa
    6. Usar la metodología MDS no quita mucho tiempo de las actividades diarias

- Norma Subjetiva / Factores Sociales
    1. Las personas que influyen en mi comportamiento piensan que debería de usar la metodología MDS
    2. Las personas que considero importantes para mí piensan que debería usar la metodología MDS
    3. Los compañeros de trabajo piensan que debería de usar la metodología MDS

- Voluntariedad
    1. Aunque podría ser útil, usar la metodología MDS no es obligatorio en mi trabajo
    2. Mi jefe no me pide que use la metodología MDS
    3. Para mí el uso de la metodología MDS es voluntario
- Compatibilidad

1. La metodología MDS es compatible con la manera en que desarrollo sistemas
2. El uso de la metodología MDS es compatible en todos los aspectos de mi trabajo
3. Usar la metodología MDS encaja bien a la forma como trabajo

- Imagen
    1. La gente en la DDA que usa la metodología MDS tiene más prestigio que aquellas que no la usan
    2. La gente en la DDA que usa la metodología MDS tiene un perfil alto
    3. Usar la metodología MDS es un símbolo de estatus en la DDA

- Visibilidad
    1. La metodología MDS tiene mucha presencia en la DDA
    2. Es fácil observar a otros usando la metodología MDS
    3. He tenido muchas oportunidades de ver la metodología MDS siendo utilizada
    4. Puedo ver cuando otros usan la metodología MDS en mi departamento

- Control del Comportamiento Percibido – Interno
    1. Creo que no hay diferencia entre mis habilidades y conocimientos existentes y las requeridas por la metodología MDS
    2. Tengo el conocimiento necesario para usar la metodología MDS

- Control del Comportamiento Percibido – Externo / Condiciones de Facilidad
    1. Tengo disponible capacitación y educación especializada de la metodología MDS
    2. Tengo disponible orientación formal para usar la metodología MDS
    3. Tenemos un grupo específico para asistencia con las dificultades de la metodología MDS
    4. Los directivos proveen toda la ayuda y recursos necesarios para permitir a la gente usar la metodología MDS
- Consecuencias de la carrera

1. Saber la metodología MDS me pone a la vanguardia en mi campo de estudio
2. El conocimiento de la metodología MDS incrementa mi oportunidad de promoción
3. Conocer la metodología MDS incrementa mi flexibilidad para cambiar de trabajo
4. El conocimiento de la metodología MDS incrementa la oportunidad de tener un trabajo más significativo
5. Conocer la metodología MDS puede incrementar la oportunidad de ganar seguridad en el trabajo

- Demostrabilidad resultado
    1. No tendría dificultad en decirle a otros el resultado de usar la metodología MDS
    2. Creo que podría comunicarle a otros las consecuencias de usar la metodología MDS
    3. Los resultados de usar la metodología MDS son evidentes para mí
    4. No tendría dificultad explicando porque la metodología MDS puede o no ser beneficiosa.

Las variables que se están tratando de medir y su descripción operacional se describen en las tablas 5.3.a, 5.3.b y 5.3.c:

Tabla 5.3.a Mapa de constructos

Modelos					
Constructo	TAM	TAM2	PCI	TPB	MPCU
Utilidad	Utilidad	Utilidad	Ventaja relativa	Actitud	Ajuste al trabajo
Facilidad de uso	Facilidad de uso	Facilidad de uso	Complejidad		Complejidad
Norma subjetiva		Norma Subjetiva		Norma Subjetiva	Factores sociales
Afecto					Afecto
Voluntariedad		Voluntariedad	Voluntariedad		
Compatibilidad			Compatibilidad		
Demostrabilidad resultado			Demostrabilidad resultado		
Imagen			Imagen		
Visibilidad			Visibilidad		
Control del Comportamiento Percibido – Interno				CCP - Interno	
Control del Comportamiento Percibido – Externo				CCP – Externo	Condiciones de facilidad
Consecuencias de carrera					Consecuencias de carrera

- En la tabla 5.3.a se muestran los diferentes modelos que se utilizaron en el presente estudio, y los constructos que componen a cada modelo. Esta tabla muestra como los constructos aparecen en los diferentes modelos y solo cambian de denominación en algunos de ellos pero siguen representando a la misma variable.

Tabla 5.3.b Tabla de constructos e hipótesis

Constructos	Hipótesis asociadas
Aceptación de tecnología	H1.a Percepción de fácil uso H1.b Percepción de Utilidad H1.c Normas Subjetivas H1.d Voluntariedad
Características de innovación	H2.a Ventaja relativa H2.b Complejidad H2.c Compatibilidad H2.d Voluntariedad H2.e Demostrabilidad resultado H2.f Imagen H2.g Visibilidad
Comportamiento planificado	H3.a Norma subjetiva H3.b Actitud H3.c Control del comportamiento percibido interno H3.c Control del comportamiento percibido externo
Utilización de la computadora personal	H4.a Factores sociales H4.b Complejidad H4.c Compatibilidad en el trabajo H4.d Consecuencias de carrera H4.e Condiciones de facilidad H4.f Afecto

- En la tabla 5.3.c se muestran la relación de los constructos con sus variables relacionadas. También se eliminan la hipótesis de afecto por los resultados del experimento llevado a cabo por (Riemenschneider et al., 2002)

Tabla 5.3.c Tabla de hipótesis y variables

Hipótesis	Variable	Descripción operacional
H1.a Percepción de fácil uso	1. Facilidad de uso	Es la medida en la cual una persona piensa que usar un sistema estará libre de esfuerzo.
H1.b Percepción de utilidad	2. Utilidad	Es la medida en la cual una persona piensa que usar un sistema mejorará su desempeño
H1.c Normas subjetivas	3. Percepción de las opiniones de terceros	Es el grado en que una persona piensa que otros quienes son importantes para ella piensan que debería realizar un comportamiento
H1.d Voluntariedad	4. No obligatoriedad	Es el grado en el cual los potenciales adoptantes perciben la decisión de adopción como no obligatoria
H2.a Ventaja relativa	Misma variable que utilidad	
H2.b Complejidad	Misma variable que facilidad de uso	Orden inverso de facilidad de uso
H2.c Compatibilidad	5. Consistencia con valores y necesidades existentes	Se refiere al grado en que una innovación es percibida como consistente con las necesidades y valores existentes y experiencias pasadas de los potenciales adoptantes
H2.d Demostrabilidad resultado	6. Ventajas tangibles	Es el grado en que una innovación es percibida como dócil para demostrar ventajas tangibles
H2.e Imagen	7. Realce de la imagen personal o estatus	Se refiere al grado el cual el uso de una innovación es percibida que realza el estatus o la imagen de una persona en un sistema social
H2.f Visibilidad	8. Visibilidad a otros	Se refiere al grado en el cual el resultado de una innovación es observable por otros
H2.g Voluntariedad	Misma variable que Voluntariedad	
H3.a Control del comportamiento percibido	9. Restricciones de comportamiento	Se refiere a las percepciones de una persona sobre las restricciones internas y externas al realizar un comportamiento
H3.b Normas subjetivas	Misma variable que normas subjetivas	
H3.c Actitud	Misma variable que utilidad	
H4.a Factores sociales	Misma variable que norma subjetiva	
H4.b Complejidad	Misma variable que facilidad de uso	
H4.c Compatibilidad en el trabajo	Misma variable que utilidad	
H4.d Consecuencias de carrera	10. Recompensas en el trabajo	Se refiere a las recompensas a largo plazo en el ambiente profesional o laboral
H4.e Condiciones de facilidad	11. Facilidades	Son factores en el ambiente que hacen de una actividad fácil de hacer
H4.f Afecto	Variable que se desecho posteriormente por los investigadores en (Riemenschneider et al., 2002)	

5.3.1 Ajustes al instrumento

Con el fin de eliminar ambigüedades que pueda traer el documento o falta de claridad en las preguntas, el cuestionario se sometió a evaluación por cinco personas que estuvieron aportando mejoras al mismo y dejando el cuestionario como se muestra en el punto anterior. Estas personas no participaron en la encuesta final.

Varias de las preguntas sufrieron una corrección en la redacción debido a la falta de claridad en las mismas. De esta revisión quedo la versión final para aplicarse en la dirección de área destino.

5.4 Recolección de información

En esta tesis la recolección de los datos fue a través de una encuesta, la cual se aplicó a una muestra de la Dirección de Desarrollo de Sistemas de Información. Los usuarios se clasificaron con base a lo descrito en el punto 5.2 de este documento.

El procedimiento de recolección de estos datos se hizo mediante un cuestionario que se envió por correo electrónico y por entrega personal impreso en papel. Dicho cuestionario fue contestado por los participantes, en este caso desarrolladores de software, de la muestra objetivo y fue recuperado por los mismos medios. Por la naturaleza en la preparación técnica de las personas encuestadas no requerirán de apoyo informático para el caso de los cuestionarios enviados por correo electrónico.

5.4.1 Tamaño de la muestra

La muestra es en esencia un subgrupo de la población. Es un subconjunto de elementos que pertenecen a ese conjunto definido en sus características al que llamamos población. En realidad, pocas veces se puede medir a toda la población, por lo que obtenemos o seleccionamos una muestra y se pretende que este subconjunto sea un reflejo fiel del conjunto de la población. (Hernández et al., 1991)

Según (Ávila, 2010), para hacer la muestra probabilística es necesario conocer algunos términos y sus definiciones:

- La población, a la que suele denominarse como *N*, es un conjunto de elementos.

- La muestra, la que se simboliza como *n,* la cual es un subconjunto de la población *N.*

- En una población N (previamente delimitada por los objetivos de la investigación), nos interesa establecer valores de las características de los elementos de N.

Nos interesa conocer los valores promedio de la población, lo cual se expresa como:

- $\bar{}$ = al valor de una variable determinada (Y) que nos interesa conocer, digamos un promedio.

También nos interesa conocer:

- V = la varianza de la población con respecto a determinadas variables (la varianza indica la variabilidad)

Como los valores de la población no se conocen, seleccionamos una muestra n, a través de los estimados de la muestra, inferimos valores de la población ($\bar{}$ será la estimación de $\bar{}$, el cual desconocemos).

En la muestra, $\bar{}$ es un estimado promedio que podemos determinar. Sabemos que en nuestra estimación habrá una diferencia ($\bar{}$ -$\bar{}$ =?), es decir, un error, el cual dependerá del número de elementos muestreados. A dicho error de le conoce como error estándar (se):

- se = la desviación estándar de la distribución muestral y representa fluctuación de $\bar{}$.

- $(se)^2$ = error estándar al cuadrado, cuya fórmula nos servirá para calcular la varianza (V) de la población (N), así como la varianza de la muestra (n) será la expresión s^2.

- s^2 = varianza de la muestra, la cual podrá determinarse en términos de probabilidad donde s^2= p(1-p).

- p = porcentaje estimado de la muestra, probabilidad de ocurrencia del fenómeno, la cual se estima sobre marcos de muestreo previos o se define, la

certeza total siempre es igual a uno, las posibilidades a partir de esto son “p” de que si ocurra y “q” de que no ocurra (p + q=1). De aquí se deriva 1 – p.

Como se ha podido observar, cuando hablamos de un término de la muestra se simboliza con una letra minúscula (b,s,se). Si se trata de un término de la población, se simboliza con una letra mayúscula (N,S).

Para determinar la muestra probabilística necesitamos principalmente dos cosas: determinar el tamaño de la muestra (n) y seleccionar los elementos muestrales, de manera que todos tengan la misma posibilidad de ser elegidos. Para lo primero, tenemos una fórmula que contiene las expresiones ya descritas. Para el segundo, requerimos un marco de selección adecuado y un procedimiento que permita la aleatoriedad en la selección.

Para definir el tamaño de la muestra según (Ávila, 2010), comenta que necesitamos saber que dado a que tenemos una población N en nuestro universo, ¿cuál es el menor número de unidades muestrales (personas) que necesito para conformar una muestra (n) que me asegure un determinado nivel de error estándar, por ejemplo menor que 0.05?

Para el caso de este trabajo de tesis como respuesta a esta pregunta, buscaremos la probabilidad de la ocurrencia de $\bar{}$, así como que el estimado de $\bar{}$ se acerque a $\bar{}$, el valor real de la población. Estableceremos el error estándar en 0.05, con esto sugerimos que la fluctuación promedio de nuestro estimado $\bar{}$ con respecto de los valores reales de la población $\bar{}$ no sea > 0.05, es decir que de 100 casos, 95 veces la predicción sea correcta y que el valor de $\bar{}$ se sitúe en un intervalo de confianza que comprenda el valor de $\bar{}$.

Las formulas para determinar el tamaño de muestra son:

1. — = Tamaño provisional de la muestra1 = varianza de la muestra/varianza de la población.

2. ———

De lo cual obtuvimos que para nuestro caso, partiendo que el universo es de 50 personas, con un nivel de confianza del 95%, nuestra muestra n es igual a 44 elementos. Por ser una población pequeña prácticamente se tuvieron que tomar todos los elementos de ésta para la muestra.

De acuerdo a las características de la población, de las personas quienes participaron en la encuesta y de la confidencialidad que se manejo, se obtuvo la siguiente distribución según la tabla 5.4.1.a:

Tabla 5.4.1.a Distribución de la muestra por roles

Rol	Cantidad	Participación
Líder de proyecto	3	6.82%
Analista/Desarrollador	40	90.90%
DBA	1	2.28%

Capítulo 6. Análisis de resultados

6.1 Análisis factorial

El análisis factorial es un método estadístico multivariado para determinar el número y naturaleza de un grupo de constructos que están subyacentes en un conjunto de mediciones. En este análisis se generan "variables artificiales" (denominadas factores) que representan constructos. Los factores son obtenidos de las variables originales y deben ser interpretados de acuerdo a éstas. Es una técnica para explicar un fenómeno complejo en función de unas cuantas variables. (Hernández et al., 1991)

Mediante un paquete estadístico se dio validez al cuestionario, en el cual se obtuvo el índice de adecuación de KMO a los datos de la muestra, obteniéndose un valor de 0.743, el cual se interpreta como aceptable. La prueba de esfericidad de Barlett ($p<0.0001$) indica que existe una adecuada correlación entre variables y que por lo tanto los datos son susceptibles al análisis factorial (Ávila, 2010).

Dicho análisis se muestra a continuación:

La prueba de Bartlett se utiliza para verificar si la matriz de correlaciones es una matriz identidad, es decir, si todos los coeficientes de la diagonal son iguales a la unidad y los externos a la diagonal son iguales a cero. Este estadístico se obtiene a partir de la transformación X^2 del determinante de la matriz de correlaciones y cuanto mayor sea y por tanto menor el grado de significación, más improbable que la matriz sea una matriz de identidad. (Ávila, 2010)

Para el caso de la presente investigación, se obtuvo un grado se significación de $p<0.0001$, lo cual nos indica que no se trata de una matriz de identidad, y por lo tanto podemos realizar el análisis factorial de los datos.

Tabla 6.1.a Prueba de KMO y Bartlett

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.743
Bartlett's Test of Sphericity	Approx. Chi-Square	691.687
	df	300
	Signification	.000

Índice de correlación de Kaiser-Meyer-Olkin (KMO)

Éste índice compara los coeficientes de correlación de Pearson obtenidos en la Tabla 6.1.b con los coeficientes de correlación parcial entre las variables. Si los coeficientes de correlación parcial entre las variables son muy pequeños, esto quiere decir que la relación entre cada par de las mismas se debe o puede ser explicada por el resto y por lo tanto se puede llevar a cabo un análisis factorial de los datos. Si la suma de los coeficientes de correlación parcial al cuadrado es muy pequeña, el KMO será un índice muy aproximado a la unidad y por lo tanto el análisis factorial un procedimiento adecuado. En cambio, valores pequeños en este índice nos darán a entender todo lo contrario. (Ávila, 2010)

Tabla 6.1.b Matriz de correlación (1ª. Parte)

		v2	v3	v4	v5	v6	v7	v8	v9	v10	v13
Correlation	v2	1.000	0.791	0.481	0.685	0.405	0.527	0.517	0.241	0.406	0.244
	v3	0.791	1.000	0.444	0.621	0.416	0.519	0.420	0.162	0.210	0.351
	v4	0.481	0.444	1.000	0.470	0.451	0.419	0.557	0.319	0.348	0.335
	v5	0.685	0.621	0.470	1.000	0.595	0.648	0.459	0.080	0.157	0.227
	v6	0.405	0.416	0.451	0.595	1.000	0.561	0.451	0.086	0.069	0.239
	v7	0.527	0.519	0.419	0.648	0.561	1.000	0.638	0.170	0.217	0.377
	v8	0.517	0.420	0.557	0.459	0.451	0.638	1.000	0.435	0.455	0.219
	v9	0.241	0.162	0.319	0.080	0.086	0.170	0.435	1.000	0.808	0.280
	v10	0.406	0.210	0.348	0.157	0.069	0.217	0.455	0.808	1.000	0.343
	v13	0.244	0.351	0.335	0.227	0.239	0.377	0.219	0.280	0.343	1.000
	v14	0.432	0.461	0.352	0.260	0.193	0.218	0.377	0.489	0.517	0.553
	v15	0.019	-0.128	0.067	0.044	0.061	0.188	0.112	0.081	0.191	0.152
	v16	0.322	0.134	0.349	0.264	0.167	0.445	0.474	0.343	0.426	0.274
	v21	0.385	0.282	0.412	0.208	0.194	0.383	0.420	0.228	0.405	0.402
	v22	0.197	0.039	0.231	0.129	0.290	0.295	0.383	0.047	0.161	0.096
	v23	0.397	0.269	0.464	0.248	0.170	0.166	0.389	0.354	0.348	0.195
	v24	0.348	0.341	0.090	0.199	0.056	0.161	0.167	0.225	0.356	-0.036
	v28	0.102	-0.077	0.175	-0.069	0.022	0.048	0.138	0.379	0.434	0.248
	v30	0.161	-0.077	0.343	-0.049	0.068	0.012	0.338	0.440	0.501	0.112
	v37	0.500	0.468	0.327	0.465	0.434	0.581	0.499	0.305	0.307	0.304
	v39	0.379	0.160	0.262	0.177	0.208	0.448	0.508	0.394	0.558	0.157
	v41	0.151	-0.059	0.281	0.011	0.188	0.120	0.407	0.355	0.523	0.157
	v42	0.390	0.446	0.473	0.396	0.335	0.395	0.499	0.522	0.456	0.304
	v44	0.401	0.411	0.487	0.473	0.334	0.479	0.604	0.400	0.399	0.271
	v45	0.447	0.459	0.497	0.469	0.469	0.539	0.509	0.294	0.289	0.274

Tabla 6.1.b Matriz de correlación (2ª. Parte)

		v14	v15	v16	v21	v22	v23	v24	v28	v30	v37	v39
Correlation	v2	0.432	0.019	0.322	0.385	0.197	0.397	0.348	0.102	0.161	0.500	0.379
	v3	0.461	-0.128	0.134	0.282	0.039	0.269	0.341	-0.077	-0.077	0.468	0.160
	v4	0.352	0.067	0.349	0.412	0.231	0.464	0.090	0.175	0.343	0.327	0.262
	v5	0.260	0.044	0.264	0.208	0.129	0.248	0.199	-0.069	-0.049	0.465	0.177
	v6	0.193	0.061	0.167	0.194	0.290	0.170	0.056	0.022	0.068	0.434	0.208
	v7	0.218	0.188	0.445	0.383	0.295	0.166	0.161	0.048	0.012	0.581	0.448
	v8	0.377	0.112	0.474	0.420	0.383	0.389	0.167	0.138	0.338	0.499	0.508
	v9	0.489	0.081	0.343	0.228	0.047	0.354	0.225	0.379	0.440	0.305	0.394
	v10	0.517	0.191	0.426	0.405	0.161	0.348	0.356	0.434	0.501	0.307	0.558
	v13	0.553	0.152	0.274	0.402	0.096	0.195	-0.036	0.248	0.112	0.304	0.157
	v14	1.000	0.027	0.426	0.409	0.225	0.534	0.427	0.286	0.323	0.235	0.266
	v15	0.027	1.000	0.599	0.177	0.185	0.122	0.198	0.334	0.322	0.223	0.457
	v16	0.426	0.599	1.000	0.286	0.244	0.417	0.345	0.348	0.507	0.350	0.616
	v21	0.409	0.177	0.286	1.000	0.687	0.493	0.286	0.204	0.309	0.321	0.439
	v22	0.225	0.185	0.244	0.687	1.000	0.517	0.271	0.198	0.290	0.239	0.456
	v23	0.534	0.122	0.417	0.493	0.517	1.000	0.430	0.209	0.464	0.064	0.264
	v24	0.427	0.198	0.345	0.286	0.271	0.430	1.000	0.316	0.244	0.082	0.327
	v28	0.286	0.334	0.348	0.204	0.198	0.209	0.316	1.000	0.457	0.156	0.439
	v30	0.323	0.322	0.507	0.309	0.290	0.464	0.244	0.457	1.000	0.081	0.457
	v37	0.235	0.223	0.350	0.321	0.239	0.064	0.082	0.156	0.081	1.000	0.528
	v39	0.266	0.457	0.616	0.439	0.456	0.264	0.327	0.439	0.457	0.528	1.000
	v41	0.228	0.451	0.441	0.471	0.483	0.281	0.159	0.296	0.544	0.433	0.625
	v42	0.414	0.140	0.410	0.353	0.119	0.380	0.260	0.003	0.367	0.237	0.279
	v44	0.236	0.075	0.405	0.369	0.214	0.323	0.094	-0.024	0.289	0.393	0.327
	v45	0.234	0.014	0.343	0.267	0.105	0.329	0.111	0.052	0.131	0.300	0.420

Tabla 6.1.b Matriz de correlación (3ª. Parte)

		v41	v42	v44	v45
Correlation	v2	0.151	0.390	0.401	0.447
	v3	-0.059	0.446	0.411	0.459
	v4	0.281	0.473	0.487	0.497
	v5	0.011	0.396	0.473	0.469
	v6	0.188	0.335	0.334	0.469
	v7	0.120	0.395	0.479	0.539
	v8	0.407	0.499	0.604	0.509
	v9	0.355	0.522	0.400	0.294
	v10	0.523	0.456	0.399	0.289
	v13	0.157	0.304	0.271	0.274
	v14	0.228	0.414	0.236	0.234
	v15	0.451	0.140	0.075	0.014
	v16	0.441	0.410	0.405	0.343
	v21	0.471	0.353	0.369	0.267
	v22	0.483	0.119	0.214	0.105
	v23	0.281	0.380	0.323	0.329
	v24	0.159	0.260	0.094	0.111
	v28	0.296	0.003	-0.024	0.052
	v30	0.544	0.367	0.289	0.131
	v37	0.433	0.237	0.393	0.300
	v39	0.625	0.279	0.327	0.420
	v41	1.000	0.183	0.234	0.035
	v42	0.183	1.000	0.692	0.590
	v44	0.234	0.692	1.000	0.720
	v45	0.035	0.590	0.720	1.000

En la tabla 6.1.c se observan siete variables con valores superiores a la unidad, lo que nos indica que éste es el número que se extraerá para el análisis estadístico. Los valores se toman en porcentajes individuales y acumulados sobre la varianza total explicada por cada variable para la solución rotada, como para la no rotada.

Tabla 6.1.c Varianza total explicada

	Valores Iniciales		Extracción de Sumas de Cargas Cuadradas			Rotación de las Sumas de Cargas Cuadradas			
Variable	Total	% de Varianza	Acumulado %	Total	% de Varianza	Acumulado %	Total	% de Varianza	Acumulado %
1	8.819	35.276	35.276	8.819	35.276	35.276	4.190	16.760	16.760
2	3.264	13.058	48.334	3.264	13.058	48.334	3.390	13.559	30.319
3	1.850	7.400	55.734	1.850	7.400	55.734	2.927	11.709	42.028
4	1.564	6.256	61.989	1.564	6.256	61.989	2.583	10.332	52.360
5	1.372	5.488	67.477	1.372	5.488	67.477	2.455	9.819	62.180
6	1.205	4.819	72.296	1.205	4.819	72.296	1.960	7.838	70.018
7	1.041	4.163	76.459	1.041	4.163	76.459	1.610	6.441	76.459
8	0.845	3.381	79.840						
9	0.732	2.928	82.768						
10	0.600	2.401	85.169						
11	0.550	2.199	87.369						
12	0.450	1.799	89.168						
13	0.402	1.608	90.775						
14	0.373	1.490	92.265						
15	0.369	1.477	93.742						
16	0.306	1.225	94.968						
17	0.261	1.044	96.012						
18	0.227	0.906	96.918						
19	0.170	0.680	97.598						
20	0.163	0.652	98.250						
21	0.144	0.577	98.827						
22	0.098	0.393	99.220						
23	0.080	0.321	99.540						
24	0.068	0.272	99.813						
25	0.047	0.187	100.000						

En la tabla 6.1.c se percibe que podemos utilizar siete variables del cuestionario (15.5%) para explicar el 76.46% de la variabilidad total, lo cual nos permite identificar los factores que influyen en el uso y aceptación de las metodologías de desarrollo de software en la Dirección de Desarrollo de Sistemas de Información.

De acuerdo al análisis, en la tabla 6.1.d, se presenta la matriz de factores resultantes, aquí es donde se generan las “variables artificiales” o factores que representan a los constructos.

Constructo	Variable	Carga
Percepción de utilidad en el trabajo y mejora de productividad	V2. Presencia de oportunidad de uso	.698
	V4. Mejora en productividad	.674
	V7. Ventajas vs. Desventajas	.672
	V8. Percepción de utilidad en el trabajo	.774
Facilitar y mejorar el trabajo y mejorar la calidad	V3. Mejora en el trabajo	.577
	V5. Mejora de la calidad del trabajo	.580
	V6. Facilitar el trabajo	.517
Percepción de una metodología clara, que usen personas importantes y sea compatible con la persona	V10. Metodología clara y entendible	.671
	V14. Percepción de pérdida de tiempo	.618
	V16. Influencia de personas importantes para usar la metodología	.665
	V21. Compatibilidad con la manera de desarrollar sistemas por la persona	.635
Estar a la vanguardia en TI, flexibilidad para cambiar de trabajo y seguridad en el trabajo	V37. Estar a la vanguardia en materia de metodologías de desarrollo	.611
	V39. Flexibilidad para cambiar de trabajo	.681
	V41. Seguridad en el trabajo	.523
Establecer claramente los resultados de usar la metodología y transmitir	V42. Trasmitir el resultado del uso de las metodologías de desarrollo	.679
	V44. Establecer claramente los resultados del uso de las metodologías	.686
	V45. Explicar beneficios y desventajas	.637
Compatibilidad personal y en el trabajo, percepción de mayor prestigio y facilidad para aprender la metodología	V9. Facilidad para aprender la metodología	.579
	V23. Compatibilidad de la metodología a la forma como trabajo	.595
	V22. Compatibilidad en los aspectos de mi trabajo	.466
	V24. Percepción de un mayor prestigio de quienes usan la metodología	.415
	V30. Visibilidad del uso de la metodología en mi departamento	.498
Percepción de la metodología como no tediosa y la influencia de las personas a usar la metodología	V13. Metodología no es tediosa	.476
	V15. Influencia de personas a usar la metodología	.305

Tabla 6.1.d Matriz de cargas factoriales

De acuerdo a la estructura presentada en la tabla 6.1.d se obtiene una lista de 7 factores que pueden explicar el modelo aplicado a la Dirección de Desarrollo de Sistemas Información de Área en estudio.

Tabla 6.1.e Factores para la aceptación o rechazo de las metodologías de desarrollo

Factor	Significado
Percepción de utilidad en el trabajo y mejora de productividad	Evalúa que tan presente tienen los desarrolladores el uso de la metodología para su siguiente proyecto, evalúa metodología respecto a si perciben si les va a ser útil y si va a mejorar su productividad. También evalúa la percepción de los desarrolladores sobre las ventajas y desventajas de la metodología
Facilitar y mejorar el trabajo y mejorar la calidad	Es la percepción de los desarrolladores de que la metodología de desarrollo les facilitará, mejorará el trabajo e incrementará la calidad de éste.
Percepción de una metodología clara, que usen personas importantes y sea compatible con la persona	Evalúa la creencia de los desarrolladores de percibir la metodología como una metodología clara, la cual la usan personas que ellos consideran importantes para ellos y que sea compatible con la forma de trabajar de ellos mismos
Estar a la vanguardia en TI, flexibilidad para cambiar de trabajo y seguridad en el trabajo	Evalúa la percepción de los desarrolladores de que al usar la metodología estarán a la vanguardia en su campo de estudio y aparte les proporcionará flexibilidad para cambiar de trabajo así como proporcionar seguridad en el trabajo
Establecer claramente los resultados de usar la metodología y transmitir este conocimiento	Evalúa al desarrollador en su percepción de poder establecer claramente los resultados de usar una metodología y transmitir este conocimiento a los demás desarrolladores.
Compatibilidad personal y en el trabajo, percepción de mayor prestigio y facilidad para aprender la metodología	Evalúa la compatibilidad de la metodología con la forma propia de trabajar y con la manera de trabajar en el área laboral, evalúa la percepción de mayor prestigio de las personas que utilizan la metodología y la facilidad para aprenderla. También evalúa la visibilidad de la metodología en el área de laboral.
Percepción de la metodología como no tediosa y la influencia de las personas a usar la metodología	Evalúa la percepción de las personas hacia la metodología como una actividad no tediosa y la influencia de las personas a usar la metodología.

6.2 Medición de la confiabilidad

La confiabilidad del cuestionario, se evaluó mediante un análisis de consistencia interna calculando el coeficiente de Alfa de Cronbach.

El alfa de Cronbach es un índice de consistencia interna, toma valores entre cero y la unidad, comprueba si el cuestionario recopila información defectuosa y que nos puede llevar a conclusiones erróneas, también nos dice si el cuestionario se trata de un instrumento fiable que proporciona mediciones estables y consistentes. Alfa de Cronbach mide la homogeneidad de las preguntas promediando todas las correlaciones entre todos los ítems para ver que, efectivamente, se parecen. (Ávila, 2010)

Mientras este índice más se acerque a la unidad será mejor es la fiabilidad, considerando una fiabilidad aceptable a partir de 0.70. (Cronbach, 1980)

Para nuestro estudio se observa que los valores de las alfas de Cronbach varían desde 0.702 a 0.858, adicionando a esto los valores de la evaluación de la consistencia interna son superiores a 0.70, el cual es el criterio mínimo aceptable para ésta prueba, excepto en el último factor que no alcanza esta cifra mínima por lo cual dicho factor tendrá que desecharse por completo.

En la tabla 6.1.f se presentan los valores del alfa de Cronbach de los factores identificados para la presente investigación.

Tabla 6.1.f Nivel de confiabilidad del alfa de Cronbach

Factor	Coeficiente alfa de Cronbach	Número de variables
Percepción de utilidad en el trabajo y mejora de productividad	.810	4
Facilitar y mejorar el trabajo y mejorar la calidad	.776	3
Percepción de una metodología clara, que usen personas importantes y sea compatible con la persona	.715	4
Estar a la vanguardia en TI, flexibilidad para cambiar de trabajo y seguridad en el trabajo	.767	3
Establecer claramente los resultados de usar la metodología y transmitir este conocimiento	.858	3
Compatibilidad personal y en el trabajo, percepción de mayor prestigio y facilidad para aprender la metodología	.702	5
Percepción de la metodología como no tediosa y la influencia de las personas a usar la metodología	.260	2

De esta manera se identificaron siete factores dentro del análisis de los resultados, de los cuales en base a las pruebas estadísticas se determina que dos son los más relevantes y uno se tiene que desechar por no pasar la prueba de fiabilidad de Cronbach.

Capítulo 7. Conclusiones

7.1 Conclusiones generales

Este estudio exploratorio se llevo a cabo con la finalidad de determinar los factores que influyen en la aceptación o rechazo del uso de las metodologías de desarrollo de software, tomando como población y posteriormente una muestra de los desarrolladores de sistemas de software que laboran en el INEGI en una Dirección de Área en particular.

Mediante la generalización de modelos de aceptación de herramientas de tecnologías de información, un grupo de investigadores en la Universidad de Arkansas, EUA; desarrollo un cuestionario mediante el cual se captó información de los desarrolladores, misma que se analizo mediante procedimientos matemáticos estadísticos para así obtener los factores que son el objetivo de esta investigación. Conocer la percepción de los desarrolladores al respecto del uso de las metodologías de desarrollo de sistemas de software.

Esta investigación puede servir como cimiento para futuras investigaciones, además de que se puede aplicar en cualquier entidad organizacional dedicada al desarrollo de software. La información que se obtiene de esta herramienta sirve a los administradores de proyectos de software para conocer la postura de los desarrolladores con respecto a la metodología que están utilizando o una nueva metodología que se pretenda implementar, con el fin de tomar mejores decisiones.

La interpretación de los factores encontrados como importantes se describe en seguida:

El factor de “establecer claramente los resultados de usar la metodología y transmitir el resultado de uso de las metodologías de desarrollo” con el Alfa de Cronbach más alto (.858) nos dice que los desarrolladores consideran importante establecer ventajas y desventajas del uso de las metodologías de desarrollo, estableciendo estas situaciones pueden tener un mayor conocimiento para aceptar o rechazar una metodología. De la misma manera, consideran importante el hecho de establecer claramente los resultados del uso de esta metodología, en sí, podemos decir que está ligado a la variable anterior ya que al establecer los resultados se pueden diferenciar aún más las ventajas y/o desventajas del uso de las metodologías; adicionalmente el factor contiene un

componente que nos dice que los desarrolladores están en la mejor disposición de transmitir el resultado del uso de las metodologías de desarrollo, esta última variable proviene del constructo de "demostrabilidad resultante" del modelo de Características de Innovación Percibidas, el cual se refiere al grado en que una innovación es percibida como susceptible de demostración de las ventajas tangibles.

Factor de "percepción de utilidad en el trabajo y mejora de productividad" con el segundo Alfa de Cronbach mas alto (.810), nos sugiere, una alta importancia a la percepción de utilidad en el trabajo, es decir, que a la metodología se le considere útil, que resuelva problemas y no genere más, posteriormente consideran la mejora en productividad, que en otras palabras es hacer mas en menos tiempo, también está involucrado en este factor, que el desarrollador percibe mayores ventajas contra las desventajas; en el factor anterior se encontró con establecer estas ventajas y desventajas, aquí ya se habla de que efectivamente es una variable importante tener más ventajas que desventajas, podemos decir que s complementaria al factor anterior y que de alguna manera estos dos factores son los que tienen más peso y están coincidiendo en estas variables; otra variable involucrada es que el desarrollador siempre tiene presente la oportunidad de usar la metodología para su próximo proyecto de desarrollo de software.

Factor "facilitar y mejorar el trabajo y mejorar la calidad" esta como tercero en importancia, tiene un Alfa de Cronbach (.776), este resalta la importancia en mejorar la calidad del trabajo, facilitar las actividades en éste mismo y en general una mejora. Si ponemos atención, es una extensión del factor anterior en donde se da importancia a la mejora de la productividad que está ligada junto con la mejora del trabajo y la calidad en un constructo de los modelos de psicología social.

Factor "Estar a la vanguardia en TI, flexibilidad para cambiar de trabajo y seguridad en el trabajo" con un Alfa de Cronbach (.767) este factor nos dice que el desarrollador acepta las metodologías de desarrollo de software debido a que le da importancia a estar actualizado en el área de las TI (Tecnologías de Información) para que a su vez esto le permita tener una flexibilidad para cambiar de trabajo, esto es, una persona capacitada tiene mejores oportunidades de cambiar de trabajo. Al mismo tiempo, una persona capacitada adquiere mayor seguridad en el trabajo, es decir, mayores posibilidades conservar su empleo.

Factor de “percepción de una metodología clara, que usen personas importantes y sea compatible con la persona” tiene un Alfa de Cronbach de (.715) nos dice que tiene una importancia el hecho de que la metodología tiene que ser clara y entendible, es decir, que una metodología difícil de comprender o llevar a cabo influye enormemente para que los desarrolladores la acepten o la rechacen, también nos dice que afecta la influencia de personas a quienes los desarrolladores consideran importantes para ellos, estas personas pueden ser líderes de proyectos, personas con un perfil alto, personas con una destreza técnica superior, etc., esto puede ser a cierto respeto o admiración y a las ganas de alcanzar un nivel parecido a ellos. También está ligada la variable de percepción de pérdida de tiempo, es decir, que el desarrollador considera al uso de la metodología como una pérdida de tiempo y no como un apoyo o una actividad adicional para el desarrollo de sus funciones.

El factor “compatibilidad personal y en el trabajo, percepción de mayor prestigio y facilidad para aprender la metodología” contiene un Alfa de Cronbach de (.702) apenas alcanza el valor mínimo como para ser un factor confiable; este factor nos dice que la metodología tiene que ser compatible con la forma como trabajan los desarrolladores de manera muy particular y también tiene que haber una compatibilidad en los aspectos del área laboral. También dice que la metodología debe presentar una facilidad para aprenderla. Los desarrolladores perciben un mayor prestigio de quienes usan la metodología y finalmente le dan una cierta importancia a que sea visible el uso de la metodología dentro del departamento donde trabajan.

El factor “percepción de la metodología como no tediosa y la influencia de las personas a usar la metodología” con un Alfa de Cronbach (.260), un índice muy bajo para ser confiable y por lo tanto este factor y sus variables se desechan.

7.2 Objetivo general

Sobre el objetivo general de identificar los factores que determinan la aceptación o rechazo del uso de las metodologías de desarrollo de software, podemos decir que se cumplió el objetivo al identificar dichos factores de manera satisfactoria. Estos factores se muestran en la tabla siguiente 6.1.f :

Tabla 6.1.f Nivel de confiabilidad del alfa de Cronbach

Factor	Coeficiente alfa de Cronbach	Número de variables
Percepción de utilidad en el trabajo y mejora de productividad	.810	4
Facilitar y mejorar el trabajo y mejorar la calidad	.776	3
Percepción de una metodología clara, que usen personas importantes y sea compatible con la persona	.715	4
Estar a la vanguardia en TI, flexibilidad para cambiar de trabajo y seguridad en el trabajo	.767	3
Establecer claramente los resultados de usar la metodología y transmitir este conocimiento	.858	3
Compatibilidad personal y en el trabajo, percepción de mayor prestigio y facilidad para aprender la metodología	.702	5
Percepción de la metodología como no tediosa y la influencia de las personas a usar la metodología	.260	2

7.3 Objetivos específicos

- Se obtuvo un instrumento de medición de (Riemenschneider et al., 2002) para determinar los factores de aceptación y rechazo al uso de metodologías de desarrollo de software (ver anexo).
- Se probó el instrumento de medición para verificar su funcionalidad y fiabilidad de los resultados mediante métodos estadísticos de Kaiser-Meyer-Olkin (KMO) con un factor de .743 y una prueba de esfericidad de Barlett de $p<.0001$.
- Se aplicó el instrumento de medición en áreas de desarrollo de la Dirección de Desarrollo de Sistemas de Información. En primera instancia, el cuestionario se envió por correo electrónico para ser autoadministrado, sin embargo por complicaciones se recurrió a cuestionario escrito y la administración fue desarrollador por desarrollador.

7.4 Limitaciones de estudio

El presente estudio se baso en la investigación hecha por (Riemenschneider et al., 2002), en la cual se desarrolla una metodología para implementarse en diferentes compañías para posteriormente llevar a cabo la obtención de los factores mediante el instrumento de medición mismo que se utilizó en esta tesis. En esta tesis no se pudo por cuestiones de tiempo y recursos llevar de la misma manera la investigación y se utilizó el instrumento de medición en un área de desarrollo en la cual se tenía conocimiento de que aplicaban una metodología de desarrollo propia. Sin embargo, al aplicar el cuestionario se encontró con la situación de que no todos aplicaban la metodología de desarrollo o bien aplicaban otra metodología distinta a la institucional. A pesar de esto, la muestra al no ser probabilística y dirigida, se estuvieron seleccionando personas que al menos conocían la diferencia entre usar y no las metodologías de desarrollo para tener una muestra un tanto más uniforme.

A pesar de esto, la muestra al no ser por completo uniforme, los resultados de las pruebas de Kaiser-Meyer-Olkin y el análisis de consistencia interna para evaluar la confiabilidad de los factores con el coeficiente de alfa de Cronbach salieron muy bajos. Aún así dieron los valores mínimos para poder continuar con el estudio.

La sugerencia es que se debe uniformizar al máximo la muestra para obtener resultados más precisos y confiables, antes de iniciar una investigación parecida se debe asegurar que el área laboral donde se aplicará el instrumento de medición cumple este requisito para un mejor resultado.

7.5 Recomendaciones

A grandes rasgos hay diversas variables que podemos agrupar fuera de los factores resultantes debido a que son similares o incluso, pertenecen a un mismo constructo de los modelos de intención de comportamiento, el agrupamiento estaría de la siguiente manera:

- Las variables relacionadas a la productividad, la utilidad y calidad en trabajo; así como también el facilitar y mejora en forma general del trabajo.

- Esta muy presentes las variables de compatibilidad en la forma como trabaja cada desarrollador, la compatibilidad en como el desarrollador elabora sistemas de software y la compatibilidad de la metodología con la forma de trabajar del área.
- Otras variables que aparecen constantemente son las de conocer perfectamente las ventajas y desventajas de la metodología, establecer los resultados y que la metodología sea clara y fácil de aprender.
- Otras variables que también arroja el instrumento de medición son la flexibilidad que ofrece a los desarrolladores para cambiar de trabajo, al mismo tiempo la seguridad en éste.
- En otro grupo tenemos que el prestigio es un punto importante de usar la metodología, la norma subjetiva y la visibilidad en el trabajo.

Sse deben buscar satisfacer o reafirmar estos puntos en la metodología, si bien para mejorar la ya existente o bien elaborar o adoptar una nueva metodología

7.6 Trabajos futuros

En este trabajo de investigación la población y posteriormente la muestra objetivo, se limitó a un área organizacional muy pequeña dentro del instituto, lo conveniente sería extender la población para poder tener la certeza de que las áreas de desarrollo dentro de todo el instituto se comportar de la misma manera. Mediante este conocimiento se pueden aplicar las medidas necesarias para implementar y evitar el rechazo en la adopción de las metodologías de desarrollo de software acorde a los factores que se descubrieron como significativos.

Además de elaborar metodologías propias del instituto, deben considerarse los factores identificados en este estudio, para fomentar su adopción y posterior aplicación.

Anexos

Cuestionario

Dirección / Dpto:	
Tipo de contrato:	
Edad:	
Carrera:	
Rol principal:	(analista, desarrollador, tester, MBDA, etc)
Antigüedad en este rol (años):	
Usas una metodología de desarrollo (si/no):	
¿Cuál?	

INTRODUCCIÓN

El presente estudio tiene por objetivo tratar de explicar los factores que influyen en los desarrolladores de sistemas de software para aceptar o rechazar el uso de una metodología de desarrollo de software. Se trata de un estudio exploratorio, es decir, no es concluyente, sino su objetivo es documentar ciertas experiencias. Este estudio pretende generar datos e hipótesis que constituyan la materia prima para investigaciones más precisas.

Muchas organizaciones intentan implementar metodologías destinadas a mejorar los procesos de desarrollo de software. Sin embargo la resistencia de los desarrolladores hacia dichas metodologías a menudo obstruye su correcta implementación. A pesar de que las inversiones en desarrollo de software que hacen las empresas se ha incrementado enormemente, evidencias sugieren que el desarrollo de software no está mejorando como debiera, la calidad se está mejorando a un ritmo menor de lo que se esperaba, la productividad de hecho se ha declinado en un 13 por ciento en años recientes, y hasta un 75 por ciento de todos los esfuerzos de desarrollo son considerados como fracasos. Mejorar los procesos de desarrollo de software ha sido y sigue siendo una de las principales preocupaciones de la administración de las Tecnologías de Información.

Este instrumento de medición tiene la finalidad de identificar las determinantes de las intenciones de los desarrolladores de software para usar una nueva metodología de

desarrollo de software introducida en una determinada área. Esta información servirá como cimiento para establecer mejoras a las metodologías que se usan o en la adopción de nuevas metodologías, proporcionando mayor conocimiento a los administradores de proyectos y de esta manera puedan tomar mejores decisiones al respecto.

Instrucciones de llenado: Lee los siguientes enunciados y marca con una X en las casillas que le siguen según qué tan de acuerdo o en desacuerdo te encuentres con ese enunciado, solo puedes tachar una de las siete opciones. **MDS** hace referencia a la Metodología de Desarrollo de Software que se usa principalmente en la dirección de área a la cual perteneces, o puede ser la metodología que usas, hayas usado o que conozcas y sea similar a ésta.

1. Tengo la firme intención de utilizar la metodología MDS en mi trabajo para mi próximo proyecto

Totalmente de acuerdo	Fuertemente De acuerdo	De acuerdo	Neutral	En desacuerdo	Fuertemente en desacuerdo	Totalmente en desacuerdo
☐	☐	☐	☐	☐	☐	☐

2. Si se presenta la oportunidad, usaría la metodología MDS
3. Si aplico la metodología MDS me permitirá mejorar mi trabajo
4. Al usar la metodología MDS se incrementa mi productividad (hacer mas en el mismo tiempo)
5. El usar la metodología MDS realza la calidad de mi trabajo
6. Si uso la metodología MDS mi trabajo será más fácil
7. Las ventajas de usar la metodología MDS superan a las desventajas
8. La metodología MDS es útil en mi trabajo
9. Aprender la metodología MDS fue fácil para mi
10. Creo que la metodología MDS fue clara y entendible
11. Usar la metodología MDS no requiere de mucho esfuerzo mental
12. Encuentro a la metodología MDS fácil de usar
13. La metodología MDS no es tediosa
14. Usar la metodología MDS no quita mucho tiempo de las actividades diarias

15. Las personas que influyen en mi comportamiento piensan que debería de usar la metodología MDS
16. Las personas que considero importantes para mí piensan que debería usar la metodología MDS
17. Los compañeros de trabajo piensan que debería de usar la metodología MDS
18. Aunque podría ser útil, usar la metodología MDS no es obligatorio en mi trabajo
19. Mi jefe no me pide que use la metodología MDS
20. Para mí el uso de la metodología MDS es voluntario
21. La metodología MDS es compatible con la manera en que desarrollo sistemas
22. El uso de la metodología MDS es compatible en todos los aspectos de mi trabajo
23. Usar la metodología MDS encaja bien a la forma como trabajo
24. La gente en el INEGI que usa la metodología MDS tiene más prestigio que aquellas que no la usan
25. La gente en el INEGI que usa la metodología MDS tiene un perfil alto
26. Usar la metodología MDS es un símbolo de estatus en el INEGI
27. La metodología MDS tiene mucha presencia en el INEGI
28. Es fácil observar a otros usando la metodología MDS
29. He tenido muchas oportunidades de ver la metodología MDS siendo utilizada
30. Puedo ver cuando otros usan la metodología MDS en mi departamento
31. Creo que no hay diferencia entre mis habilidades y conocimientos existentes y las requeridas por la metodología MDS
32. Tengo el conocimiento necesario para usar la metodología MDS
33. Tengo disponible capacitación y educación especializada de la metodología MDS
34. Tengo disponible orientación formal para usar la metodología MDS
35. Tenemos un grupo específico para asistencia con las dificultades de la metodología MDS
36. Los directivos proveen toda la ayuda y recursos necesarios para permitir a la gente usar la metodología MDS
37. Saber la metodología MDS me pone a la vanguardia en mi campo de estudio
38. El conocimiento de la metodología MDS incrementa mi oportunidad de promoción
39. Conocer la metodología MDS incrementa mi flexibilidad para cambiar de trabajo
40. El conocimiento de la metodología MDS incrementa la oportunidad de tener un trabajo más significativo

41. Conocer la metodología MDS puede incrementar la oportunidad de ganar seguridad en el trabajo
42. No tendría dificultad en decirle a otros el resultado de usar la metodología MDS
43. Creo que podría comunicarle a otros las consecuencias de usar la metodología MDS
44. Los resultados de usar la metodología MDS son evidentes para mí
45. No tendría dificultad explicando porque la metodología MDS puede o no ser beneficiosa.

Bibliografía

Abud, M. A. (2005). Calidad en la Industria del Software: La Norma ISO 9126.

Ajzen, I. (1988). Attitudes, Personality and Behavior.

Ajzen, I., & Fishbein, M. (1980). *Understanding Attitudes and Predicting Social Behavior*. Prentice Hall.

Ávila, R. (2010). *Factores que determinan el uso y aceptación de tecnologías de información en el INEGI*.

Barranco, J. (2001). *Metodología del análisis estructurado de sistemas* (2º ed.). Universidad Pontífica.

Brewer, J. L., Dittman, K., & Ghatge, G. (2005). 1 A Study of Software Methodology Analysis:“Great Taste or Less Filling”. *ISECON*, *22*, 1-13.

Chau, P. (1996). An Empirical Investigation on Factors Affecting the Acceptance of CASE by System Developers. *Information & Management*, *30*, 269 - 280.

Chau, P. Y. K., & HU, P. J. (2001). Information technology acceptance by individual professionals: A model comparison approach. *BNet*, Business Publications.

Conference, I. R. M. A. I., & Khosrowpour, M. (1990). *Managing information resources in the 1990s (en linea)*. Idea Group Inc (IGI).

Davis, F. (1989). 2 Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MisQuartely*.

Falgueras, B. C. (2003). *Ingeniería del software*. Editorial UOC.

Fred, D. D., Richard, P. B., & Paul, R. W. (1989). 3 User Acceptance of Computer

Technology: A Comparison of Two Theoretical Models. *Management Science*, *35*(8).

Gacitúa, R. A. (2003). Métodos de Desarrollo de Software: El Desafío Pendiente de la Estandarización. *Theoria*, *12*, 23-42.

Gracia, T. I. (2004). *Introducción a la psicología social*. Editorial UOC.

Hernández, R., Fernández, C., & Baptista, P. (1991). *Metodología de la Investigación*. McGraw-Hill.

Jacobson, I. (1992). *Object-oriented software engineering*. ACM Press.

Johnson, R., Hardgrave, B., & Doke, E. (1999). An Industry Analysis of Developers Beliefs About Object-Oriented System Development.

Khalifa, M., & Verner, J. (2000). 9 Drivers for Software Development Method Usage.

Moore, G., & Benbasat, I. (1991). Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation. *Information System Research*, *2*, 192 - 222.

Pfleeger, S. (1999). Understanding and Improving Technology Transfer in Software Engineering.

Pressman, R. S. (1998). *Ingeniería del software*. McGraw-Hill.

Riemenschneider, C., Hardgrave, B., & Davis, F. (2002). 1 Explaining software developer acceptance of methodologies: a comparison of five theoretical models. *Software Engineering, IEEE Transactions on*, 0098-5589, *28*(12), 1135 -1145.

Roberts, T., Gibson, M., & Fields, K. (1998). 8 Factors That Impact Implementing a System Development Methodology.

Roberts, T., Gibson, M., & Fields, K. (1999). System Development Methodology Implementation: Perceived Aspects of Importance.

Santana Tapia, R. G. (2003, February). *CALIDAD EN EL DESARROLLO DE SOFTWARE*

EN EL GOBIERNO DEL ESTADO DE TAMAULIPAS Y ALTERNATIVAS DE MEJORA (Técnico).

Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.

Standish Group. (1994). *Chaos Report 1994* (Reporte Anual). Standish Group.

Standish Group. (2009). *Chaos Report 2009* (Reporte Anual). Standish Group.

Taylor, S., & Todd, P. (1995). Understandig Information Technology Usage: A Test of Competing Models. *Information System Research*, *6*, 144-176.

Thompson, R., Higgins, C., & Howell, J. (1991). Personal Computing: Toward a Conceptual Model of Utilization. *MisQuartely*, *15*, 125 - 143.

Torres, C. L., & Alférez, G. (2008). Establecimiento de una Metodología de Desarrollo de Software
para la Universidad de Navojoa Usando OpenUP. COMP-004-2008.

Triandis, H. (1979). Values, Attitudes, and Interpersonal Behavior.

Tsui, F. F., & Karam, O. (2006). *Essentials of Software Engineering*. Jones & Bartlett Publishers.

Venkatesh, V. (2000). Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model.

Venkatesh, V., & Davis, F. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, *46*.

Venkatesh, V., Smith, R., Morris, M., Davis, G., & Davis, F. (2003). 4 User Acceptance of Information Technology: Toward a Unified View. *MisQuartely*, *27*(3), 425-478.

Verner, J., & Cerpa, N. (1997). Prototiping: Does your view of its advantages depend on your job?

Weitzenfeld, A. (2004). *Ingeniería de software orientada a objetos con UML, Java e*

Internet. Cengage Learning Editores.

Yourdon, E. (2006). *Just Enough Structured Analysis*. Prentice Hall.

Cronbach (1980) “Coefficient Alpha and the Internal Structure of Tests”