

Gestión de Proyectos de Software



AUTORES

Francisco Javier Álvarez
Julio Ariel Hurtado Alegría
Margarita Mondragón Arellano
Jaime Muñoz Arteaga
César Eduardo Velázquez Amador
Yosly Caridad Hernández Bieliukas

Gestión de Proyectos de Software

1a ed. - Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014. 79 pag.

Primera Edición: Marzo 2014

Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn)

<http://www.proyectolatin.org/>



Los textos de este libro se distribuyen bajo una licencia Reconocimiento-CompartirIgual 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/deed.es_ES

Esta licencia permite:

Compartir: copiar y redistribuir el material en cualquier medio o formato.

Adaptar: remezclar, transformar y crear a partir del material para cualquier finalidad.

Siempre que se cumplan las siguientes condiciones:



Reconocimiento. Debe reconocer adecuadamente la autoría, proporcionar un enlace a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciador o lo recibe por el uso que hace.



CompartirIgual — Si remezcla, transforma o crea a partir del material, deberá difundir sus contribuciones bajo la misma licencia que el original.

Las figuras e ilustraciones que aparecen en el libro son de autoría de los respectivos autores. De aquellas figuras o ilustraciones que no son realizadas por los autores, se coloca la referencia respectiva.



Este texto forma parte de la Iniciativa Latinoamericana de Libros de Texto abiertos (LATIn), proyecto financiado por la Unión Europea en el marco de su Programa ALFA III EuropeAid. El Proyecto LATIn está conformado por: Escuela Superior Politécnica del Litoral, Ecuador (ESPOL); Universidad Autónoma de Aguascalientes, México (UAA), Universidad Católica de San Pablo, Perú (UCSP); Universidad Presbiteriana Mackenzie, Brasil(UPM); Universidad de la República, Uruguay (UdelaR); Universidad Nacional de Rosario, Argentina(UR); Universidad Central de Venezuela, Venezuela (UCV), Universidad Austral de Chile, Chile (UACH), Universidad del Cauca, Colombia (UNICAUCA), Katholieke Universiteit Leuven, Bélgica (KUL), Universidad de Alcalá, España (UAH), Université Paul Sabatier, Francia (UPS).

Índice general

Prólogo	7
Prefacio	9
Introducción	11
1 Gestión del Alcance y de los Requerimientos de un Proyecto.	13
2 Gestión de personal y liderazgo	17
3 Gestión de la calidad de un proyecto	35
4 Gestión de Riesgos en proyectos de software	41
5 Procesos de desarrollo de software	51
Bibliografía	71
Resumen de Currículos de los docentes	77

Prólogo

Durante el proceso de desarrollo de software se busca lograr un balance efectivo entre la productividad, la calidad, la armonía organizacional y por tanto la competitividad. La industria de software requiere de la capacidad ingenieril para desarrollar sus proyectos, pero el éxito va más allá de la capacidad técnica, y una de esas relevantes capacidades es la de poder administrar adecuadamente sus proyectos.

Este libro busca brindar los fundamentos y las herramientas conceptuales y técnicas para formar ingenieros de software orientados hacia la gestión práctica de procesos, con el ánimo de fortalecer la formación de los estudiantes de computación, de forma acorde a las necesidades de la creciente industria de software en Latinoamérica, quienes se desempeñarán como líderes de proyectos, de calidad, consultores, analistas de procesos, así como la investigación, innovación y desarrollo de nuevos modelos de gestión de proyectos de software.

Derivado de la necesidad de definir un perfil específico para la Ingeniería de software acorde a los requerimientos de la Industria latinoamericana de Tecnologías e Información y Comunicación. La academia podrá utilizar el presente como referente para la creación de programas y/o adaptación de su plan de estudios; a la Industria como referente para la selección y contratación de personal competente para cubrir sus necesidades de Ingeniería de Software; y para el aspirante como referente de los requisitos necesarios para obtener el conocimiento como “Ingeniero de Software” en el área de procesos y gestión de proyectos.

Prefacio

- La principal aportación del libro será unificar contenidos para la Industria de Software latinoamericana considerando las diferentes capacidades y experiencias de los académicos de las institucionales participantes.
- Los niveles académicos que pretender abordar el libro son la diferentes Licenciaturas e Ingenierías de tengas tópicos en Ingeniería de Software así como Posgrados relacionados con la temática.
- Aproximadamente se podría tener un impacto de al menos 200 alumnos de las instituciones participantes a nivel de licenciatura y de 40 alumnos a nivel de posgrado.

El objetivo del libro es brindar elementos conceptuales y técnicos para definir y modelar procesos de software de tal forma que sean utilizados por los distintos roles dentro de la ingeniería de software de la organización y que su representación, adaptación y análisis por parte del equipo de procesos sea apoyado a través de un procesamiento computacional.

Introducción

El libro de Interacción Gestión de Proyectos de Software tiene la siguiente estructura:

1. GESTIÓN DEL ALCANCE Y DE LOS REQUERIMIENTOS DE UN PROYECTO.
2. GESTIÓN DE PERSONAL Y LIDERAZGO.
3. GESTIÓN DE LA CALIDAD DE UN PROYECTO.
4. GESTIÓN DE RIESGOS.
5. PROCESOS DE DESARROLLO DE SOFTWARE.

Cada capítulo incluye un objetivo, contenidos y actividades de aprendizaje.

1 — Gestión del Alcance y de los Requerimientos de un Proyecto.

Margarita Mondragón Arellano

Margarita Mondragón Arellano (mmondrago@correo.uaa.mx), Departamento de Sistemas de Información de la Universidad Autónoma de Aguascalientes (UAA)

OBJETIVO

Comprender en base al proceso propuesto por el PMBOK los conceptos básicos y las actividades generales que involucra la Gestión del Alcance y de los Requerimientos de un Proyecto. Aplicar los fundamentos teóricos mediante un caso de estudio.

RESUMEN DEL CAPÍTULO

La gestión ha comenzado a verse como un elemento que ayuda a ejecutar las acciones necesarias para ordenar, disponer y organizar los recursos de un proyecto, utilizando procedimientos específicos y optimizando la relación entre los recursos y los resultados.

Si no se cuenta con una buena definición del alcance del proyecto desde el inicio entonces su gestión se tornará deficiente generando con ello pérdidas tanto de tiempo como de dinero. Un alcance que no es esclarecido puede dar lugar a malas interpretaciones, por ejemplo el cliente puede considerar agregar un nuevo elemento dentro del proyecto, hecho que afectará el alcance.

CONOCIMIENTOS PREVIOS

Participación en el Ciclo de Desarrollo de Proyectos de Tecnologías de Información; así como el conocimiento teórico-básico del mismo.

INTRODUCCIÓN

La Gestión del Alcance del Proyecto incluye los procesos necesarios para asegurarse que el proyecto incluya todo el trabajo requerido, y sólo el trabajo requerido, para completar el proyecto satisfactoriamente. La gestión del alcance del proyecto se relaciona principalmente con la definición y el control de lo que está y no está incluido en el proyecto.

El alcance de un proyecto es la suma total de todos los productos y sus requisitos o características. Se utiliza a veces para representar la totalidad de trabajo necesario para dar por terminado un proyecto.

PROCESOS DE GESTIÓN DEL ALCANCE DEL PROYECTO

1. Planificación del Alcance.
2. Definición del Alcance.
3. Crear EDT.
4. Verificación del Alcance.
5. Control del Alcance.

Estos procesos interaccionan entre sí, cada uno puede involucrar el esfuerzo de una o más personas o grupos de personas, sobre la base de las necesidades del proyecto. Cada proceso tiene lugar por lo menos una vez y se produce en una o más fases del proyecto.

FUNDAMENTOS TEÓRICOS

CONCEPTOS BASE DE LA GESTIÓN DEL ALCANCE Y DE LOS REQUERIMIENTOS DEL PROYECTO.

- **Proyecto.** *Esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.* Cada proyecto tiene un inicio y un fin. El final se alcanza cuando:
 - Se han logrado los objetivos del proyecto.
 - Se determina que no se va a poder terminar.
 - La necesidad del proyecto ya no existe y por lo mismo se cancela.
- **Administración de Proyectos.** Aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requerimientos del mismo.
- **Alcance.**
- **Alcance del Proyecto.** Trabajo que debe realizarse para entregar un producto, servicio o resultado con las funciones y características especificadas.
- **Alcance del Producto.** Características y Funciones que caracterizan a un producto, servicio o resultado.
- **Administración del Alcance.** Es la encargada de describir los procesos y actividades que forman parte de los diversos elementos de la dirección de proyectos, que se identifican, definen, combinan, unen y coordinan dentro de los Grupos de Procesos de Dirección de Proyectos.

PLAN DE LA GESTIÓN DEL ALCANCE.

Esta actividad se encarga de la creación de un plan para la gestión del alcance que documente, cómo será definido el alcance del proyecto además de validado y controlado.

La principal ventaja de contar con un plan es que se proporciona orientación y dirección en la forma en que se gestionará el alcance en todo el proyecto.

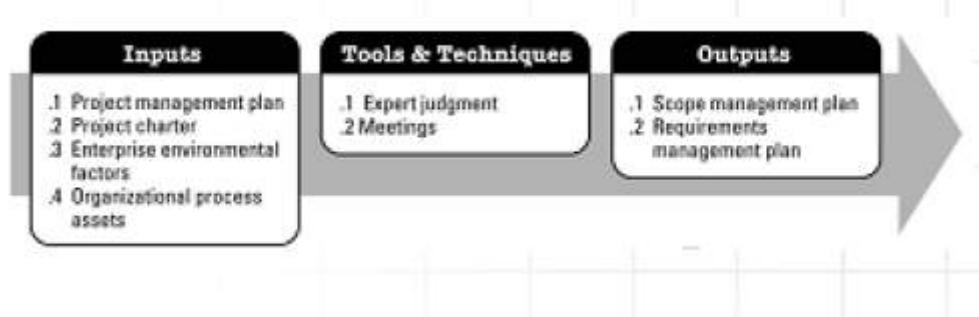


Figura 1.1: Plan de la Gestión del Alcance

RECOPILACIÓN DE REQUERIMIENTOS.

Esta actividad se encarga de la creación de un plan para la gestión del alcance que documente, cómo será definido el.

Ahora bien, para realizar el diseño de un AMS en términos de patrones es necesario primero recopilar en la etapa de análisis la información y facilitar la interpretación de dicha información, sobre esta base se plantea “el qué hará”, mientras que el diseño plantea “el cómo lo hará”.

DEFINICIÓN DEL ALCANCE.

Esta actividad

CREACIÓN DEL EDT (WBS).

Esta actividad

VALIDACION DEL ALCANCE.

Esta actividad

CONTROL DEL ALCANCE.

Esta actividad

CASO PRÁCTICO

Con el fin de aplicar la metodología propuesta de la sección anterior, aquí se presenta el análisis y diseño de un ambiente simulador de señales. El simulador permite al usuario generar señales y visualizar diferentes tipos de filtros así como diferentes tipos de tratamiento para procesar las señales. En la interfaz de usuario del simulador (ver figura 2)

EJERCICIOS RESUELTOS

Por favor describa trabajos relacionados al tema interfaz de usuario y la funcionalidad de un AMS utilizando respectivamente los patrones de interacción y los patrones de diseño de software.

EJERCICIOS A RESOLVER

La autoevaluación sobre el diseño de ambientes de modelado y simulación se presenta a través de diversos tipos de encuestas, estrategias de análisis y de discusión, a saber.

1. ¿Cuál es el objetivo principal de un ambiente de modelado y simulación?
2. ¿Cómo un ambiente de modelado y simulación o hace más fácil la tarea el usuario?
3. ¿En qué beneficia el paradigma de patrones en el diseño de ambientes de simulación?

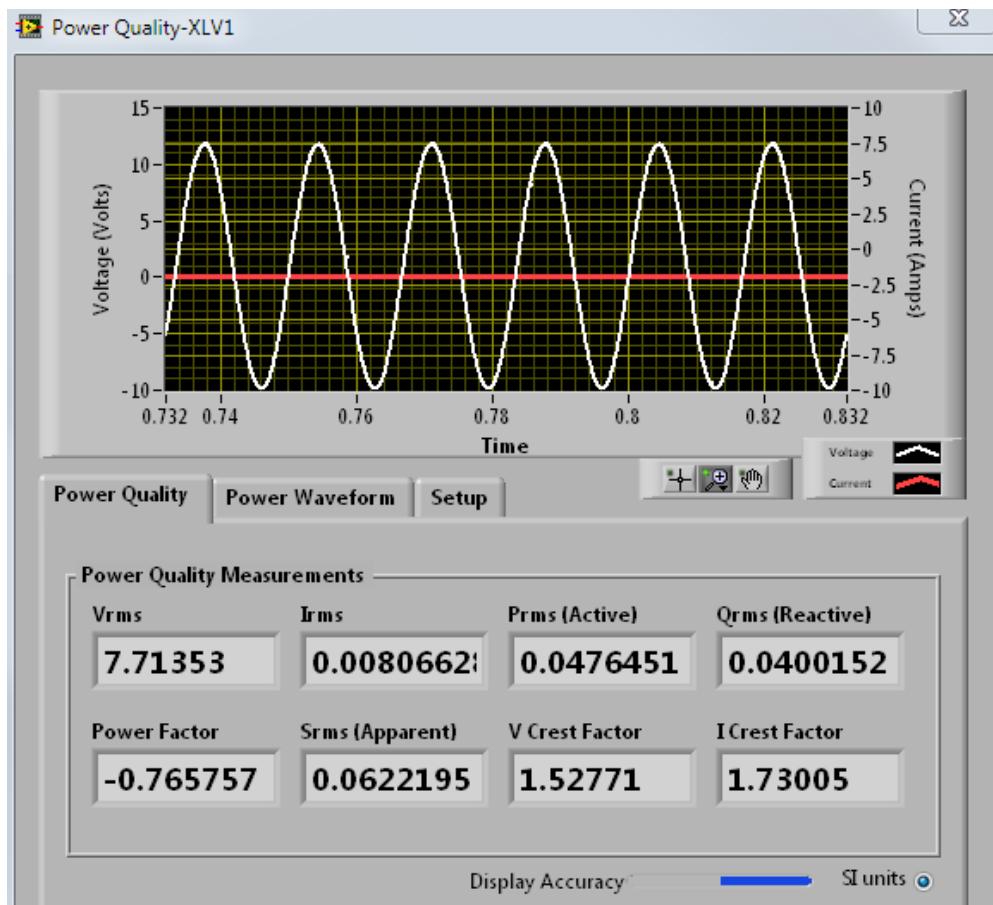


Figura 1.2: Interfaz de usuario del ambiente de simulación de señales

BIBLIOGRAFIA

Project Management Institute (PMI): A Guide to the Project Management Body of Knowledge (PMBOK Guide). Newtown Square: Project Management Institute, Inc, Fifth Edition (2013).

Perrin, Richard: PMP EXAM PREP BOOT CAMP (Based on the PMBOK Guide, 5th Ed.) : EDWel Programs, (2013).

2—Gestión de personal y liderazgo

Jaime Muñoz Arteaga¹, Francisco Javier Álvarez Rodríguez²

¹ *Jaime Muñoz Arteaga (jmunoz@correo.uaa.mx), Departamento. de Sistemas de Información de la Universidad Autónoma de Aguascalientes (UAA)*

² *Francisco Alvarez Rdz. (fjalvar@correo.uaa), Departamento. de Ciencias de la Computación Sistemas de la Universidad Autónoma de Aguascalientes (UAA)*

OBJETIVO

Comprender en base del proceso para manejo de personal y liderazgo como uno de los elementos básicos para la administración de proyectos de software.

CONOCIMIENTOS PREVIOS

Participación en el Ciclo de Desarrollo de Proyectos de Tecnologías de Información; así como el conocimiento teórico-básico del mismo.

Resumen:

En este trabajo se reseña brevemente la evolución de la gestión de los recursos humanos, su historia, definiciones y objetivos. Se valora su importancia en el ámbito de las organizaciones actuales, se mencionan los rasgos que caracterizan la concepción moderna de la gestión de recursos humanos. Se hace referencia a varios conceptos de estrategias de Recursos Humanos, así como se detallan diversos modelos de la Gestión del Capital Humano que en los últimos tiempos se han desarrollado, los cuales tienen el fin común de lograr la competitividad.

Evolución histórica de la Dirección de Recursos Humanos (DRH). De la dirección de personal a la DRH

El factor humano siempre ha estado presente en los estudios llevados a cabo dentro de las distintas escuelas o enfoques dentro de la teoría de la administración, si bien su tratamiento dentro de la empresa ha sufrido una fuerte evolución, al igual que ha sucedido con el proceso de administración de la empresa que ha evolucionado hasta la dirección estratégica, la dirección de recursos humanos ha pasado por una serie de etapas hasta culminar en su actual concepción.

1. Dirección de personal clásica: como una primera etapa dentro de este proceso de evolución se encuentra la dirección de personal clásica, la cual partía de la idea de que el factor humano es un factor productivo más, con motivaciones esencialmente económicas y generalmente pasivo en cuanto a su participación en las decisiones empresariales, limitándose a ejecutar su trabajo, con una menor o mayor eficiencia según su nivel de remuneración y formación.

La dirección de personal realizaba más bien una tarea administrativa, consistente en la confección y pago de las nóminas, formalización de la contratación que el empleo de alguien implicaba, así como estadísticas de abastecimiento y confección de plantillas. Esta concepción de la dirección de personal hacía que su importancia en relación con otras áreas funcionales fuera menor y por consiguiente contara con un directivo de menor grado.

2. Moderna dirección de personal: el papel del factor humano se tornó más activo y se fue tomando conciencia de su importancia, la dirección de personal fue incorporando más funciones y asumiendo más decisiones. Las políticas de empleo, remuneración, formación y promoción, además de las tareas antes mencionadas.

3. Dirección de recursos humanos: en la actualidad va más allá de la moderna dirección de personal, el factor humano tiene un gran peso en la formulación e implementación de estrategias, dada su capacidad de ayudar a fortalecer la competitividad y el desarrollo de las empresas en un entorno turbulento. En este sentido debe asegurar:

- La inserción del personal en la empresa
- La dinamización y la movilización del personal de modo tal que se utilicen sus capacidades para el logro de los objetivos de la empresa
- El progreso del personal, donde aportan los conocimientos que serán necesarios a los miembros de la empresa para que actúen al mayor nivel de competencia posible
- La comunicación, o sea el diálogo fluido permanente y consecutivo
- La calidad de vida, seguridad y compensaciones compatibles con los resultados de la empresa
- La imagen de la empresa y por tanto la imagen del factor humano

El factor humano se ha visto de diferentes maneras con el transcurso del tiempo. En la primera etapa el recurso humano es considerado como un coste que hay que minimizar. En la segunda etapa, la DRH comienza a tomar en cuenta las necesidades sociales y psicológicas de las personas, persiguiendo la adaptación del personal a la organización.

Desde esta perspectiva la relación organización-empleado va más allá de los aspectos retributivos. Sin embargo el recurso humano sigue considerándose como un coste que hay que minimizar, pero las acciones que se inician son de carácter proactivo.

En la última etapa, el factor humano es considerado como un elemento determinante en el desarrollo de la empresa, la motivación y la eficiencia de éste están en función de la manera en que es empleado el personal de la organización. En esta fase las acciones que se toman no sólo son proactivas, la atención a las personas y su desarrollo es considerado un factor clave en el cumplimiento de la estrategia empresarial, iniciándose así la concepción estratégica del capital humano.

El Capital Humano como herramienta estratégica de la empresa

Las nuevas realidades económicas están presionando a la dirección del capital humano para que amplíen su punto de mira y abarquen no solo su tradicional papel administrativo, sino también un papel estratégico. Puesto que la principal fuente de producción de la economía ha pasado del capital físico al capital intelectual, los directivos del capital humano se ven obligados a demostrar cómo crean valor para sus organizaciones. El problema del capital humano cuyo impacto en la estrategia de una empresa es apenas visible, es la calidad que constituye una

materia prima de potencial competitivo sostenido.

Es por ello que los directivos del capital humano deben entender la estrategia de la empresa, es decir su plan para desarrollar y sostener una ventaja en el mercado. Después deberán comprender las implicaciones de esa estrategia para el capital humano. Ser un departamento estratégico requiere que los profesionales del capital humano sepan exactamente qué capacidades son necesarias para la implementación favorable de la estrategia de sus empresas y cómo el capital humano influye en dichas capacidades.

La importancia que se va otorgando al Capital humano en el enfoque estratégico responde a una doble necesidad:

- Una necesidad planteada desde dentro de la empresa y que se puede considerar resultado de un mayor nivel de formación de las personas y por tanto un mayor nivel de exigencias en busca de la satisfacción del trabajo, calidad de vida, así como la participación en la toma de decisiones
- Una necesidad creada por la evolución del entorno cada vez más competitivo y cambiante al que tienen que enfrentarse las empresas

Por tanto, el capital humano es imprescindible para el éxito organizacional, ya que es un factor que se puede acomodar a las necesidades de cualquier empresa y se encuentra inmerso en la implementación estratégica de la misma, que es la clave para conseguir una ventaja competitiva sostenida.

La Gestión de los RRHH (GRH). Objetivos estratégicos

Happer y Lynch, (1992) enfocan la gestión de los recursos humanos como una serie de acciones encaminadas a maximizar el potencial humano tanto a nivel individual como colectivo en pro de llevar a cabo un adecuado desempeño y por consiguiente lograr los objetivos propuestos.

La concepción moderna de la gestión de recursos humanos se caracteriza por una serie de rasgos:

- En el siglo XXI los Recursos Humanos constituyen el recurso más importante de la empresa
- Los recursos humanos son inversiones no costes
- La gestión de recursos humanos se hace como función integral de toda la organización, además es proactiva
- El aumento de la productividad del trabajo y la satisfacción del trabajo son objetivos inmediatos de la gestión de recursos humanos
- El desafío fundamental de la gestión de recursos humanos es lograr eficiencia y eficacia en las organizaciones, lo que permite asignarle a éste la importancia que se merece

Para llevar adelante la gestión integrada de Recursos Humanos se necesita:

- Obtener, formar, motivar, retribuir, y desarrollar el número de personas que la organización requiere para lograr sus objetivos
- Diseñar e implementar la estructura, sistemas y mecanismos organizativos que combinen los esfuerzos de las personas y sus recursos para que los objetivos se consigan de la forma más eficaz posible
- Ver el factor humano como el motor impulsor del éxito organizacional, por lo que se debe actuar con antelación para obtener cambios en el comportamiento de estas personas como un elemento básico en la Dirección Estratégica de la empresa.

La integración de la Gestión de los Recursos Humanos a la Dirección Estratégica ha ayudado al cambio del enfoque para incorporar nuevos principios:

- Orientación a su entorno y su evolución
- Identificación de los cambios internos necesarios para adaptarse a la evolución del entorno
- Planificación de las acciones con visión a largo plazo
- Rediseño de la organización en función de la estrategia
- Énfasis en los procesos de cambio
- Integración de todas las áreas en el marco de los objetivos estratégicos

Objetivos Estratégicos en la Gestión de Recursos Humanos

Los objetivos son el establecimiento de prioridades a todos los niveles de la empresa dentro de un procedimiento organizado jerárquicamente, en el que cada objetivo está formulado por referencia a los objetivos superiores en el seno de la empresa.

Objetivo No1: Lograr la equidad e integración del personal. Está orientado hacia la empresa, a través de sistemas formales para establecer justicia, normas, valoraciones, comités de consulta, negociación colectiva y mejora en la calidad de vida laboral.

Objetivo No2: Elevar el rendimiento competitivo y operativo. Está orientado hacia el mercado, hacia el exterior, la respuesta de la gestión de recursos humanos está encaminada a la dirección por objetivos, remuneración por resultados, planificación y desarrollo de Recursos humanos.

Objetivo No3: Innovación y flexibilidad (Eficiencia a largo plazo). Está orientado hacia el interior y el exterior de la empresa, la respuesta de la gestión de recursos humanos está en las coexistencias de estructuras formales e informales, comunicaciones abiertas y gestión participativa, así como la seguridad psicológica, garantía de empleo dirigida a la formación, movilidad, participación en beneficios, información económica y estrategias, además de las oportunidades del futuro que favorezcan el cambio.

La Planificación Estratégica del Capital Humano

En las dos últimas décadas se ha observado un creciente interés de los profesionales del capital humano en involucrarse en la planificación estratégica de las organizaciones, al mismo tiempo expresan las expectativas de un profesional proactivo y dinámico, lo cual es un factor de éxito indispensable en todo ambiente competitivo. Los autores definen la Planificación Estratégica del Capital Humano como: el proceso de análisis de las necesidades de capital humano, conforme cambian los entornos internos y externos de la organización y la aplicación de la consiguiente estrategia proactiva para asegurar la disponibilidad del capital humano demandada por la organización.

Stoner, (1996) señala cinco atributos respecto a la Planeación Estratégica. Plantea que esta se ocupa de las cuestiones fundamentales, que ofrece un marco para una planeación más detallada y para las decisiones ordinarias, supone un marco temporal más largo, ayuda a orientar las energías y recursos de la organización hacia las actividades de alta prioridad, es una actividad de alto nivel en el sentido de que la alta gerencia debe participar y la planeación operacional procura lograr la eficiencia.

La disponibilidad del sistema de información, así como el análisis y valoración de puestos son requisitos claves que se suman a la existencia de un plan estratégico organizacional como factores esenciales para elaborar la planeación estratégica del capital humano, sin estos elementos no se puede determinar el inventario de competencias del nivel actual y futuro del Capital Humano demandado, ni sus costes. Los requerimientos o necesidades de personal no representan únicamente insuficiencias o excedentes del nivel y calificaciones del Capital Humano, lo más importante es que sus resultados demandan un programa de acciones para que este nivel de capital humano se ajuste de forma consistente con los objetivos del plan estratégico, de ahí que pueden ser necesarias políticas de reclutamiento y contratación o políticas de reducción de personal.

Teniendo en cuenta que el capital humano puede deteriorarse y las habilidades y conocimientos pueden convertirse en obsoletas y perder su potencial de contribución al rendimiento organizacional, se hace necesario mantener un programa de inversiones en actualización de conocimientos y desarrollo de habilidades de manera continua para que este Capital no pierda valor.

La arquitectura de Recursos Humanos como activo estratégico de la empresa

Si el objetivo de la estrategia empresarial es crear una ventaja competitiva sostenida, el objetivo de la estrategia de recursos humanos es maximizar su contribución para asegurar el mismo objetivo y así crear valor. La base del papel esencial de los recursos humanos se encuentra en las tres dimensiones de la cadena de valor de la empresa: la función, el sistema y la conducta de los trabajadores. Becker et al, 2001 diseñan la arquitectura de RRHH que se analiza a continuación:

La función de Recursos Humanos

Los autores antes mencionados señalan que la base de una estrategia de Recursos Humanos de creación de valor es una infraestructura directiva que comprende e implementa la estrategia de la empresa, esto implica un alejamiento de la orientación funcional tradicional de muchos directivos de RRHH hacia una dirección más profunda del papel estratégico que Recursos Humanos puede jugar en la empresa. La dirección de Recursos Humanos para ser eficiente requiere de dos dimensiones esenciales: la dirección técnica de RRHH que incluye las tareas básicas como el proceso de selección del personal, de compensación e incentivos, y la dirección estratégica de RRHH que incluye llevar a cabo dichos servicios de forma que apoye la implementación de la estrategia empresarial, que constituye la influencia más importante en el rendimiento financiero de una organización. Las competencias que los directivos de RRHH necesitan desarrollar con más profundidad y que además tienen un mayor impacto en el rendimiento de la empresa son las competencias estratégicas.

El sistema de RRHH

En cuanto al sistema de RRHH, los autores estudiados lo enmarcan como el eje de la influencia estratégica de Recursos Humanos, cada elemento de dicho sistema está diseñado para maximizar la calidad del Capital Humano en la organización.

Para crear y mantener un Capital Humano con talento se necesita:

- Vincular sus decisiones de selección y promoción con modelos de competencia.
- Desarrollar estrategias que ofrecen apoyo efectivo y oportuno para las capacidades solicitadas por la implantación de la estrategia empresarial.
- Presentar políticas de compensación y rendimiento que atraen, retienen, y motivan a los empleados que demuestran un rendimiento más alto.

Conductas estratégicas de los trabajadores.

En cuanto al papel estratégico de los recursos humanos los autores lo centran implícitamente en las conductas productivas de las personas en la organización. Se define como conductas estratégicas un subgrupo de conductas productivas que sirven

La función de RRHH.

Profesionales de RRHH con competencias estratégicas.

El sistema de RRHH.

Alto rendimiento, políticas y acciones estratégicamente combinadas.

Conducta de los trabajadores.

Competencias, motivación y conductas asociadas estratégicamente planeadas. directamente para implementar la estrategia de la empresa; se trata de conductas que se consideran fundamentales para el buen funcionamiento de la empresa en todos los niveles, así como conductas específicas de situación que son esenciales y claves en las unidades de la cadena de valor en la empresa. La importancia de las conductas está determinada por su papel esencial en la implementación de la estrategia empresarial.

Entender cómo crean valor las personas y los procesos de la empresa es el primer paso, ese análisis revelará tanto los tipos de conductas que se requieren en la empresa, como las que tienen un valor específico en puntos claves de la cadena de valor.

Definiciones de Estrategia

Mantenerse, desarrollarse y tener éxito en un entorno turbulento, complejo, dinámico, cambiante e incierto obliga a las organizaciones a trazar estrategias para imponer su voluntad al entorno y anticiparse a sus efectos aprovechando las oportunidades que éste le ofrece y las fortalezas que ella posee. Se trata del desarrollo de una actitud proactiva, anticipadora, crítica y autocrítica del sujeto organizacional en sus relaciones con el entorno.

Jarniou, (1975); citado por Ansoff, (1976) ven la estrategia como el conjunto de decisiones que determinan la coherencia de las iniciativas y reacciones de la empresa frente a su entorno, Mengunzzato, Renal, (1984) plantean que trata de las proyecciones futuras de la empresa hacia su entorno, de los grandes y generales objetivos, así como de las principales líneas o vías de acción para conseguirlos. Esta definición de estrategia contempla dos aspectos: por un lado, “el qué” la misión, los objetivos, el fin, etc. y por el otro, el cómo los planes y programas de acción, las vías o caminos, en fin los medios para llevar a cabo “el qué”. Los que se centran en “el qué” definen la estrategia como sistema de objetivos generales o resultados esperados. Mientras que los que se sitúan en la perspectiva del “cómo” piensan en la estrategia como la vía o el camino para lograr los objetivos y entonces definen la estrategia organizacional como el programa para definir y alcanzar los objetivos de la organización y poner en práctica su misión, donde el programa es un proceso consciente y activo, que es definido con precisión por parte de la dirección de la organización que asume una actitud estratégica. Harper, Linch, (1992) consideran que es establecer un sistema dinámico de anticipación en el que se destacan y agrupan los aspectos estratégicos diferenciadores empresariales en el marco de un entorno abierto procurando desarrollar una cultura empresarial que apoye las ventajas competitivas que la empresa tiene.

Actividad de aprendizaje Integradora.**Generar un portafolio de evidencias con las siguientes actividades evidencias por producto:**

1. Redactar un ensayo de una cuartilla sobre las competencias y características que debe tener un administrador para la gestión de proyectos en una organización
2. Determinar plan de recursos humanos
3. Personal
4. Generación de informes de rendimiento del equipo de trabajo
5. Generación de reportes de control de fechas y horas consumidas en el proyecto por cada miembro del equipo
6. Generar revisiones de desempeño del personal
7. Generar un reporte sobre los reconocimientos y recompensas aplicados al personal.

Formato de apoyo para realizar la actividad integradora:

1. Redactar un ensayo de una cuartilla sobre las competencias y características que debe tener un administrador para la gestión de proyectos en una organización

PERFIL DE UN ADMINISTRADOR DE LA GESTIÓN DE PROYECTOS

Con la evolución sistemática de la orientación a funciones a la administración basada en proyectos que las organizaciones, los administradores están adoptando cada día más el rol de administrador para la gestión de proyectos, los cuales se están convirtiendo en pieza clave. Ellos son los agentes de cambio: hacen las metas del proyecto propio y utilizan sus habilidades y experiencia para inspirar un sentido de propósito común dentro del equipo del proyecto. Ellos disfrutan de la adrenalina organizada de los nuevos retos y la responsabilidad de la conducción empresarial a su objetivo. En sentido estricto deben realizar una serie de actividades que se listan a continuación.

- Definir y revisar el modelo de negocio para determinar los requisitos, revisiones y controles que garantizan que el proyecto reúne las expectativas de clientes internos y externos.
- Inician y planifican el proyecto mediante el establecimiento de su formato, la dirección, y las líneas de base que permiten cualquier medida de varianza y control de cambios, colaborando y gestionando con usuarios finales y patrocinadores para establecer el progreso y la dirección del proyecto por el logro de metas, objetivos, llegar a la solución de problemas y mitigar riesgos.
- Gestionan la tecnología y administran el personal del proyecto bajo el presupuesto sin dejar de crear un entorno propicio a la entrega de una forma rentable así como son capaces de gestionar la incertidumbre, el cambio rápido, la ambigüedad, sorpresas y un ambiente menos definido en la relación con el cliente mediante el uso adecuado de la información manteniendo una relación respetada y productiva.

Para ello, los administradores para la gestión de proyectos deben contar con una serie de habilidades que podemos definir en tres categorías: personales, técnicas y de administración.

Habilidades personales. Liderazgo, saben predicar con el ejemplo, tienen una actitud positiva, definen las expectativas claramente, son considerados, respetuosos y directos además de tener un buen sentido de urgencia.

Habilidades de administración. Planificación y organización, reclutamiento de personal, negociación eficaz, herramientas de software para administración de proyectos, estimación precisa y control de costos, ejecución y control de proyectos, elaboración de informes y presentaciones de alto nivel ejecutivo, gestión de riesgos y toma de decisiones, gestión eficaz de problemas, gestión del rendimiento, gestión de los proyectos dentro de la organización, gestión de proyectos profesionales (PMP) revisión de exámenes, crecer y mantener un equipo de alto rendimiento y por último deben saber gestionar el cambio.

Habilidades técnicas. En cuanto a éste último punto existen dos tipos de pensamientos aquel que considera que los administradores de la gestión de proyectos no necesitan estar íntimamente relacionados con los detalles técnicos y dejan estos a personal especializado enfocándose ellos más en las tareas de administración y el pensamiento que considera un administrador deben tener un amplio conocimiento técnico del producto a desarrollar. Desde mi punto de vista dependerá de la magnitud del proyecto el tipo de perfil, en proyectos pequeños es común encontrar a los administradores de proyectos haciendo tareas técnicas como administración de las bases de datos, programación o desarrollo de la arquitectura, en proyectos muy grandes es difícil encontrar administradores que conozcan todo el detalle técnico dado que se involucra un amplio número de tecnologías que en sí mismas forma todo una materia de conocimiento.

2. Determinar plan de recursos humanos

Introducción

Gestión de recursos humanos es una parte importante del proyecto de implementación de la norma Moprossoft-Nivel 2. El plan de gestión de recursos humanos es una herramienta que ayuda en la gestión de las actividades de recursos humanos de este proyecto a través del inicio y hasta el cierre.

El plan de gestión de recursos humanos incluye:

- Roles y responsabilidades de los miembros del equipo a través del proyecto
- Organigrama del Proyecto
- Plan de gestión de personal que incluye:
 1. ¿Cómo se adquieren los recursos?
 2. Cronología de los recursos
 3. Formación necesaria para desarrollar habilidades
 4. Como se llevarán a cabo revisiones de desempeño
 5. El reconocimiento y el sistema de recompensas

El propósito del plan de gestión de recursos humanos es lograr el éxito del proyecto, para garantizar que los recursos humanos apropiados adquieren las habilidades necesarias, los recursos son entrenados en caso de detectar cualquier deficiencia en las habilidades, las estrategias de trabajo en equipo se definen claramente, y las actividades del equipo se gestionan con eficacia.

Roles y responsabilidades

Para el proyecto de implementación los siguientes roles and responsabilidades fueron definidos:

ROL	RESPONSABILIDAD	REF	COMPETENCIAS
Patrocinador	Es responsable por el financiamiento del proyecto y representa a la compañía.	PA	Conocimiento de los objetivos y expectativas del proyecto.
Consultor externo	Asesorar al proyecto	CE	Conocimiento en la norma Mo-proSoft y experiencia en la implementación de la misma.
Grupo Directivo	Es el grupo responsable de tomar las decisiones que impactan el objetivo del proyecto y que pongan en riesgo su terminación en tiempo y forma	GD	Conocimiento del esfuerzo requerido para llevar a cabo la planificación estratégica, y sobre todo estar comprometido con éste.
Grupo de Gestión	Es responsable de articular documentar y/o actualizar la Misión, Visión y Valores en conjunto con el plan estratégico.	GG	Conocimiento para administrar los proyectos e implantar los procesos Definidos.
Responsable de Gestión de Negocio	Es responsable de establecer y gestionar el plan de implementación del proceso de gestión de negocio.	RGN	Conocimiento del esfuerzo requerido para llevar a cabo la planificación de Gestión de Proyectos.
Responsable de Gestión de Proyectos	Es responsable de establecer y gestionar el plan de implementación del proceso de gestión de proyectos.	RGPY	Conocimiento de las actividades necesarias para llevar a cabo la gestión de proyectos.
Responsable de Gestión de Recursos	Es responsable de establecer y gestionar el plan de implementación del proceso de gestión de recursos.	RGR	Conocimiento de las actividades necesarias para planificar exitosamente el subproceso de Recursos Humanos y Ambiente de Trabajo.
Responsable de Recursos Humanos y Ambiente de Trabajo	Es responsable de establecer y gestionar el plan de implementación del proceso de gestión de recursos humanos y ambiente de trabajo.	RRHAT	Conocimiento de las actividades necesarias para implantar exitosamente el subproceso de Recursos Humanos y Ambiente de Trabajo.

Responsable de Capacitación	Es responsable de establecer y gestionar el plan de capacitación.	RC	Conocimiento de las actividades necesarias para implantar exitosamente la capacitación Solicitudada.
Responsable de Bienes, Servicios e Infraestructura	Es responsable de establecer y gestionar el plan de implementación del proceso de bienes, servicios e infraestructura.	RBSI	Conocimiento de las actividades necesarias para implantar el subproceso de Bienes, Servicios e Infraestructura
Responsable del Conocimiento de la Organización	Es responsable de establecer y gestionar el plan de implementación del proceso de conocimiento de la organización.	RCO	Conocimiento de definición y administración de repositorios documentales o automatizados.
Grupo de Responsables de Procesos	Es el grupo de los líderes responsables de los planes de implementación de MoproSoft-nivel 2	GRP	Conocimiento de necesidades del proceso con respecto a la Base de Conocimiento.
Responsable de la Administración del Proyecto Específico	Es responsable de definir el Proceso Específico del proyecto a partir del proceso de Desarrollo y Mantenimiento de Software	RAPE	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal, delegación y supervisión, finanzas y desarrollo de Software.
Cliente	Es responsable de realizar la solicitud del proyecto específico	CL	Conocimiento en la expedición de Solicitudes de Cambios.
Responsable del Subcontrato	Es responsable de efectuar el plan y cierre con subcontratistas de acuerdo al contrato establecido.	RSC	Conocimiento en la administración de proyectos.
Responsable de Desarrollo y Mantenimiento de Software	Es responsable del grupo de desarrollo	RDM	Conocimiento y experiencia en el desarrollo y mantenimiento de Software.
Equipo de Trabajo de desarrollo	Son responsables de desarrollar el producto de software en cuestión.	ET	Conocimiento y experiencia de acuerdo a su rol.

Responsable de Administración del Proyecto Específico	Es responsable de elaborar el plan de desarrollo del proyecto específico y gestionar éste hasta entregar el producto al cliente	RAPE	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de Software.
Analista	Responsable del análisis y gestión de los requerimientos del producto de software	AN	Conocimiento y experiencia en la obtención, especificación y análisis de los requerimientos.
Diseñador de Interfaz de Usuario	Responsable de diseñar la interfaz de usuario del producto de software	DU	Conocimiento en diseño de interfaces de usuario y criterios Ergonómicos.
Diseñador	Es responsable de diseñar la arquitectura de software	DI	Conocimiento y experiencia en el diseño de la estructura de los componentes de software.
Programador	Responsable de codificar el producto de software y/o sus componentes en caso de definirse	PR	Conocimiento y/o experiencia en la programación, integración y pruebas unitarias.
Responsable de Pruebas	Responsable de diseñar y llevar a cabo las pruebas de software.	RPU	Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema.
Revisor	Es responsable de revisar y retroalimentar a los líderes de los hallazgos encontrados en las técnicas de desarrollo de software	RE	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de Software.
Responsable de Manuales	Es responsable de crear la documentación del producto de software y/o sus componentes	RM	Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de Software.
Usuario	Es quien finalmente usa el producto de software creado	US	Ninguna

Organigrama del proyecto

El siguiente diagrama muestra la línea de autoridad de los roles esta se realizó basado en los grupos alineación ya que para efectos prácticos del proyecto cada miembro de un equipo tiene el mismo nivel de participación en el reporte de sus actividades y autoridades (Figura 2.1).

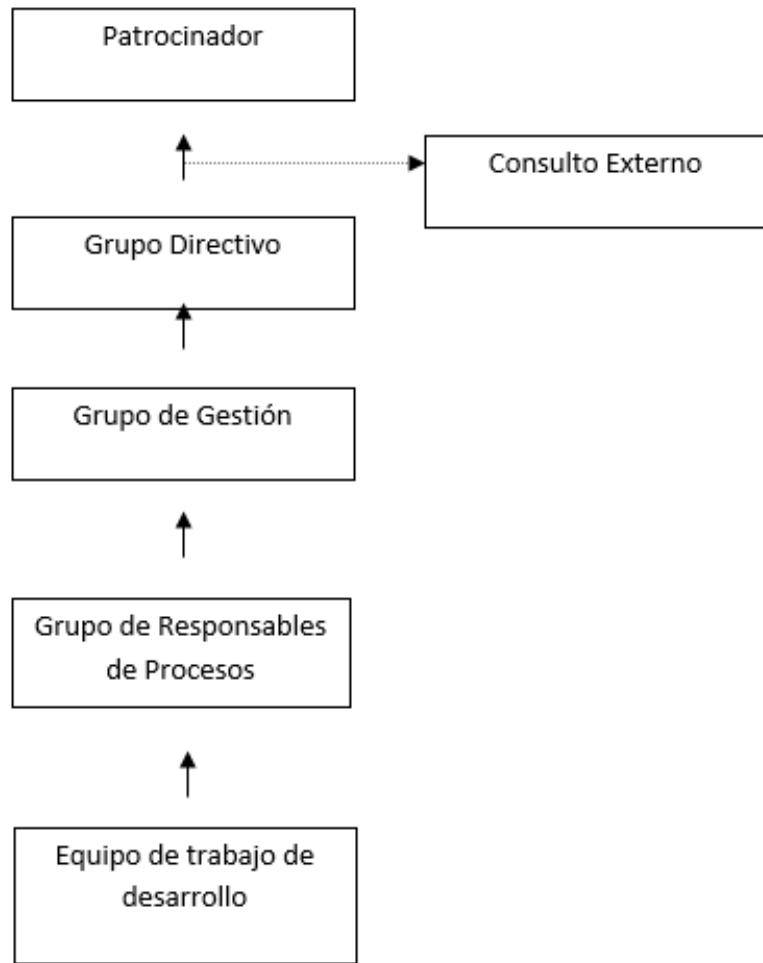


Figura 2.1: Organigrama del proyecto

Gráficas organizacionales del proyecto

La siguiente gráfica RACI muestra la relación entre las tareas del proyecto y miembros del equipo. Los cambios propuestos a las responsabilidades del proyecto deben ser revisados ??y aprobados por el director del proyecto. Los cambios se propondrán de conformidad con el proceso de control de cambios del proyecto. A medida que se realizan cambios en los documentos del proyecto se actualizarán y se redistribuyan en consecuencia.

Clave:

R - Responsable de completar el trabajo

A - Responsable de velar por la realización de tareas / cierre de sesión

C - Consultado antes de tomar cualquier decisión

I - Informado de cuando se ha hecho una acción / decisión

	ROL									
Proceso-MoproSoft-NIVEL 2	PA	CE	GD	GG	RGR	RRHA	RC	RBSI	RCO	GRP
Gestión de Negocio	C	I	AC	AI						
Gestión de Procesos	C	I	AC	AI						
Gestión de Proyectos	C	I	AC	AI						
Gestión de Recursos	C	I	AC	AI	ACI					
Recursos Humanos y Ambiente de Trabajo	C	I	AC	AI		ACI				
Bienes Servicios e Infraestructura	C	I	AC	AI				ACI		
Conocimiento de la Organización	C	I	AC	AI					ACI	
Administración de Proyectos Específicos	C	I	AC	AI	ACI					AI
Desarrollo y Mantenimiento de Software	C	I	AC	AI						
Capacitación		R					RA			

	ROL														
Proceso-MoproSoft-NIVEL 2	RAPE	CL	RSC	RDM	ET	AN	DU	DI	PR	RPU	RE	RM	ET	US	
Administración de Proyectos Específicos	ACI	RCI	A	A	A										
Desarrollo y Mantenimiento de Software	ACI	RCI	A	A	A	A	A	A	A	A	A	A	A	A	A

Adquisición de Personal:

Para el proyecto el personal del proyecto estará compuesto por recursos internos y una empresa consultora que se encargará de las funciones de capacitación y soporte al personal interno en la norma. El gestor de procesos y de proyecto va a negociar con los gerentes funcionales y el departamento con el fin de identificar y asignar los recursos de acuerdo con la estructura de la organización del proyecto. Todos los recursos deben ser aprobados por el gerente / departamento funcional adecuada para que el recurso puede comenzar cualquier proyecto de trabajo. El equipo

del proyecto no se podrá posicionar para este proyecto y todos los recursos se mantendrá en su espacio de trabajo actual.

Los calendarios de recursos:

El proyecto tendrá una duración de diez meses. Todos los recursos son necesarios antes de iniciar el proyecto. El histograma de recursos a continuación muestra la distribución de horas requeridas por cada rol se ajusto para no exceder un máximo de 100 horas durante los 10 meses (Figura 2.2).

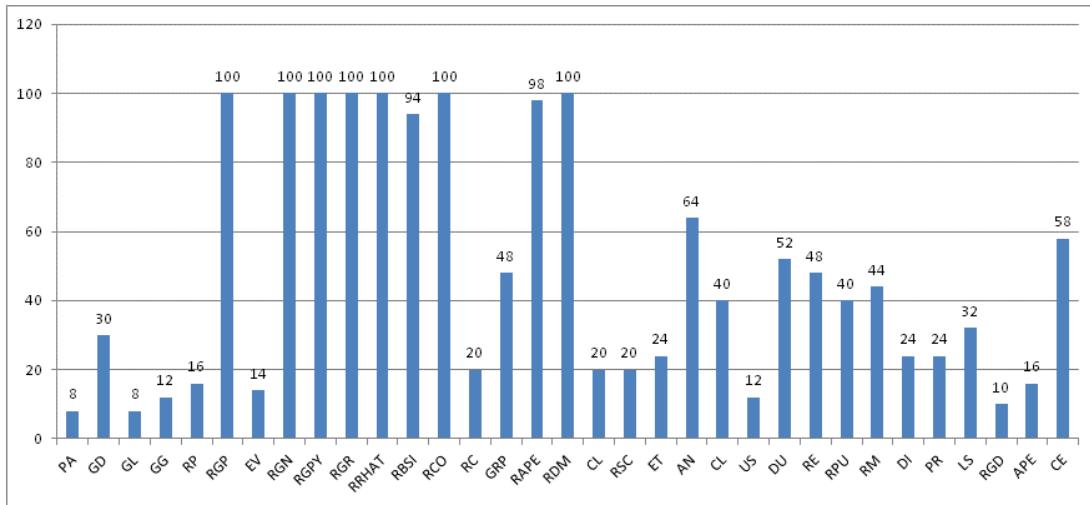


Figura 2.2: Calendario de los recursos

El siguiente cronograma muestra el calendario de recursos ajustado de acuerdo al total de actividades. Las celdas en color amarillo muestran la distribución de las horas de consultoría las cuales se ajustaron a 8 horas por semana sin exceder las 250 horas totales del proyecto como se explico en los requerimientos iniciales.

Formación:

La empresa consultora deberá capacitar a todo el personal de la organización de acuerdo a la función o rol descrito anteriormente y conforme al plan de capacitación. Si se identifican otras necesidades de formación, la financiación será proporcionado de la reserva de proyectos.

Evaluación del Desempeño:

El director del proyecto revisará las actividades de trabajo asignadas a cada miembro del equipo en el inicio del proyecto y comunicará todas las expectativas de trabajo a realizar. El director del proyecto evaluará a cada miembro del equipo a través del proyecto para evaluar su desempeño y la eficacia con que están completando su trabajo asignado. Antes de la liberación de los recursos del proyecto, el director del proyecto se reunirá con el gerente funcional adecuada y proporcionar retroalimentación sobre el desempeño del proyecto empleado. Los gerentes funcionales a continuación, realizarán una revisión formal del funcionamiento de cada miembro del equipo.

Reconocimiento y recompensas:

Aunque el alcance de este proyecto no permite el tiempo suficiente para proporcionar el entrenamiento cruzado o potencial de recompensas monetarias hay varias reconocimiento y gratificación artículos previstos para los miembros del equipo del proyecto.

- Al completar con éxito el Proyecto de Actualización del software , un partido se llevará a cabo para celebrar el éxito de cada miembro del equipo con las familias presentes los miembros del equipo.
- Una vez finalizado con éxito el proyecto, cualquier miembro del equipo que ha completado satisfactoriamente todos los paquetes de trabajo asignados a tiempo recibirán un certificado de agradecimiento del director general.
- Los miembros del equipo que completan con éxito todas las tareas asignadas tendrán su foto tomada para su inclusión en el boletín de la empresa.
- La empresa proporcionará entradas de cine familiar gratuitos para los dos principales actores en cada proyecto.
- Los líderes gestores de cada proceso que logren completar todas sus actividades considerando un ahorro del 30 % de las horas asignadas serán considerados para una promoción en el corto plazo.

Políticas de Seguridad

Toda la información deberá ser manejada en formato electrónico y deberá actualizarse y consultarse únicamente dentro del portal designado para éste uso.

La impresión de reportes o cualquier otro tipo de documento que fuera estrictamente necesario deberá contener la leyenda “Confidencial – Propiedad de GarySystems no permitida su distribución o reproducción” y no podrá ser portada fuera de la compañía.

Todo el personal deberá portar su gafete de identificación dentro de las instalaciones y el personal consultor deberá portar el gafete que lo identifica como tal.

Sólo se permite el uso de laptop dentro de la compañía la cual deberá ser autorizada por el área de IT bajo las políticas de seguridad de la información establecidas.

VO.BO. PATROCINADOR DEL PROYECTO

Aprobado por Andrés Torres Barrera en su carácter de Director de la Compañía GarySystems SA de CV.

Firma: _____ Date: _____

4. Generación de informes de rendimiento del equipo de trabajo

Cuadro 2.2: Tabla de Balance

Factores	Bajo	Mod/Bajo	Moderado	Mod/Alto	Alto
Control	PA, RGPY, RC, RM, DI, PR, EV, RBSI, CL, RPU	RP, RGR, RAPE, LS, RGN, RCO, US, DU, RE, RGP, RRHAT	RDM, APE, CE		
Influencia Social	RP, RGR, RAPE, LS, EV, RBSI, CL	PA, RGPY	RC, RM, DI, PR, , RPU, , DU, RE	RGN, RCO, US	RGP, RRHAT, RDM, APE, CE
Paciencia	EV, RBSI, CL, RPU	EV, RBSI, CL, RPU	RP, RGR, RAPE, LS	PA, RGPY, RC, RM, DI, PR	RGN, RCO, US, DU, RE
Precisión	RP, RGR, RAPE, LS, EV, RBSI, CL	RP, RGR, RAPE, LS	RPU	DI, PR, RGN	PA, RGPY, RC, RM, RCO, US, DU, RE
Ambición	PA, RGPY, RC, RAPE, LS	RM, DI, PR, RP, RGR	RGP, APE, CE	EV, RBSI, RGN, RCO, US, DU, RE	CL, RPU, RRHAT, RDM
Expectativas positivas	RM, DI, PR, RP, RGR	PA, RGPY, RC, RAPE, LS	CL, RPU, RRHAT, RDM	RGP, APE, CE	EV, RBSI, RGN, RCO, US, DU, RE
Compostura	CL, RPU, RRHAT, RDM	RM, DI, PR, RP, RGR	PA, RGPY, RC, RAPE, LS	EV, RBSI, RGN, RCO, US, DU, RE	RGP ,APE, CE
Analítico	EV, RBSI, RGN, RCO, US, DU, RE	RGP, APE, CE	RM, DI, PR, RP, RGR	PA,RGPY,RC, RAPE,LS	CL, RPU, RRHAT,RDM
Orientación a resultados	RGP , APE, CE	EV, RBSI, RGN, RCO, US, DU, RE	PA, RGPY, RC, RAPE, LS	CL, RPU, RRHAT, RDM	RM, DI, PR, RP, RGR
Expresividad	CL, RPU, RRHAT, RDM	PA, RGPY, RC, RAPE, LS	RGP, APE, CE	RM, DI, PR, RP, RGR	EV, RBSI, RGN, RCO, US, DU, RE
Trabajo en equipo	PA, RGPY, RC, RAPE, LS	CL, RPU, RRHAT, RDM	RM, DI, PR, RP, RGR	EV, RBSI, RGN, RCO, US, DU, RE	RGP ,APE, CE
Orientación a la calidad	CL, RPU, RRHAT, RDM	RM, DI, PR, RP, RGR	EV, RBSI, RGN, RCO, US, DU, RE	RGP,APE, CE	PA, RGPY, RC, RAPE, LS

Desempeño general del equipo. El equipo muestra un desempeño balanceado en 11 de las 12 fortalezas necesarias, lo cual es bastante bueno los miembros del equipo muestran fortalezas y debilidades promediadas y al menos existe un elemento del equipo que muestra fortaleza todas las áreas. Salvo las primeras dos fortalezas. A continuación se detalla el análisis.

Control. El control se define como la tendencia a hacerse cargo, ser assertivo, y / o tomar el

control de una situación.

Calificación - Menor

- Esto no es un factor con características típicas de los roles PA, RGPY, RC, RM, DI, PR, EV, RBSI, CL, RPU y es escasamente presentado en los roles RP, RGR, RAPE, LS, RGN, RCO, US, DU, RE, RGP, RRHAT.

Calificación Mayor

- Puede ser abrumador para otros miembros del equipo
- Puede ser contundente y sarcástico con los demás
- Decisivo y seguro de sí mismo, con una tendencia a tomar decisiones rápidas
- Por lo general, responde a los desafíos

Calificación Mod/Alto

- toma decisiones con facilidad
- Puede tener falta de tacto y diplomacia
- Le gusta iniciar la actividad

Recomendaciones finales:

1.- Se recomienda fortalecer a cada miembro en el factor en que resulta débil para todas las 10 competencias que resultaron bien. No sé ven riesgos específicos pero se puede incrementar la eficiencia del equipo.

2.- En la competencia de control el líder deberá ejecutar las siguientes recomendaciones.

- Ayudar a desarrollar una mayor sensibilidad hacia las personas.
- Hacer preguntas específicas.
- Utilizar respuestas directas a sus preguntas.
- Sea claro y firme en el establecimiento de límites.

Bibliografía

1. Mengunzzato, M.; Renau, J.J. 1991. La Dirección Estratégica de la Empresa. Un enfoque innovador del management. Ediciones Ariel economía pp.87,212
2. Becker, E et al. 2001. El cuadro de mando de RR.HH vinculado a las personas, la estrategia y el rendimiento de la empresa, Ediciones Gestión 2000, pp. 19.
3. Pierre Louart. Gestión de los Recursos Humanos. Ediciones gestión 2000 S.A. Pág. 55.
4. Kaplan,R. ; Norton, D. 2006. Incrementando los resultados mediante el alineamiento estratégico en toda la organización. Ediciones Gestión 2000, pp. 43-47
5. Ulrich, D. et al, 2003. El futuro de la dirección de recursos humanos, Ediciones Gestión 2000
6. Harper y Lynch, 1992. Motivación de personal y clima laboral. Ediciones de Publicaciones Económicas. España
7. Stoner, J. et al, 1996. Administración. Ediciones Pearson Educación, 6^a edición, Maxico
8. Werther, W; Davis, K. 1996. Administración de personal y recursos humanos. Ediciones MacGraw-Hill
9. Ansoff, I. et al, 1976. From strategic planning to strategic management. Ediciones Wiley, 1976. Michigan
10. Besseyre, 1997. Gestión estratégica de los recursos humanos. Ediciones Deusto
11. Cuesta, A. 2005. Tecnología de gestión de Recursos Humanos. Ediciones Academia, 2^a edición.

3 — Gestión de la calidad de un proyecto

César Eduardo Velázquez Amador¹, Yosly Caridad Hernández Bieliukas²

¹ César Eduardo Velázquez Amador (covelazq@correo.uaa.mx), Departamento. de Sistemas de Información de la Universidad Autónoma de Aguascalientes (UAA)

²Yosly Caridad Hernández Bieliuka (yosly.hernandez@ciens.ucv.ve), Facultad de Ciencias, Unidad de Educación a Distancia, Universidad Central de Venezuela,

OBJETIVO

El presente capítulo tiene como objetivo proporcionar los elementos teóricos necesarios para comprender en qué consiste la gestión de la calidad en un proyecto de desarrollo de software.

RESUMEN DEL CAPÍTULO

El capítulo presenta primeramente los fundamentos teóricos relacionados a la gestión de la calidad, continua con la calidad del proceso y del producto, el aseguramiento de la calidad, las revisiones técnicas formales, el plan de calidad, las métricas técnicas formales, los factores de calidad de McCall y las métricas de calidad.

CONOCIMIENTOS PREVIOS

Para una mejor comprensión de los temas expuestos en este capítulo es recomendable que el estudiante cuente con experiencia tanto en el análisis y diseño de sistemas, así como en los procesos de desarrollo de los mismos; esto facilitará el comprender y contextualizar la actividad de gestión de la calidad dentro del desarrollo de sistemas.

INTRODUCCIÓN

Actualmente en el desarrollo de sistemas formal, se reconoce la importancia de las actividades encaminadas a asegurar la calidad del producto de software. Una adecuada gestión de la calidad requiere de actividades previas al desarrollo del software como es una adecuada planeación de la calidad y de actividades posteriores a la entrega del producto, como son el registro de las incidencias y de las lecciones aprendidas para la mejora de los procesos. Lo anterior pone de manifiesto la importancia del estudio y de una adecuada implementación de los conceptos sobre la gestión de la calidad por parte de los profesionales dedicados al desarrollo de software; este es el propósito que se sigue en el presente capítulo.

FUNDAMENTOS TEÓRICOS

Lograr un alto nivel de calidad de un producto o servicio es el objetivo de muchas organizaciones. La responsabilidad de los administradores de la calidad en una organización es asegurar que se cumpla el nivel requerido de calidad de un producto.

La administración de la calidad del software se estructura en tres actividades principales:

- Aseguramiento de la calidad.- Establecimiento de un marco de trabajo de procedimientos y estándares organizacionales que condice a software de alta calidad.
- Planeación de la calidad.- La selección de procedimientos y estándares adecuados a partir de este marco de trabajo y la adaptación de éstos para un proyecto de software específico.
- Control de la calidad.- La definición y promulgación de los procesos que aseguran que los procedimientos y estándares para la calidad del proyecto son seguidos por el equipo de desarrollo de software.

Puesto que los miembros del equipo de aseguramiento y control de la calidad son independientes, pueden tomar una visión objetiva del proceso y reportar problemas y dificultades a los administradores principales de la organización.

Aseguramiento de la Calidad

Las actividades de aseguramiento de la calidad (QA) definen un marco de trabajo para lograr la calidad del software. Los procesos de QA comprenden definir o seleccionar estándares aplicables al proceso de desarrollo de software o a los productos de software.

Existen dos tipos de estándares como parte del proceso de aseguramiento de la calidad:

- Estándares del producto.- Aplican al producto de software a desarrollar. Incluyen estándares de documentos (estructura de los documentos, formato de los encabezados, etc) y estándares de codificación.
- Estándares del proceso.- Definen los procesos a seguir durante el desarrollo del software. Incluyen definiciones de los procesos de especificación, de diseño y de validación y una descripción de los documentos a generar en el transcurso de estos procesos.

Existen tres actividades para asegurar la calidad de un producto de software, estas son inspección, demostración de funcionamiento correcto y pruebas.

La inspección es un proceso orientado al trabajo en equipo para asegurar la calidad y se aplica a todas las etapas del proceso de software. Se hacen revisiones para determinar si el equipo de trabajo está siguiendo los estándares de producto y proceso definidos.

Una demostración de funcionamiento correcto es una técnica matemática o lógica usada para convencernos de que un programa hace lo que debe. Esta demostración son métodos formales. Los programas no se ejecutan durante este proceso de demostración, solo se inspecciona su código fuente.

En las pruebas se ejecutan los programas.

Calidad del Proceso y del Producto

La calidad del proceso de desarrollo afecta directamente a la calidad de los productos a entregar, como se muestra en la Figura 3.1

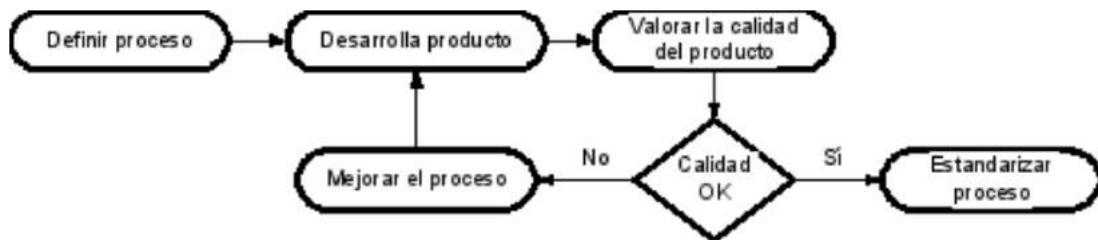


Figura 3.1: Calidad del Proceso y del Producto

Actividades de SQA

El Instituto de Ingeniería de Software recomienda un conjunto de actividades de SQA que son desarrolladas por un grupo independiente y son las siguientes:

- Establecimiento de un plan de SQA para un proyecto
- Participación en el desarrollo de la descripción del proceso de software del proyecto.
- Revisión de las actividades de ingeniería de software para verificar su ajuste al proceso de software definido.
- Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software
- Asegurar que las desviaciones del trabajo y los productos de software se documenten y se manejen de acuerdo con un procedimiento establecido.
- Revisar lo que no se ajuste a los requisitos e informar a sus superiores.

Revisiones Técnicas Formales

Una revisión técnica formal (RTF) es una actividad de garantía de la calidad del software que es llevada a cabo por los ingenieros de software. Sus objetivos son:

- Descubrir errores en la función, la lógica o la implementación.
- Verificar que el software alcance sus requisitos
- Garantizar que el software es representado de acuerdo a ciertos estándares
- Conseguir un software desarrollado en forma uniforme.
- Hacer que los proyectos sean más manejables.

Cada RTF se lleva a cabo mediante una reunión que deberá estar bien planificada, controlada y atendida. Cada RTF se centra en una parte específica (y pequeña) del software total.

Se realiza un informe de la revisión en donde se responde:

- ¿Qué fue revisado?
- ¿Quién lo revisó?
- ¿Qué se descubrió y cuáles son las conclusiones?

Al final de la revisión se decide si (1) se acepta el producto sin posteriores modificaciones, (2) si se rechaza el producto o (3) si se acepta el producto provisionalmente.

El plan de Calidad

- 1.- Introducción al plan de calidad
 - a) Propósito del plan.
 - b) Ambito del plan.
 - c) Descripción del producto (Mercado al que se dirige, expectativas de calidad).
- 2.- Descripción de los productos de trabajo de ingeniería del software del Plan de Calidad (Modelos, Documentos, Código Fuente).
- 3.- Estándares y prácticas aplicables en todo el proceso de software.
 - a) Descripción de estándares aplicables en todo el proceso de software.
 - b) Descripción de buenas prácticas aplicables en todo el proceso de software.
 - c) Identificación y justificación de los atributos de calidad importantes del producto (Seleccionar de la lista de atributos de calidad del software).
 - d) Metas de calidad para cada atributo de calidad importante del producto.
 - e) Procedimientos para información y seguimiento de errores.
- Algunos de los atributos de calidad del software que se pueden emplear son los siguientes:
 - Seguridad
 - Protección
 - Fiabilidad
 - Flexibilidad
 - Robustez
 - Comprensión
 - Experimentación
 - Adaptabilidad
 - Modularidad
 - Complejidad
 - Portabilidad
 - Usabilidad
 - Reutilización
 - Eficiencia
 - Aprendizaje
- 4.- Acciones y tareas del plan de calidad y su ubicación en el proceso de software.
 - a) Planes del producto (Proceso de software)
 - b) Inserción de las acciones y tareas del plan de calidad en el plan del producto
 - Pruebas.
 - Aseguramiento de que las desviaciones de trabajo y productos de software se documenten de acuerdo a un proceso de software.
 - Revisiones técnicas formales (RTF).- Permiten la revisión del cumplimiento del proceso de software definido, descubrir errores en la función, la lógica o implementación, verificar que el software alcanza sus requisitos.
- 5.- Herramientas y métodos que soportan las acciones del plan de calidad (Herramientas automatizadas, CASE, diagramas, métricas, etc.).
- 6.- Procedimientos de gestión de configuración de software para gestionar el cambio.
- 7.- Métodos para ensamblar, salvaguardar y mantener todos los registros relacionados al plan de calidad.
- 8.- Papeles y responsabilidades en la organización relativas a la calidad del producto.
 - a) Definir papeles
 - b) Definir responsabilidades
 - c) Designación del personal en cada papel

9.- Riesgos y administración de riesgos (Contiene los riesgos clave que podrían afectar la calidad del producto y las acciones para abordar estos riesgos).

Métricas Técnicas del Software

Un elemento clave de cualquier proceso de ingeniería es la medición.

Empleamos las medidas para valorar la calidad de los productos de ingeniería o los sistemas que construimos.

La ingeniería de software no está basada en leyes cuantitativas básicas de la Física, en su lugar se usan medidas indirectas que dan lugar a métricas que proporcionan una indicación de la calidad de algún tipo de representación del software.

Factores de calidad de McCall

McCall, Richards y Walters propusieron una clasificación útil de los factores que afectan la calidad del software. Estos factores se concentran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad para experimentar cambios y su capacidad para adaptarse a nuevos entornos.



Figura 3.2: Factores de calidad de McCall.

Corrección.- Grado de cumplimiento de la especificación.

Integridad.- El grado de control de acceso al software.

Flexibilidad.- El esfuerzo necesario para modificar un programa en operación.

Interoperabilidad.- El esfuerzo necesario para acoplar un sistema con otro.

Es difícil y en algunos casos imposible, desarrollar medidas directas de estos factores de calidad, por lo que muchas solo pueden medirse subjetivamente; por lo que es común que las métricas adquieran la forma de una lista de comprobación que se emplea para “asignar una graduación” a atributos específicos del software.

Métricas de Calidad

A continuación se enlistan algunas de las principales métricas de gestión de la calidad empleadas en la gestión de proyectos.

- FURPS (Funcionalidad, facilidad de empleo, fiabilidad, rendimiento y capacidad de soporte)

- Métricas basadas en la función (PF)
- Métrica de Bang
- Métricas de Calidad de Especificación
- Métricas de Diseño de Componentes
- Métricas de Complejidad
- Métricas de la Interfaz
- Métricas del Código Fuente

CONCLUSIONES

En este capítulo se expusieron los conceptos fundamentales referentes a la gestión de la calidad en el desarrollo de sistemas de software. Es necesario adaptar lo aquí expuesto a las características que presente tanto la organización, como el proyecto a desarrollar, pero sin perder nunca de vista la importancia de la gestión de la calidad para mejorar los procesos y productos de software.

BIBLIOGRAFÍA

- Braude Eric: Ingeniería de Software: una perspectiva orientada a objetos, Alfaomega (2003)
- Center for Software Engineering of the USC
- Fraunhofer Institute for Software Engineering Germany)
- Grady Booch: Análisis y Diseño Orientado a Objetos, Segunda Edición, Addison Wesley Longman, Pearson
- Ian Sommerville: Ingeniería de Software, 6^a Edición, Addison Wesley
- Ivar Jacobson, Grady Booch, James Rumbaugh: El Proceso Unificado de Desarrollo de Software, Rational Software Corporation, Addison Wesley, Edición en Español (2000)
- Joseph Schmuller: Aprendiendo UML en 24 horas, Prentice Hall
- Pankaj Jalote: “An Integrated Approach to Software Engineering”, Springer (1997)
- Roger S. Pressman: Ingeniería de Software, un enfoque práctico, Sexta Edición, McGraw Hill (2006)
- Shari Lawrence Pfleeger: Ingeniería de Software, Teoría y Práctica, Primera Edición, Prentice Hall (2002)
- Software Engineering Institute of Carnegie-Mellon University
- Stephen R. Schach: Ingeniería de Software Clásica y Orientada a Objetos, Sexta Edición, McGraw Hill (2006)

4 — Gestión de Riesgos en proyectos de software

Julio Ariel Hurtado¹, Francisco Javier Álvarez Rodríguez²

*¹Julio Ariel Hurtado (ccollazo@unicauca.edu.co), Grupo IDIS de la Universidad del Cauca.
Colombia,*

*²Francisco Alvarez Rdz. (fjalvar@correo.uaa), Departamento. de Ciencias de la Computación
Sistemas de la Universidad Autónoma de Aguascalientes (UAA)*

a) Introducción

Los motivos por los que los proyectos de software fracasan son diversos. Sin embargo, una de las principales causas es la ausencia de mecanismos para realizar revisiones constantes sobre cuáles son los factores que pueden fallar en el proyecto, es decir, los riesgos. Un riesgo es una situación potencial que puede amenazar el correcto desempeño de un proyecto y que puede surgir tanto a partir de los procesos existentes, como de las acciones futuras. Un riesgo en un proyecto software es cualquier factor que amenace o limite sus metas, objetivos y/o artefactos.

b) Gestión del Riesgo

La gestión de los riesgos, consiste en predecir la probabilidad de que una determinada acción tenga lugar, así como del impacto de dicha ocurrencia. La gestión de riesgos juega un papel fundamental en el mantenimiento de la estabilidad y en la eficiencia del proyecto. En forma planeada, esta gestión permite manejar los obstáculos potenciales que puedenemerger y que impiden el éxito del proyecto y/o bloquean que el equipo del proyecto alcance sus metas.

i) Planificación del riesgo

Durante la etapa de Planificación del proyecto, es importante considerar la planificación de la gestión de los riesgos. El objetivo de la planificación de la gestión de los riesgos es poder anticipar, identificar y dirigir aquellas amenazas que pueden interferir en el éxito del proyecto; es decir, aquellas situaciones difíciles que, desatendidos, pueden impactar negativamente sobre el proyecto.

El Plan de Gestión de Riesgos documenta la forma en que se pueden dirigir las actividades de gestión tales como la identificación, el análisis, los planes de respuesta y monitoreo, el control

y el reporte de riesgos. Aunque todos los riesgos pueden ser eliminados, se pueden desarrollar planes de mitigación y contingencia para reducir su impacto en el caso de que surjan. El Plan de Gestión de Riesgos determina un método consistente para definir los recursos, las actividades y los artefactos necesarios como respuesta a los riesgos. El Plan de Gestión de Riesgos documenta la gestión de estos procesos e identifica los roles y las responsabilidades del equipo del proyecto, del organismo de gestión y del resto de participantes. El Plan describe quién es responsable de los riesgos de gestión, cómo tiene que realizarse su seguimiento a lo largo de todo el ciclo de vida del proyecto y cómo deben desarrollarse e implementarse los planes de migración y contingencia.

ii) Priorización del riesgo

Manejar un gran número de riesgos es un riesgo por sí mismo. Por ello es importante priorizar el riesgo para permitir focalizar el esfuerzo de la gestión. Para dicha priorización normalmente se utiliza la probabilidad de ocurrencia del riesgo, el impacto de que ocurra, así como la exposición.

- Probabilidad: es un valor de 0 a 1 que cuantifica la posibilidad de que el riesgo ocurra.
- Impacto: es un valor del 0 al 10 que cuantifica el daño que el riesgo le haría al proyecto en caso de que ocurra.
- Exposición: es el producto de la probabilidad de ocurrencia con el impacto, permite determinar la magnitud del riesgo la cual corresponde a otra escala de evaluación.

Un **plan de mitigación**, también llamado plan de riesgos o plan de respuesta a los riesgos, es un documento que registra el parecido de los eventos riesgosos que sucederán en un proyecto y reduce el impacto de dichos eventos si llegaran a suceder. Se desarrollan opciones y acciones en un plan de mitigación para mejorar las oportunidades del proyecto y también las amenazas a los objetivos del proyecto se reducen a "por debajo de un umbral aceptable".

Un **plan de contingencia** es el conjunto de procedimientos alternativos, cuyo fin es permitir el normal funcionamiento de esta, aún cuando alguna de sus funciones se viese dañada por un accidente interno o externo.

iii) Gestionando el Riesgo en la práctica

La tabla 4.1 permite ver el consolidado de la planeación y monitoreo de dos riesgos, la rotación de personal y la complejidad de la tecnología en el marco de un proyecto que sigue un conjunto de 4 fases. La magnitud ha sido medida de acuerdo a las tablas 4.2 y 4.3. Se puede observar como a medida que se avanza uno de los riesgos se logra mitigar mientras el otro se vuelve un problema que cada vez genera mayor magnitud.

Magnitud	Fase/Iteración del proyecto				Riesgo
	Concepción	Elaboración	Construcción	Transición	
1	1	1	1	1	
M	S	S	A		Rotación de Personal
A	S	S	M		Tecnología compleja

Cuadro 4.1: Planeación y seguimiento del riesgo en el Proceso Unificado

Magnitud	Exposición
A – Alto	6.4 a 10.0
S – Significativo	3.6 a 6.4
M – Moderado	1.6 a 3.6
I – Inferior	0.4 a 1.6
B – Baja	0 a 0.4

Cuadro 4.2: Relación entre la magnitud y la exposición del riesgo

Nivel	Valores de Probabilidad	Valores de Impacto
Alto	0.8 a 1.0	8.0 a 10.0
Significativo	0.6 a 0.8	6.0 a 8.0
Moderado	0.4 a 0.6	4.0 a 6.0
Inferior	0.2 a 0.4	2.0 a 4.0
Bajo	0.0 a 0.2	0.0 a 2.0

Cuadro 4.3: Relación entre la magnitud y los factores de la exposición al riesgo

iv) Documentando el Riesgo

Para que se permita hacer la cuantificación y análisis se debe tener lo siguiente: una claridad sobre lo que es el riesgo, sus indicadores expresando la necesidad de monitorización y los planes de contingencia y mitigación asociados. La tabla 4.4 presenta una pequeña plantilla a manera de ejemplo para la gestión del riesgo. Es importante que cada riesgo sea atendido por alguien ese es el responsable de gestionar el riesgo.

Riesgo	Rotación de Personal
Magnitud	Moderado
Impactos	Retraso en el proyecto, Calidad del producto comprometida
Indicadores	Más de un cambio de personal al mes
Plan de mitigación	Trabajo por parejas con rotación para que haya propagación de conocimiento.
Plan de Contingencia	Cuantificar pérdidas y negociar tiempos de entrega y costos asociados
Responsable del riesgo	Jefe de Proyecto

Cuadro 4.4: Plantilla para la gestión de riesgo

Actividad de aprendizaje Integradora.

Generar un portafolio de evidencias con las siguientes actividades evidencias por producto:

1. Generar lista de Riesgos

2. Realizar análisis de riesgos
3. Planear la respuesta a los riesgos
4. Realizar seguimiento y control de riesgos

Formato de apoyo para realizar la actividad integradora:

1. Generar lista de Riesgos

Resumen Ejecutivo

GarySystems empresa tiene siete años en el mercado de desarrollo de software a medida para diversos organismos gubernamentales, así como soluciones para las empresas privadas. A pesar de que tiene tiempo en el mercado de la compañía no ha adoptado un modelo de calidad para la resolución de problemas de la mala organización y la gestión de procesos para que la cultura de la calidad no exista.

De esta manera, la empresa se preocupa por hacer frente al final del año con un aumento del personal, sin embargo un nuevo proyecto de desarrollo de software permitirá obtener la liquidez necesaria en el cargo de los gastos de iniciar un proyecto de implementación de mejora de la calidad. Teniendo en cuenta las experiencias exitosas de muchas pequeñas empresas de software en México a través la adopción del modelo MoProSoft que se decida adoptar este modelo en su empresa.

Objetivo del Proyecto/Justificación

En esta sección se describe la finalidad y justificación del proyecto en forma de caso de negocio y objetivos. El caso de negocio debe proporcionar el razonamiento detrás de la necesidad de este proyecto en lo que respecta a la función de la empresa.

Los objetivos de la empresa para este proyecto son en apoyo directo de nuestro plan estratégico corporativo para implementar MoProSoft y mejorar la cultura de la calidad. Dichos objetivos son descritos continuación.

- 1. *Implementar MoProSoft (NMX-I-059-NYCE-2005) a nivel 2 cual incluye los nueve procesos siguientes en x días.*
- 2. *Implementar un modelo formal de comunicación en toda la organización en los caminos verticales y horizontales.*

Roles and Responsabilidades

Los Roles y Responsabilidades son aquellos definidos en el plan de Administración de Recursos Humanos.

Identificación del Riesgo

Durante la identificación de riesgos se desarrollan las posibles fuentes de riesgo y eventos de riesgo potenciales. Las categorías de riesgo se pueden adaptar con el tiempo, según la demanda específica del riesgo. Después de identificar y clasificar el evento de riesgo, éste debe inscribirse en el registro de riesgos.

El siguiente diagrama muestra las categorías de riesgo predefinidos para éste proyecto.

Categorías de riesgo predefinidas nos proporcionan una estructura que ayuda a asegurar que podemos encontrar el conjunto de riesgos asociado a cada etapa del proyecto.

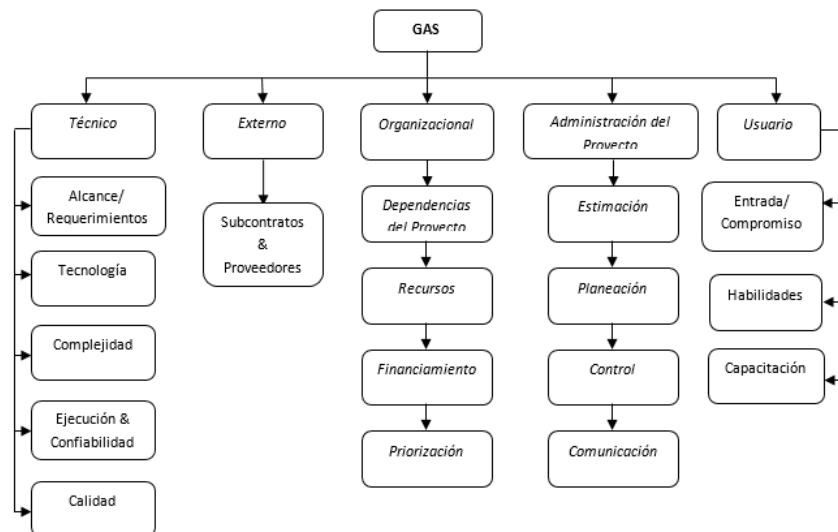


Figura 4.1: Diagrama de categorización de riesgos

Fuente

La identificación de riesgos se realiza a través de los ciclos de vida de un proyecto, aunque la mayoría de los riesgos deben ser identificados desde el principio para la planificación de respuesta y seguimiento adecuados puede ocurrir. A continuación se debe considerar como herramientas y técnicas para la identificación de riesgos:

- Análisis de las prestaciones de alto nivel
- Análisis de la EDT y el cronograma del proyecto
- Análisis de las solicitudes de cambio de alcance
- Análisis de los supuestos del proyecto
- La entrada del equipo de proyecto (que puede tomar la forma de entrevistas , sesiones de lluvia de ideas , y / o la técnica Delphi)
- Las partes interesadas y el patrocinador de entrada
- Sesiones de identificación de riesgos formales
- Lecciones aprendidas anteriores
- Las auditorías y revisiones SQA
- Los informes de rendimiento y el estado
- Diagramas de técnicas como causa y efecto, diagramas de procesos o flujos de sistemas y diagramas de influencia

Documentación

Todos los riesgos identificados deben estar documentados y se introducen en el registro de riesgos, que se mantiene en la siguiente lista

Categoría de riesgo	Riesgo de disparo	Resultado potencial	Creado Por	Fecha Creado	Fuente

4. Análisis de Riesgo

Después de un riesgo o grupo de riesgos se ha identificado y documentado, se debe realizar el análisis de riesgos. Durante el análisis de riesgos, cada evento de riesgo potencial se analiza para:

- La probabilidad de que ocurra el riesgo
- El impacto del riesgo en caso de producirse

Para efectos de este proyecto las probabilidades de riesgo son las que se definen a continuación:

Categoría	Probabilidad	Descripción
Muy Alto	0.90	Experado a ocurrir
Alto	0.70	No es común que ocurra
Probable	0.50	Puede o no ocurrir
Bajo	0.30	Es común que no ocurra
Muy bajo	0.10	No es esperado a ocurrir

Cuadro 4.5: Matriz de probabilidad de riesgo

La matriz muestra la combinación de impacto y la probabilidad a su vez que producen una prioridad de riesgo que se muestra por los matices de color donde rojo equivale a prioridad 1, amarillo es prioridad 2 verde prioridad 3.

Probabilidad	Impacto				
	0.05	0.09	0.18	0.36	0.72
0.90	0.04	0.07	0.14	0.28	0.56
0.70	0.03	0.05	0.10	0.20	0.40
0.50	0.02	0.03	0.06	0.12	0.24
0.30	0.01	0.01	0.02	0.04	0.08
0.10	0.05	0.10	0.20	0.40	0.80

Cuadro 4.6: Matriz de probabilidad e impacto de riesgo

Es fundamental entender la prioridad para cada riesgo, ya que permite que el equipo del proyecto para comprender adecuadamente la importancia relativa de cada riesgo.

Análisis Cualitativo

El análisis del impacto del riesgo será cualitativo. El análisis cualitativo es una forma más rápida y por lo general más rentable de analizar los riesgos (en contraposición a un análisis cuantitativo).

El análisis debe llevarse a cabo con el objetivo de recoger datos sobre:

- La probabilidad de ocurrencia del riesgo.
- El impacto cualitativo en el proyecto.
- La calidad de los datos sobre el riesgo de ser.

Documentación

Los resultados del análisis de riesgos deben documentarse en el registro de riesgos. La siguiente información se hará constar en el Registro:

<i>Impacto del riesgo</i>	<i>Probabilidad del riesgo</i>	<i>Prioridad de Riesgo</i>	<i>Impacto cualitativo</i>	<i>Puntos matriz de riesgos - calculado por la hoja de registro después del impacto y probabilidad</i>

Plan de Respuesta

Durante la planificación de la respuesta, se consideran las siguientes estrategias de respuesta al riesgo las cuales aplicarán de acuerdo a la amenaza.

Estrategias de Riesgos

Hay varios métodos para responder a los riesgos.

Evitar

Evitar el riesgo implica cambiar los aspectos del plan general de gestión de proyectos para eliminar la amenaza, el aislamiento de los objetivos del proyecto o la relajación de los objetivos que se encuentran en peligro (por ejemplo, la ampliación del horario o reducir el ámbito de aplicación). Los riesgos que se identifican al principio del proyecto se pueden evitar mediante la aclaración de los requisitos, obtener más información, la mejora de las comunicaciones, o la obtención de experiencia.

Transferencia

Transferencia del riesgo implica cambiar el impacto negativo de una amenaza (y la propiedad de la respuesta) a un tercero. Transferencia del riesgo no elimina una amenaza, sino que simplemente hace que otra persona responsable de su gestión.

Mitigación

Mitigación de riesgos consiste en reducir la probabilidad y / o el impacto de la amenaza de riesgo a un nivel aceptable. Pasar a la acción temprana y proactiva frente a un riesgo es a menudo más eficaz que intentar reparar el daño que un riesgo dado cuenta ha causado. Desarrollo de planes de contingencia son ejemplos de mitigación de riesgos.

Aceptación

La aceptación se toma a menudo como una estrategia de riesgo, ya que es muy difícil de planificar las respuestas para cada riesgo identificado. Aceptación de riesgos debe normalmente sólo se da

por los riesgos de baja prioridad. Aceptación de riesgos puede ser pasiva, donde no se realiza ninguna acción en absoluto, o activa. El enfoque activo más común de aceptación del riesgo es el desarrollo de un coste y / o programa de.

Documentación

Los resultados de la planificación de respuesta se deben documentar en el registro de riesgos. La siguiente información se hará constar en el Registro:

Seguimiento y Control

Durante el seguimiento y control de riesgos identifican, analizan y planifican nuevos riesgos a través de juntas de revisión basadas en la retroalimentación de la hoja de riesgos donde se realizarán las siguientes tareas:

- Realizar un seguimiento de los riesgos identificados y controlar las condiciones de disparo
- Revisar la información de los resultados del proyecto (como progreso / estado de los informes , los problemas y las acciones correctivas)
- Volver a analizar los riesgos existentes para ver si la probabilidad, impacto, o el plan de respuesta adecuada ha cambiado
- Revisar la ejecución de las respuestas al riesgo y analizar su eficacia
- Asegúrese de que las políticas adecuadas de gestión de riesgos y se están utilizando procedimientos

Documentación

Los resultados del seguimiento y control de riesgos deben documentarse en el registro de riesgos. La siguiente información se hará constar en el Registro:

Estado - estados válidos son:

Estado (identificado / analizado / planificado / Activado / Resuelto / Retirado)	<i>Fecha de activación</i>	Notas o Comentarios

Los estados se definen bajo el siguiente esquema.

- o Identificado - Riesgo documentado, pero no se realizó un análisis
- o Analizado - Análisis de riesgo realizado, pero no lleva a cabo la planificación de respuestas
- o Planificado completo - la planificación completa de Respuesta
- o Activado - Se ha producido trigger riesgo y la amenaza se ha realizado
- o Resuelto - riesgo realizadas se ha contenido
- o Retirado - Riesgo Identificado ya no requiere la supervisión activa (por ejemplo, activación de riesgo ha pasado)

Los casos no resueltos o previstos serán expuestos en las juntas de control y análisis para determinar si generar modificación alguna a éste plan de riesgos.

VO.BO. PATROCINADOR DEL PROYECTO

Aprobado por Andrés Torres Barrera en su carácter de Director de la Compañía GarySystems SA de CV.

Firma: _____ Date: _____

5 — Procesos de desarrollo de software

Julio Ariel Hurtado¹,

*Julio Ariel Hurtado (ccollazo@unicauca.edu.co), Grupo IDIS de la Universidad del Cauca.
Colombia,*

a) Introducción

La industria de software viene representando una actividad económica de gran importancia en el mundo, ofrece múltiples fuentes de negocio y se perfila como una de las oportunidades económicas más grande para los países en desarrollo. El continuo avance tecnológico en las áreas de las tecnologías de la información, ha llevado al fortalecimiento de la industria de software, consolidándose como una de las principales actividades económicas; es tal el auge del software en la actualidad, que éste se encuentra en todos los ámbitos de la sociedad y es pieza fundamental del quehacer humano.

Debido al estado de la industria de software mundial, han surgido organizaciones que definen métodos, técnicas y prácticas asociadas al proceso de desarrollo de software y contribuyen al crecimiento continuo de esta industria, tal es el caso del Instituto de Ingeniería de Software de los Estados Unidos—SEI (Software Engineering Institute), el Instituto Europeo de Software—ESI (European Software Institute), entre los más representativos en este ámbito. En los países latinoamericanos la industria de software, aunque ya se ha empezado un proceso de maduración, las empresas del sector se enfrentan a una indudable desventaja, la falta de competitividad, en precios y calidad.

Los procesos juegan un papel fundamental en la calidad y en la gestión de proyectos, puesto que estos son la traza para la planificación, el seguimiento y control de los proyectos. Los procesos determinan las estrategias de gestión del proyecto, su calidad, así como los riesgos asociados al proyecto. La definición más básica de proceso lo define como un conjunto de fases sucesivas de un fenómeno natural o de una operación artificial [7]. En informática se manejan varias definiciones que aluden a diversos elementos: proceso puede ser simplemente una operación o conjunto combinado de operaciones con datos, o bien una secuencia de acontecimientos definida única y delimitada, que obedece a una intención operacional en condiciones predeterminadas. También se denomina proceso a una función que se está ejecutando [8].

b) Calidad de Software

La calidad engloba todo el proceso, de construcción de un producto y está determinada por factores directos e indirectos. Aunque calidad es un concepto amplio, complejo, multifacético,

subjetivo y muchas veces ambiguo, puesto que este puede describirse desde diferentes perspectivas. La ingeniería del software no escapa de esta realidad pero sin lugar a dudas se enfrenta con muchos obstáculos para poder dominar la calidad en comparación con otras ciencias. Para numerar solo algunos:

- **No existe una definición estándar y universal de qué es calidad:** En realidad algunos organismos e instituciones como ISO, IEEE, SEI, ESI, entre otros, brindan definiciones aceptables pero no son homogéneas, dando como resultado que cada profesional utilice su propia versión de calidad.
- **La calidad debe satisfacer a una amplia gama de entes relacionados pero no mutuamente excluyentes:** Clientes, procesos, organismos, productos (documentos, aplicaciones, mediciones...).
- **Crear cultura de calidad:** esto implica un compromiso constante, tedioso, costoso, y a largo plazo por parte de la organización y las personas que lo componen.

El término calidad es ambiguamente definido y pocas veces comprendido, esto se debe a que:

- La calidad no es una sola idea, es un concepto multidimensional.
- La dimensión de calidad incluye el interés de la entidad, el punto de vista de la entidad, y los atributos de la entidad.
- Por cada concepto existen diferentes niveles de abstracción.
- Varía para cada persona en particular.

Como ocurre con el concepto general de la calidad no existe una única definición de calidad del software, según la IEEE, la calidad debe ser medible y predecible. Los factores que la determinan son: la ausencia de defectos, la satisfacción de los usuarios y la conformidad con los requisitos. ISO define el concepto de calidad como: "Conjunto de características de una entidad (producto o servicio) que le confieren su aptitud para satisfacer necesidades expresadas e implícitas". Para asegurar la calidad del software, hay que tener en cuenta que se deben establecer como primera medida un **sistema de gestión de calidad**, el cual provea: gestión de la calidad, planificación de la calidad y definición de políticas de calidad. Además, para lograr el **aseguramiento de la calidad**, se debe hacer uso de técnicas de verificación y validación del software tales como la revisión e inspecciones de los productos de software que previenen la inserción de errores en el sistema, así como las prácticas de **control de calidad** tales como las pruebas de software que permiten controlar que el producto no llegue al cliente con problemas.

c) El Proceso de Software

Un proceso define *quién* está haciendo *qué, cuándo y cómo* alcanzar un determinado objetivo. En la ingeniería del software el objetivo es construir un producto software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad; captura y presenta las mejores prácticas que el estado actual de la tecnología permite [1]. De esta forma, un proceso de desarrollo de software debe ser capaz de evolucionar con los años. Durante esta evolución y en un momento de tiempo dado, deber limitar su alcance a las realidades que permitan las tecnologías, herramientas y participantes de una organización. Así, el proceso debe construirse sobre las tecnologías; los procesos y las herramientas deben desarrollarse en paralelo. Un creador del proceso debe limitar el conjunto de habilidades necesarias para que pueda ser ejecutado con las capacidades actuales de los miembros de la organización o del equipo. Además, el creador del proceso debe adaptar el proceso a las realidades del momento. Cada una de estas

cuatro circunstancias (*Tecnologías, Herramientas, Personas y Patrones de organización*) debe mantener el equilibrio del proceso de desarrollo de software, de modo que el proceso pueda evolucionar. La figura 5.1 representa el concepto de proceso de software.



Figura 5.1: El proceso de Software

Existen numerosas metodologías que inciden en distintas dimensiones del proceso de desarrollo. Por un lado tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán [14]. Ejemplo de estas metodologías son, el modelo en cascada, modelo de construcción de prototipos, desarrollo rápido (RAD), modelo V, modelo en espiral, modelo en espiral win-win, modelo de desarrollo concurrente, modelo de entrega incremental, entre otros [14]. Otra aproximación es centrarse en otras dimensiones, como por ejemplo, el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles (livianas), las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Algunas de las metodologías más sobresalientes de este tipo son: eXtreme Programming (XP)¹, Scrum², Evolutionary Project Management (Evo)³, Cristal Methods⁴, Agile Modeling⁵, entre otros.

Uno de los fines de la tecnología de los procesos de desarrollo de software es lograr que la representación de procesos pueda ser usada para conducir los actuales procesos de desarrollo y mantenimiento de software [22]. El concepto de representación del proceso ha evolucionado al

¹ <http://www.extremeprogramming.org/> – Página semi-oficial de XP

² <http://jeffsutherland.com/> – Sitio de Jeff Sutherland, con vínculos a Scrum y artículos sobre tecnologías ligadas a XML y SOAP.

³ <http://www.gilb.com/> – Es el sitio de Tom Gilb, creador de Evo. Dispone de una buena cantidad de documentos, gráficas y glosarios, incluyendo el manuscrito de 600 páginas de Competitive Engineering.

⁴ <http://alistair.cockburn.us/crystal/aboutmethodologies.html> – Página inicial de Crystal Methodologies, de Alistair Cockburn.

⁵ <http://www.agilemodeling.com/> – Sitio oficial de Agile Modeling de Scott Ambler.

punto de involucrar la noción de **Modelo del Proceso (MP)** que obedece a una representación abstracta de un conjunto de procesos que se especifican formalmente a partir del modelado de procesos. La existencia de los modelos de proceso permite en una gran medida aprovechar capacidades adicionales para completar procesos, automatizarlos, dirigirlos y aumentar su eficiencia, gracias a que sirve de base para lograr una comunicación fluida entre los procesos.

i) Modelos de Ciclos de Vida Clásicos

1. Modelo Cascada

El modelo de cascada se define como una secuencia de actividades a ser seguidas en orden, donde la estrategia principal es definir y seguir el progreso del desarrollo de software hacia puntos de revisión bien definidos. La Figura 5.2 muestra un diagrama conceptual del modelo describiendo el orden a seguir de las actividades del desarrollo de software. No se muestra una etapa explícita de “documentación” dado que ésta se llevaba a cabo durante el transcurso de todo el desarrollo.

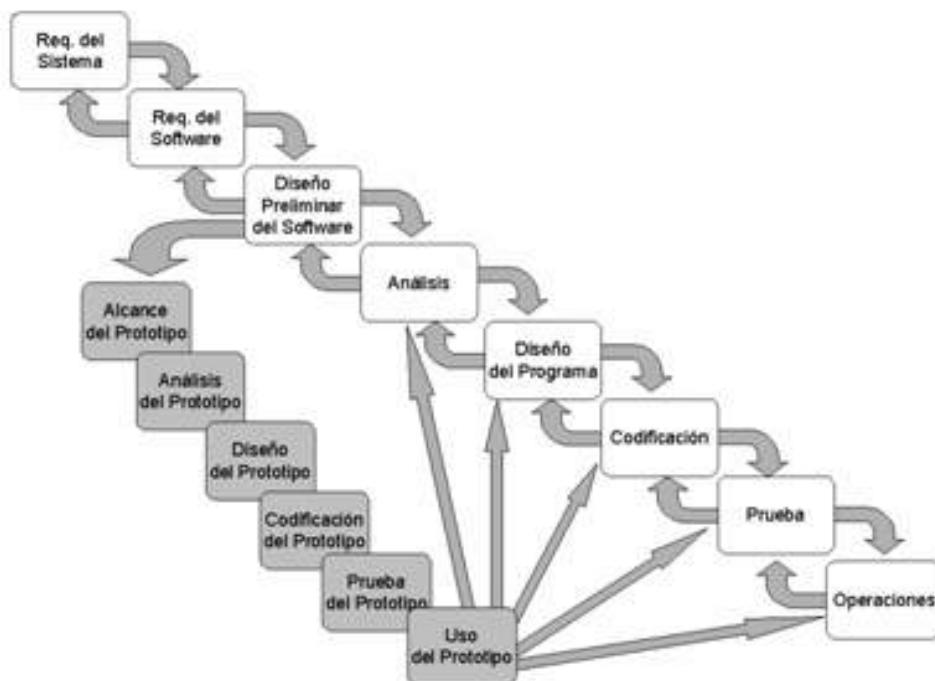


Figura 5.2: El Modelo en Cascada

Problemas presentados con el Modelo en Cascada:

- Los documentos a entregar rigen el proceso de software.
- Toma demasiado tiempo ver resultados.
- Depende de requisitos estables correctos.
- Hace difícil rastrear (ver la dependencia) de los requisitos iniciales y el código final.
- Retrasa la detección de errores hasta el final.
- No promueve la reutilización de software.
- No promueve el uso de prototipos.
- No se practica de manera formal.
- Funciona tan rápido como el más lento de los subprocesos

2. Modelo Espiral

El modelo de espiral se basa en una estrategia de reducción del riesgo, al contrario del modelo de cascada que es dirigido por documentos. El modelo enfatiza ciclos de trabajo, cada uno de los cuales estudia el riesgo antes de proceder al siguiente ciclo. Cada ciclo comienza con la identificación de los objetivos para una parte del producto, formas alternativas de lograr los objetivos, restricciones asociadas con cada alternativa, y finalmente procediendo a una evaluación de las alternativas. Cuando se identifica incertidumbre, se utilizan diversas técnicas para reducir el riesgo en escoger entre las diferentes alternativas. Cada ciclo del modelo de espiral termina con una revisión que discute los logros actuales y los planes para el siguiente ciclo, con el propósito de lograr la incorporación de todos los miembros del grupo para su continuación. La revisión puede determinar si desarrollos posteriores no van a satisfacer las metas definidas y los objetivos del proyecto. En tal caso, se terminaría el espiral.

La Figura 5.3 muestra un diagrama conceptual del modelo de espiral describiendo los distintos ciclos del espiral. Nuevamente, no se muestra una etapa explícita de “documentación” dado que ésta se llevaba a cabo durante el transcurso de todo el desarrollo [30].

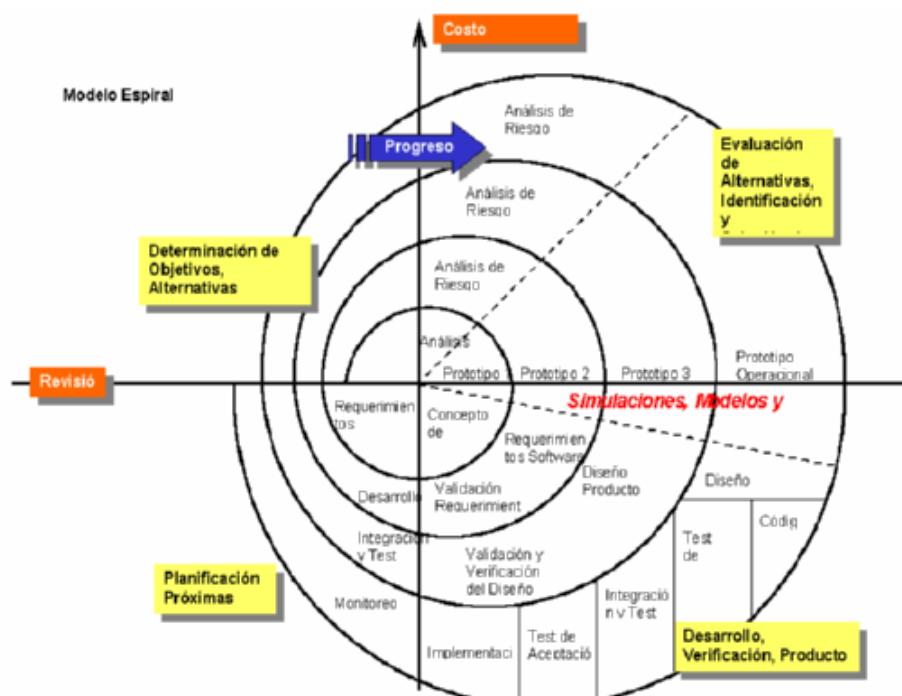


Figura 5.3: El Modelo en Espiral

3. Modelo de Construcción de prototipos

Dos de las críticas que se hacían al modelo de ciclo de vida en cascada consistían en la dificultad para tener claros todos los requisitos del sistema al inicio del proyecto, y que no se dispone

de una versión operativa del programa hasta las fases finales del desarrollo, lo que dificulta la detección de errores y deja también para el final el descubrimiento de los requisitos inadvertidos en las fases de análisis. Para pulir estas deficiencias se ha propuesto un modelo de ciclo de vida basado en la construcción de prototipos.

En general, cualquier aplicación que presente mucha interacción con el usuario, o que necesite algoritmos que puedan construirse de manera evolutiva, de lo más general a lo más específico es un buen candidato a utilizar el modelo de ciclo de vida de construcción de prototipos. No obstante, hay que tener en cuenta la complejidad: si la aplicación necesita que se desarrolle una gran cantidad de código para poder tener un prototipo que enseñar al usuario, las ventajas de la construcción de prototipos se verán superadas por el esfuerzo de desarrollar un prototipo que al final habrá que desechar o modificar mucho.

En otros casos, el prototipo servirá para modelar y poder mostrar al cliente cómo va a realizarse la entrada y salida de datos en la aplicación, de forma que éste pueda hacerse una idea de como va a ser el sistema final, pudiendo entonces detectar deficiencias o errores en la especificación aunque el modelo no sea más que una cáscara vacía [31].

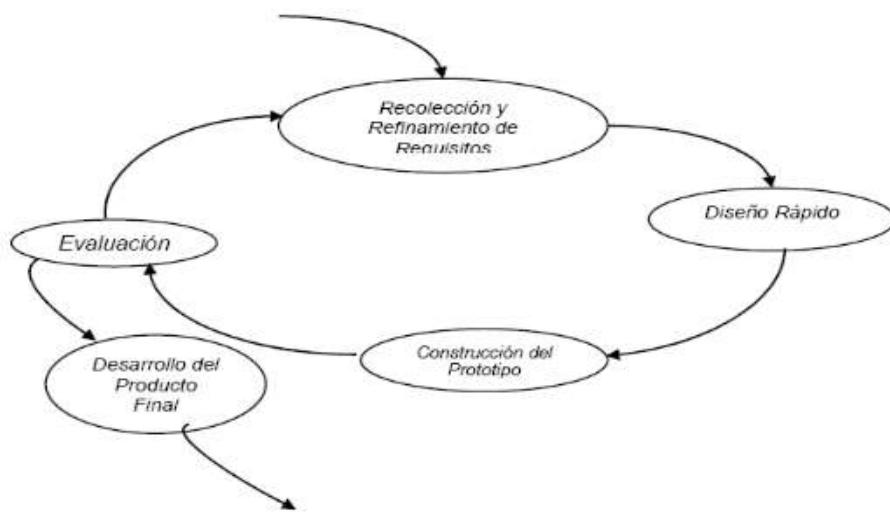


Figura 5.4: El Modelo de Construcción de Prototipos

4. Desarrollo Rápido de Aplicaciones – DRA

Es una adaptación a “alta velocidad” del modelo en cascada en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Puede permitir el desarrollo de un sistema completamente funcional en períodos cortos de tiempo (de 60 a 90 días). Los componentes que se desarrollen se pueden reutilizar en posteriores proyectos (Repositorio de componentes). El sistema se descompone en un conjunto de bloques que se pueden desarrollar de manera independiente por distintos equipos de desarrollo.

Sólo puede aplicarse cuando se cumplen una serie de condiciones:

- Se comprenden muy bien los requisitos del sistema a desarrollar. Ya sea porque los conoce el propio desarrollador o porque se tiene una experiencia previa en un sistema similar.

- Se delimita muy bien el ámbito del problema.
- La interacción del software con el nuevo sistema no es complicada o se utilizan nuevas tecnologías que no son dominadas por el equipo de desarrollo.

Inconvenientes

- Debe haber un compromiso por parte del equipo de desarrollo y del cliente en el desarrollo rápido de actividades.
- Requiere recursos suficientes para crear el número de equipos necesarios [31] .

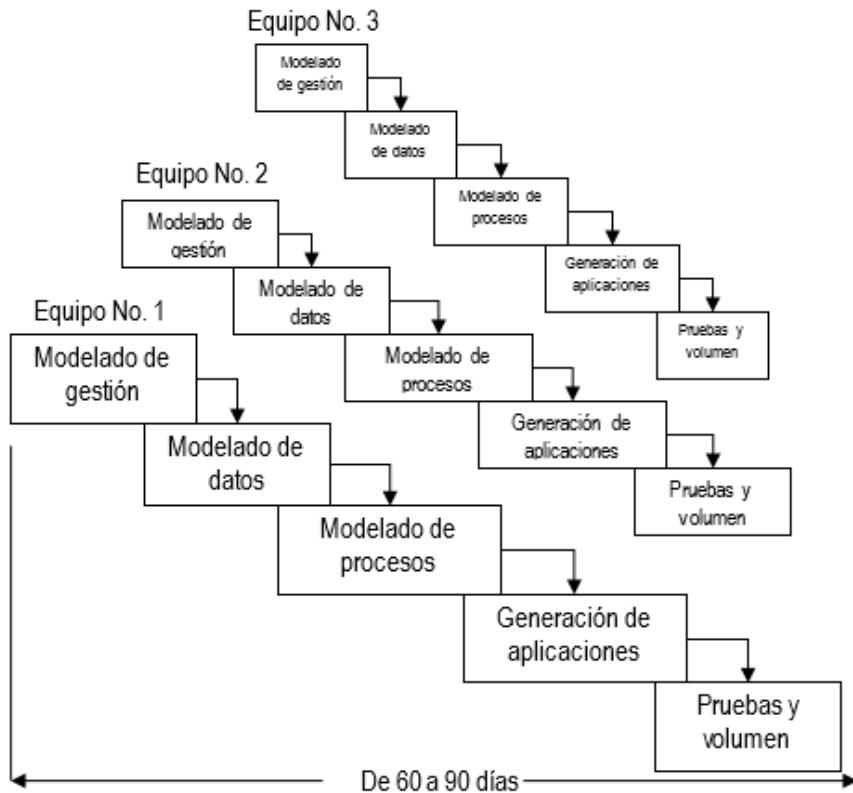


Figura 5.5: El Modelo DRA

5. Modelo en V

Muchos de los modelos de procesos comúnmente usados son por lo general conectados por medio de un modelo-V donde la “V” describe la configuración gráfica de las fases individuales. La “V” es también un sinónimo de verificación y validación. El modelo es muy simple y fácil de entender, por el ordenamiento de actividades en secuencias de tiempo y con niveles de abstracción de conexión entre actividades de desarrollo y pruebas llegan a ser claras. Del lado opuesto, las actividades se colocan una sobre la otra, es decir, sirven como base para las actividades de prueba. Por ejemplo, las pruebas de sistema son llevadas a cabo con base en los resultados de la fase de especificación. La Figura 5.6 muestra el esquema general del modelo en V [33].

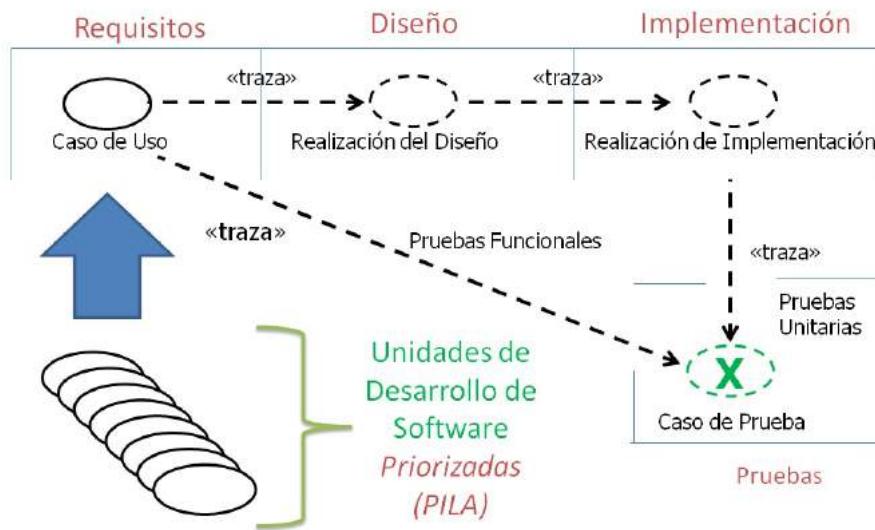


Figura 5.6: El Modelo “V”

ii) Un modelo de proceso basado en un enfoque clásico

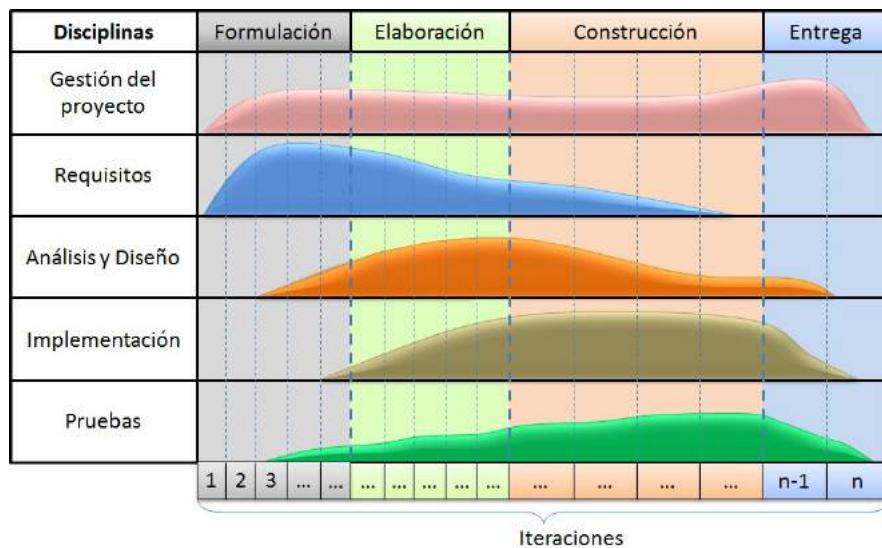
El Proceso Unificado (Unified Process - UP) está basado en una integración del trabajo de tres metodólogos, Ivar Jacobson, Grady Booch y James Rumbaugh. Estos autores, formaron un marco de metodologías unificadas, cohesivas y comprehensivas de desarrollo de sistemas de software. Su trabajo, que producen durante varios años y basados en metodologías probadas, han dado a lugar a importantes normas en la comunidad de desarrollo, incluida la aceptación general de los casos de uso y del Lenguaje de Modelado Unificado (Unified Modeling Language – UML). El Proceso Unificado tiene tres características distintivas:

1. **Dirigido por Casos de Uso:** el proceso utiliza los casos de uso para guiar el proceso de desarrollo desde la Incepción hasta el Despliegue. Los casos de uso son priorizados y usados como unidades de desarrollo. Así, en una iteración un conjunto de casos de uso son realizados a diferentes niveles de abstracción (análisis, diseño, implementación y pruebas). De esta forma grupos de requerimientos son convertidos en sistema funcional que puede incrementalmente ser evaluado por el cliente.
2. **Centrado en Arquitectura:** el proceso busca entender los aspectos estáticos y dinámicos más significativos en términos de arquitectura de software. La arquitectura se define en función de las necesidades de los usuarios y se determina a partir de los casos de uso críticos del negocio, que esconden un reto técnico o que simplemente tienen asociados riesgos del producto.
3. **Iterativo e Incremental:** el proceso reconoce que es práctico dividir grandes proyectos en proyectos más pequeños o mini-proyectos. Cada mini-proyecto comprende una iteración que resulta en un incremento. Una iteración puede abarcar la totalidad de los flujos del proceso. Las iteraciones son planificadas y ejecutadas utilizando como unidad el casos de uso y sus realizaciones.



El proceso Unificado consta de cuatro fases: inicio, elaboración, construcción y transición:

1. **Fase de Inicio:** durante la fase inicial se tiene la idea central del producto, se arma el **documento de visión**. En esta fase, se revisan y confirma el entendimiento sobre los objetivos centrales del negocio, se entienden los argumentos comerciales para determinar la viabilidad del proyecto. La fase de inicio establece la viabilidad del producto y delimita el alcance del proyecto.
2. **Fase de elaboración:** durante la fase de elaboración los casos de uso de mayor impacto arquitectónico son especificados en detalle y la arquitectura del sistema es diseñada. El objetivo de esta fase es determinar la viabilidad del proyecto. Se identifican los riesgos significativos y se preparan el plan, el equipo de trabajo y el costo del proyecto.
3. **Fase de construcción:** durante la fase de construcción, el foco del producto se mueve de la arquitectura base a un sistema lo suficientemente completo como para llevarlo al usuario. La línea base de arquitectura crece en complejidad y se convierte en un sistema completo, de la misma manera, se refina el diseño para llevarlo a código fuente.
4. **Fase de Transición:** en la fase de transición el objetivo es garantizar que los requisitos se han cumplido, con la satisfacción de las partes interesadas. Esta fase a menudo se inicia con una versión beta de la aplicación. Otras actividades incluyen la preparación del ambiente, se completan, se identifican y corrigen defectos. La fase de transición termina con un cierre dedicado a las lecciones aprendidas, las cuales quedan para futuros ciclos.



El Proceso Unificado identifica a los flujos de trabajo fundamentales que se producen durante el proceso de desarrollo de software. Estos flujos se refieren a los cuerpos de conocimiento más importantes del proceso, los cuales son identificados como disciplinas: el modelado de negocio, requerimientos, análisis, diseño, implementación y testing. Los flujos no son secuenciales y serán realizados en diferente intensidad durante las cuatro fases. Los flujos son descritos separadamente en el proceso por claridad, pero pueden ser ejecutados en forma secuencial o concurrente, interactuando y utilizando los artefactos que cada uno genera. A continuación se presentan brevemente los flujos.

1. **Requerimientos:** su propósito es lograr el desarrollo del sistema correcto, es decir los requisitos que el sistema debe satisfacer. Esto se lleva a cabo mediante la descripción de los requisitos del sistema de forma tal que se pueda llegar a un acuerdo entre el cliente, otros participantes y los desarrolladores del sistema, acerca de lo que el sistema debe hacer y acerca de lo que no.
2. **Análisis:** su propósito es analizar los requisitos descritos en la captura de requisitos, mediante su refinamiento y estructuración. El objetivo es lograr una comprensión más precisa de los requisitos, obtener una descripción de los requisitos que sea fácil de mantener y organizar la estructura física y lógica del sistema.
3. **Diseño:** su propósito principal es obtener los modelos que se centran en los requisitos funcionales y no funcionales, más orientado al dominio de la solución (tecnología) y que sirven de base para la implementación y pruebas del sistema.
4. **Implementación:** el propósito es implementar el sistema en términos de componentes, es decir código fuente, archivos binarios, archivos de configuración, ejecutables, entre otros.
5. **Prueba:** el propósito es determinar si la implementación satisface los requerimientos y las necesidades del cliente, mediante las pruebas en cada construcción(a cada iteración), así como las versiones finales del sistema que van a ser entregadas al cliente.

iii) Metodologías Ágiles

Los métodos ágiles corresponden a un tipo de enfoque de ingeniería de software donde el proceso de software no es descrito en forma descriptiva, ni prescriptiva, sino que obedece a las características del proyecto y del equipo, de acuerdo a la aplicación de un conjunto de reglas individuales y colectivas. En las metodologías ágiles, la agilidad se refiere a la habilidad para bien sea crear o responder a los cambios con el fin de cubrir un ambiente de negocio inestable.

En estos métodos, la agilidad se refiere a la habilidad, bien sea para crear o responder a los cambios con el fin de cubrir un ambiente de negocio inestable. Así la agilidad, denota la cualidad de ser ágil, facilidad para moverse hacia la meta. Durante algún tiempo se les dio el nombre de metodologías ligeras, pero el término finalmente acuñado es el de métodos ágiles [49]. Las metodologías ágiles están motivadas por la necesidad de resolver la aguda crisis del software, y como una revolución a los enfoques centrados en el proceso. El término ágil nace en febrero del 2001, tras una reunión realizada en Utah, Estados Unidos, en la que se creó “The Agile Alliance”, La Alianza Ágil, una organización dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten sus conceptos. En esta reunión se construyó un documento que resume la filosofía ágil denominado el Manifiesto Ágil. Este documento describe los métodos ágiles a partir de un conjunto de principios y valores.

1. Programación Extrema - XP

La programación extrema es un conjunto de normas y recomendaciones encaminadas a producir software de calidad. XP fue diseñado teniendo en cuenta los problemas que existían con las metodologías tradicionales de programación en cuanto a tiempos de entrega y satisfacción del cliente. El objetivo principal para XP es buscar la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto. XP está diseñado para proyectos no muy grandes pero cuyos requerimientos cambien constantemente o que sean de alto riesgo [29].

Según algunos Programación Extrema es una forma deliberada y disciplinada de desarrollar software [29]. Otros aseguran que XP es una metodología ágil, la cual consiste básicamente en un conjunto de guías generales para el desarrollo de software por equipos de pequeño y mediano tamaño. La programación extrema está fundamentada en cinco bases principales que son:

- *Trabajo en equipo:* El trabajo en equipo es fundamental para sacar un proyecto adelante. XP sugiere grupos no muy grandes (entre 2 y 12 desarrolladores) para manejar mejor los problemas de requerimientos cambiantes. Es necesario que todos los miembros del grupo estén comprometidos con todo el software producido. XP acepta que uno de los principales y más importantes factores en un equipo de desarrollo son los integrantes del equipo en sí.
- *Participación del cliente:* Para que el cliente pueda quedar satisfecho al final de un proyecto de software debe ser parte activa del equipo. Un cliente no puede entregar los requerimientos, irse y luego volver a mirar los resultados. Involucrar al cliente no significa que el esté al tanto de los detalles mínimos de implementación, simplemente la idea es que el cliente (idealmente un representante) este disponible para responder a dudas o preguntas que surjan dentro de los ciclos de desarrollo.
- *Simplicidad:* Se cree que manteniendo tanto el diseño como el código lo más simple posi-

ble se logra una reducción muy importante en los costos de mantenimiento. Por lo tanto se pide al desarrollador ante todo simplicidad a la hora de diseñar o implementar. Es tal la importancia de la simplicidad que si se encuentra una forma más sencilla de implementar algo después de hecho, uno debería considerar seriamente volver a implementarlo. Esto puede sonar un poco anti-intuitivo, pero en realidad el tiempo ahorrado por simplificar algo a lo que se le va a hacer mantenimiento debe considerarse como una inversión y no un gasto.

- *Buenas pruebas:* Con XP se busca eliminar malas prácticas como probar todo el software en los últimos días antes de salir el proyecto. Para esto se propone un conjunto de normas cuyo objetivo es lograr un muy buen diseño de pruebas y una ejecución completa y eficiente. En realidad XP introduce el concepto de pruebas automatizadas las cuales ayudan a desarrollar mejores aplicaciones. Las pruebas automatizadas proveen al equipo de desarrollo una garantía de que los cambios hechos durante el desarrollo no afectan a otros componentes de la aplicación, o si los afectan, poder ubicar las partes comprometidas fácilmente.
- *La productividad es secundaria:* La satisfacción del cliente debe ser el primer objetivo. Idealmente XP mejora la productividad de los desarrolladores pero si en la realidad la productividad baja, no debe importar mientras que se logre la satisfacción del cliente.

XP está catalogada como una metodología ligera pues se cree que metodologías como RUP tienen una carga excesiva de documentación que en una buena parte de los casos nunca se utiliza y que hace que el desarrollador no se enfoque en el código que es lo más importante. XP se puede ver como una selección de unas pocas reglas de otras metodologías y la inclusión de algunas nuevas que resuelve el problema anterior sin perder calidad en el software.

Así como RUP, XP trabaja por medio de ciclos o iteraciones de desarrollo. Un ciclo o iteración es un periodo de tiempo en el que se realiza un conjunto de funcionalidades determinadas que en el caso de XP corresponden a un conjunto de historias de usuario. Los ciclos aquí son particularmente cortos ya que se piensa que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial encaminada a programar los ciclos de desarrollo y cada ciclo contiene diseño, codificación y pruebas, con la diferencia de que no son etapas separadas sino que están superpuestas de tal manera que no es correcto separar un proyecto XP en estas fases porque el esquema de trabajo no se divide en el tiempo de esta forma, cosa que si sucede en RUP.

2. Scrum

Esta es, después de XP, la metodología ágil mejor conocida y la que otros métodos ágiles recomiendan como complemento, aunque su porción del mercado (3 % según el Cutter Consortium) es más modesta que el ruido que hace. Como metodología ágil específicamente referida a ingeniería de software, Scrum fue aplicado por Jeff Sutherland y elaborado más formalizadamente por Ken Schwaber. Poco después Sutherland y Schwaber se unieron para refinar y extender Scrum, en el que se aplicaron principios de procesos de control industrial, junto con experiencias metodológicas de Microsoft, Borland y Hewlett-Packard. Schwaber, en particular, había trabajado con científicos de Du Pont para comprender mejor los procesos definidos de antemano, y ellos le dijeron que a pesar que los modelos de madurez se concentraba en hacer que los procesos de desarrollo se tornaran repetibles, definidos y predecibles, muchos de ellos eran

formalmente impredecibles e irrepetibles porque cuando se está planificando no hay primeros principios aplicables, los procesos recién comienzan a ser comprendidos y son complejos por naturaleza. Schwaber se dio cuenta entonces de que un proceso necesita aceptar el cambio, en lugar de esperar predictibilidad [15].

Al igual que Agile Modeling, Scrum no está concebido como método independiente, sino que se promueve como complemento de otras metodologías, incluyendo XP, MSF o RUP. Como método, Scrum enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; de allí su deliberada insuficiencia y su complementariedad. Scrum se define como un proceso de gestión y control que implementa técnicas de control de procesos; se lo puede considerar un conjunto de patrones organizacionales. Los valores de Scrum son:

- ***Equipos auto-dirigidos y auto-organizados.*** No hay manager que decida, ni otros títulos que “miembros del equipo”.
- ***Una vez elegida una tarea, no se agrega trabajo extra.*** En caso que se agregue algo, se recomienda quitar alguna otra cosa.
- ***Encuentros diarios con las tres preguntas diarias (que se hizo, que falta, que problemas han surgido).*** Se realizan siempre en el mismo lugar, en círculo.
- ***Iteraciones de tres semanas*** se admite que sean más frecuentes.
- Demostración a participantes externos al fin de cada iteración.
- Al principio de cada iteración, planeamiento adaptativo guiado por el cliente.

Scrum define seis roles:

1. ***El Scrum Master.*** Interactúa con el cliente y el equipo. Es responsable de asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de Scrum y que progrese según lo previsto. Coordina los encuentros diarios, formula las tres preguntas canónicas y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par.
2. ***Propietario del Proyecto.*** Es el responsable oficial del proyecto, gestión, control y visibilidad de la lista de acumulación o lista de retraso del producto (product backlog). Es elegido por el Scrum Master, el cliente y los ejecutivos a cargo. Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.
3. ***Equipo de Scrum.*** Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos. El equipo posee la misma estructura del “equipo quirúrgico” desarrollado por IBM aunque se destaca que su naturaleza auto-organizadora la hace distinta.
4. ***Cliente.*** Participa en las tareas relacionadas con los ítems del registro.
5. ***Management.*** Está a cargo de las decisiones fundamentales y participa en la definición de los objetivos y requerimientos. Por ejemplo, selecciona al Dueño del Producto, evalúa el progreso y reduce el registro de acumulación junto con el Scrum Master.

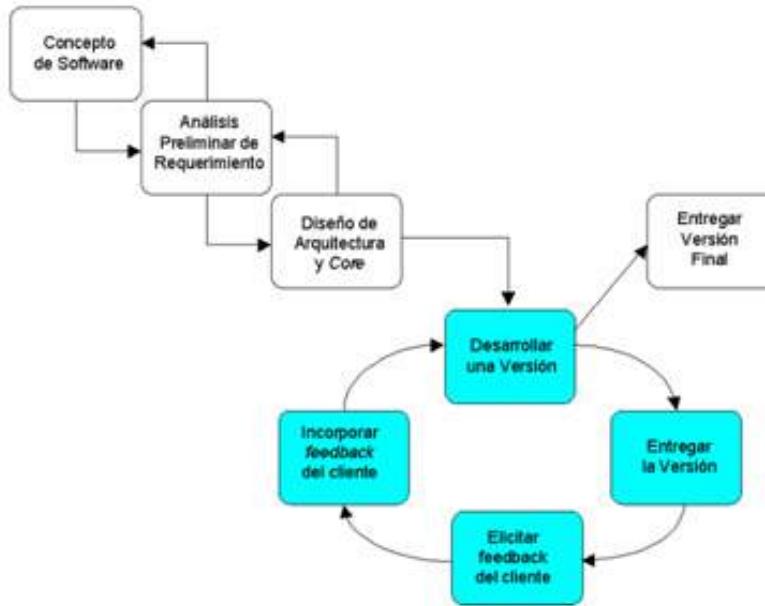


Figura 5.7: Modelo de entrega evolutiva

En proyectos evolutivos, las Metas se desarrollan tratando de comprender de quiénes vienen (Participantes), qué es lo que son (medios y fines) y cómo expresarlas (cuantificables, medibles y verificables). Se procura pasar el menor tiempo posible en tareas de documentación. En las formas más simples y relajadas de Entrega Evolutiva, a veces llamada Entrega Incremental, liberar parte de una solución es un Paso de Entrega Evolutiva. En la Entrega Evolutiva más pura y más rica, sólo las mejoras en las Metas de los Participantes se consideran un Paso de Entrega Evolutiva. Hay que distinguir bien, asimismo, entre los Medios y los Fines, por un lado, y las Metas y las Soluciones por el otro. Las Metas deben separarse de las Soluciones; las Metas deben ser sagradas, y deben alcanzarse por cualquier medio; las Soluciones son sólo posibles, contingentes: son caballos de trabajo, y deben cambiarse cuando se consigue un caballo mejor [15].

A diferencia de otras MAs que son más experimentales y que no tienen mucho respaldo de casos sistemáticamente documentados, Evo es una metodología probada desde hace mucho tiempo en numerosos clientes corporativos: la NASA, Lockheed Martin, Hewlett Packard, Douglas Aircraft, la Marina británica. El estándar MIL-STD-498 del Departamento de Defensa y su correspondiente estándar civil IEEE doc 12207 homologan el uso del modelo de entrega evolutiva. Sólo DSDM y RUP han logrado un reconocimiento comparable [15].

iv) Modelado de Procesos

El Modelado de Procesos es un paso fundamental para la comprensión y mejora continua de los procesos de una organización. El modelado del proceso software trata de capturar las características principales del proceso que se lleva a cabo, identificando las actividades necesarias, los responsables de llevarlas a cabo, los productos obtenidos, etc.[22].

Las metodologías de modelado de procesos principalmente se dividen en tres categorías: basadas en comunicaciones, basadas en artefactos y basadas en actividades [9] .

1. Modelado basado en comunicaciones

Este tipo de metodologías representa una acción en una comunicación que se ha establecido entre un cliente y un proveedor. Esta comunicación consiste en 4 fases: **Petición**, cuando un cliente requiere que se lleve a cabo una acción; **Negociación**, el cliente y proveedor se ponen de acuerdo en la acción que ha de ser ejecutada; **Realización**, Fase en la que la acción a llevar a cabo es ejecutada; y **Aceptación**, donde el cliente informa de la conformidad con la acción realizada por el proveedor.

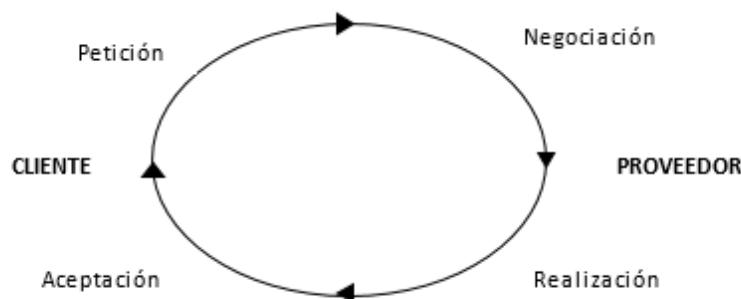


Figura 5.8: Modelo basado en comunicación

2. Modelado basado en artefactos

La aproximación basada en artefactos se centra en los objetos (artefactos) que son creados, modificados y utilizados durante un proceso (ver Figura No. 12 Modelo basado en artefactos). Esto supone que el modelado de un proceso está basado en la creación de los documentos generados durante él mismo y su camino a través de una serie de actividades del flujo de trabajo.

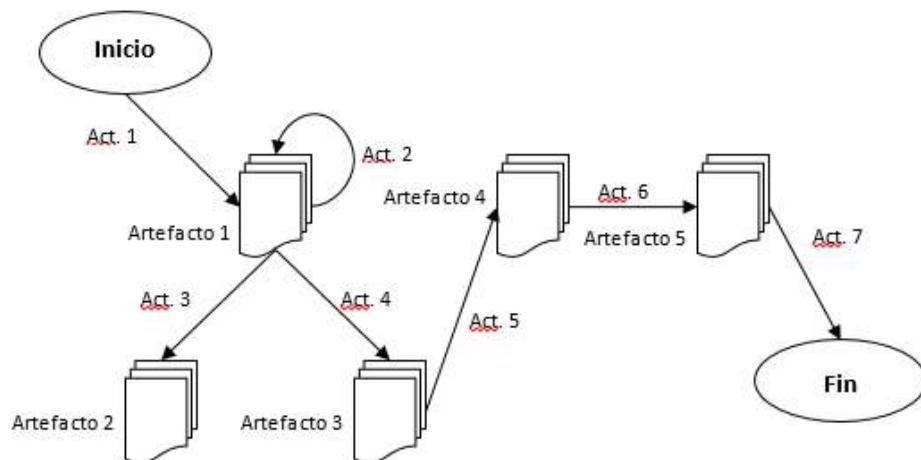


Figura 5.9: Modelo basado en artefactos

3. Modelado basado en actividades

Las metodologías basadas en las actividades se centran en el modelo de trabajo, en contraste con los casos previos donde los acuerdos establecidos entre los participantes son las bases del modelado de procesos. Esta aproximación está basada en la descomposición de procesos en tareas que son ordenadas de acuerdo con las dependencias existentes entre ellas. La principal ventaja reside en que este modelo puede ser fácilmente convertido en la especificación de un workflow del cual se dará una breve explicación más adelante. Además, este tipo de modelado es susceptible de ser representado mediante un lenguaje orientado a objetos.

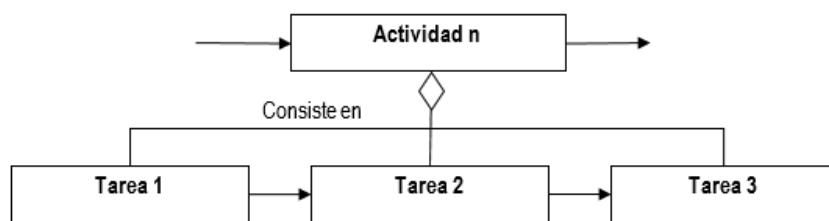


Figura 5.10: Modelo basado en actividades

v) Lenguajes de Modelado de Procesos – LMP^s⁶

Los procesos de software al igual que cualquier entidad abstracta requieren de un medio o plataforma para poderlos expresar –El Modelo de Procesos (MP)-, estos a su vez, para su entendimiento y facilidad de interpretación requieren de la utilización de una serie de reglas sintácticas y semánticas para describir sus elementos, como por ejemplo, roles, actividades, recursos, etc. Un LMP permite hacer la transición del proceso a modelos de proceso. Los elementos primarios de un proceso que debe ser capaz de modelar un LMP son: Actividades, productos, roles, personas, herramientas y soporte para la evolución. Un LMP debe ofrecer soporte para la evolución del modelo de procesos tanto a nivel técnico como a nivel conceptual, este último mediante un modelo asociado. Además de los elementos primarios los LMPs incluyen la posibilidad de representar elementos propios del meta proceso, entre los cuales podemos mencionar:

proyectos/organizaciones, contextos de trabajo, vistas de usuario, etc. [25]. Los lenguajes de modelado de procesos se pueden dividir en tres categorías:

Lenguajes Formales que tienen sintaxis y semántica formales y, por tanto, proveen soporte para verificación y análisis formales, simulación y ejecución. Este tipo de lenguajes utiliza diferentes tipos de connotaciones lingüísticas y paradigmas entre las cuales encontramos: lenguajes *basados en reglas, basados en estados, funcionales, procedurales, orientados a objetos, entre otros*. Algunos ejemplos de LMPs formales son: basados en reglas: Marvel, Oz, Atlantis; Orientados a Objetos: E3, EPOS/Spell, Lenguajes de Programación: APPL/A, JIL, Grafos y gramáticas: Hakoniwa, Redes de Petri; SLANG/SPADEF [27].

Lenguajes Semiformales están provistos de una notación formal (normalmente gráfica) pero no tienen semántica formal. Esto último hace que no sean ejecutables. Por ejemplo el Lenguaje de

⁶ En algunos textos este término se encuentra como PML (Process Modeling Language) Por sus siglas en Inglés

Modelado Unificado UML, y el RUP que se fundamenta en la utilización de UML para modelar visualmente los procesos. En esta misma línea podemos catalogar a SPEM 2.0 puesto que al igual que RUP se deriva de UML.

Lenguajes Informales no tienen sintaxis ni semánticas formales (por ejemplo, el lenguaje natural).

vi) Evaluación de Procesos

La evaluación de los procesos software tiene como objetivo detectar los aspectos de un proceso software que se pueden mejorar. Para ello es necesario proporcionar un marco efectivo para la medición de los procesos y productos software en una organización.

La integración de esta área con el modelado de procesos es un factor fundamental para que una organización alcance un alto grado de madurez en sus procesos tal como identifican diversos estándares entre los que destacan especialmente: CMM, ISO 15504 y sobre todo CMMI. Por ello, es imprescindible comprender bien los procesos, para lo cual es necesaria una definición de los mismos, y mejorarlos, (lo que incluye un proceso de evaluación). El modelado del proceso software ha recibido mucha atención de la comunidad científica en los últimos años, sin embargo, y teniendo en cuenta que los modelos de procesos son el punto de partida para el análisis, mejora y ejecución del proceso, la necesidad de establecer un acoplamiento estricto entre el modelado y la medición del proceso no ha sido claramente establecido.

Para que una organización pueda realizar una gestión integrada de sus procesos software es muy importante que establezca una base rigurosa para:

- La definición de sus modelos de procesos con una terminología única y con una semántica precisa y bien definida.
- La gestión integrada de la medición en la organización mediante un meta.

Obtener el modelo de la medida que sea el marco de referencia para la creación de modelos concretos de medida (medida de la base de datos, medidas de los productos de trabajo resultado del análisis o diseño, medida de los modelos de procesos, etc.).

REFERENCIAS BIBLIOGRÁFICAS

1. JACOBSON. Ivar. BOOCH. Grady. RUMBAUGH. James. “El Proceso Unificado de Desarrollo de Software”. Edición en Español. Ed. Addison Wesley. 2000.
2. HOLLINGSWORTH. David. “Workflow Management Coalition the Workflow Reference Model”. 55 p. Enero de 1995
3. OBJECT MANAGMENT GROUP. “Software Process Engineering Metamodel Specification” Versión 1.0. Noviembre de 2002.
4. HURTADO. Julio Ariel. “Sistema Integral para el Mejoramiento de los Procesos de Desarrollo de Software en Colombia (SIMEP-SW)”. Departamento de Sistemas, Universidad del Cauca. Popayán, Julio de 2003.
5. HURTADO. Julio Ariel. “El modelo integral de mejoramiento Agile SPI”. Departamento de Sistemas, Universidad del Cauca. Popayán, Agosto de 2004.
6. FUGGETA. Alfonso. “SOFTWARE PROCESS: A Roadmap”. Dipartimento di Elettronica e Informazione. Politecnico di Milano. 1999. <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalfuggetta.pdf>
7. REAL ACADEMIA DE LA LENGUA ESPAÑOLA. Diccionario de la Lengua Española. Vigésima segunda edición. 2004. <http://www.rae.es/>
8. UNIVERSIDAD NACIONAL DE SAN JUAN. “Glosario Informático”. Argentina 2004. <http://adig.com.ar/cg3/glosario/>
9. UNIVERSIDAD CARLOS III DE MADRID. “Modelado de Procesos”. <http://www.ie.inf.uc3m.es/Tesis/Documentos/ProcesosSoftware/documentos.htm>
10. HERNÁNDEZ Muñoz. José Luís. “Rentabilidad del desarrollo de proyectos de cómputo”. México 2002. <http://www.monografias.com/trabajos13/renta/renta.shtml>
11. BEDINI G. Alejandro. “Calidad tradicional y de software”. Departamento de Industrias, Universidad Técnica Federico Santa María, Chile. <http://www.willydev.net/descargas/Articulos/General/CalidadSoftware.pdf>
12. BUSINESS SOFTWARE ALLIANCE. “Industria del software podría crecer 645% hasta el año 2005”. Febrero 2003. <http://global.bsa.org/colombia/press/newsreleases/2003-02-21.1476.phtml>
13. BERTOA. Manuel F, TROYA. José M y VALLECILLO. Antonio “Atributos de Calidad para Componentes COTS: Una valoración de la información ofrecida por los vendedores”. <http://www.lcc.uma.es/~{}av/Publicaciones/03/TICS03.pdf>
14. LETELIER. Patricio y PENADÉS. Carmen. “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”. Facultad de Informática. Universidad Politécnica de Valencia. 2004 <http://www.willydev.net/descargas/masyxp.pdf>

-
15. REYNOSO. Carlos. “Métodos heterodoxos en desarrollo de software”. http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.asp
 16. MAMANI. Luís. “Sistema Informático para el Seguimiento y Supervisión de Proyectos de Desarrollo de Software”. Universidad Peruana de Ciencias Aplicadas. Perú 2002. <http://espejos.unesco.org.simplac2002/Ponencias/Infoedu/IE020%20Luis%20Mamani.doc>
 17. OKTAVA. Hanna. “Modelo de Procesos para la Industria de Software MoProSoft” Versión 1.1. México 2003. <http://www.lania.mx/biblioteca/manuals/moprossoft/V%201.1%20DocumentoBase.pdf>
 18. HERNÁNDEZ. Orallo. Enrique. “El lenguaje Unificado de Modelado (UML)” Departamento de informática y computadores. UPV. 2000. <http://dark.disca.upv.es/enheror/pdf/ActaUML.pdf>
 19. BPMWin. “What is Business Process Management (BPM)?”. 2004. <http://www.bpmtutorial.com/BPM/BPM.aspx>
 20. SQS-SA. “SQS – RequirementsWORKFLOW. Herramienta que implementa eXtreme Programming para la gestión de requisitos”. 2004. <http://www.sqs.es/documentos/eXtreme.pdf>
 21. PIKATZA. Juan M. “Modelo jerárquico de roles para organizaciones de pequeño tamaño”. Departamento de Lenguajes y Sistemas Informáticos. UPV/EHU. 2002. <http://www.sc.ehu.es/jiwdocoj/remis/adis2002.htm>
 22. GARCÍA, Felix. “Gestión Integrada de modelado y de la medición del proceso software”. Grupo Alarcos. Escuela Superior de Informática. Universidad de Castilla – La Mancha. 2002. <http://lsi.ugr.es/~{}gedes/actividades/Dolmen4/a5.pdf>
 23. ABIAN. Miguel Ángel. “J2EE Y .NET: LA RIVALIDAD PERMANENTE”. <http://www.javahispano.org/articles.article.action?id=55>
 24. MAÑAS. José A. “Pruebas de programas”. <http://www.upv.es/protel/usr/jotrofer/pascal/testing.htm>
 25. RUIZ G. Francisco. “MANTIS: Definición de un Entorno para la Gestión del Mantenimiento de Software”. Departamento de Informática. Universidad de Castilla – La Mancha. Junio de 2003. Documento Digital.
 26. PUMAREJO. Johanna. “Descripción del sector del software - análisis de mercado”. Departamento de Inteligencia de Mercados. FEDESOFT 2002. Documento Digital.
 27. RUIZ. Francisco. “Mantenimiento Avanzado de Sistemas de Información”. Universidad de Castilla – La Mancha. Ciudad Real. España 2004. Documento Digital.
 28. GRUPO DE MEJORA DEL PROCESO DE SOFTWARE. “Resumen del libro Workflow and Process Automation”. Departamento de Informática. Universidad Carlos III de

Madrid. 1998. Documento Digital.

29. DON. Wells. “Extreme Programming: A gentle introduction”. <http://www.extremeprogramming.org/>
30. ANTARESX. “Aprendiendo Struts”. http://www.programacion.com/blogs/46_aprendiendostruts
31. INSTITUTO TECNOLÓGICO DE MORELIA. “III Enfoques Metodológicos del Desarrollo de Sistemas de Información.” <http://antares.itmorelia.edu.mx/cursos/file.php?file=/3/enfoquesmetodologicos.pdf>
32. UNIVERSIDAD DE JAEN. “Ingeniería de la programación” <http://www.di.ujaen.es/asignaturas/ingprog/Tema1.pdf>
33. SPILLNER. Andreas. “The W-MODEL – Strengthening the Bond Between Development and Test”. University of Applied Sciences Bremen. Germany. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>
34. GILB. Tom. “10 Evolutionary Project Management (Evo) Principles”. Abril de 2002. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>
35. GILB. Kai. “Evolutionary Project Management & Product Development (or The Whirlwind Manuscript)”. 2003. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>
36. COCKBURN. Alistair. “Balancing Lightness with Sufficiency”. Septiembre de 2000. <http://alistair.cockburn.us/crystal/articles/blws/balancinglightnesswithsufficiency.html>
37. AMBLER. Scott. “Agile Modeling and the Unified Process”. Enero de 2005. <http://www.agilemodeling.com/principles.htm>

Bibliografía

Project Management Institute (PMI): A Guide to the Project Management Body of Knowledge (PMBOK Guide). Newtown Square: Project Management Institute, Inc, Fifth Edition (2013).

Perrin, Richard: PMP EXAM PREP BOOT CAMP (Based on the PMBOK Guide, 5th Ed.) : EDWel Programs, (2013).

JACOBSON. Ivar. BOOCH. Grady. RUMBAUGH. James. “El Proceso Unificado de Desarrollo de Software”. Edición en Español. Ed. Addison Wesley. 2000.

HOLLINGSWORTH. David. “Workflow Management Coalition the Workflow Reference Model”. 55 p. Enero de 1995

OBJECT MANAGMENT GROUP. “Software Process Engineering Metamodel Specification” Versión 1.0. Noviembre de 2002.

HURTADO. Julio Ariel. “Sistema Integral para el Mejoramiento de los Procesos de Desarrollo de Software en Colombia (SIMEP-SW)”. Departamento de Sistemas, Universidad del Cauca. Popayán, Julio de 2003.

HURTADO. Julio Ariel. “El modelo integral de mejoramiento Agile SPI”. Departamento de Sistemas, Universidad del Cauca. Popayán, Agosto de 2004.

FUGGETA. Alfonso. “SOFTWARE PROCESS: A Roadmap”. Dipartimento di Elettronica e Informazione. Politecnico di Milano. 1999. <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalfuggetta.pdf>

REAL ACADEMIA DE LA LENGUA ESPAÑOLA. Diccionario de la Lengua Española. Vigésima segunda edición. 2004. <http://www.rae.es/>

UNIVERSIDAD NACIONAL DE SAN JUAN. “Glosario Informático”. Argentina 2004. <http://adig.com.ar/cg3/glosario/>

UNIVERSIDAD CARLOS III DE MADRID. “Modelado de Procesos”. <http://www.ie.inf.uc3m.es/Tesis/Documentos/ProcesosSoftware/documentos.htm>

HERNÁNDEZ Muñoz. José Luís. “Rentabilidad del desarrollo de proyectos de cómputo”. México 2002. <http://www.monografias.com/trabajos13/renta/renta.shtml>

BEDINI G. Alejandro. “Calidad tradicional y de software”. Departamento de Industrias, Universidad Técnica Federico Santa María, Chile <http://www.willydev.net/descargas/Articulos/General/CalidadSoftware.pdf>

BUSINESS SOFTWARE ALLIANCE. “Industria del software podría crecer 645 % hasta el año 2005”. Febrero 2003. <http://global.bsa.org/colombia/press/newsreleases/2003-02-21.1476.phtml>

BERTOA. Manuel F, TROYA. José M y VALLECILLO. Antonio “Atributos de Calidad para Componentes COTS: Una valoración de la información ofrecida por los vendedores”. <http://www.lcc.uma.es/~{}av/Publicaciones/03/TICS03.pdf>

LETELIER. Patricio y PENADÉS. Carmen. “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”. Facultad de Informática. Universidad Politécnica de Valencia. 2004 <http://www.willydev.net/descargas/masyxp.pdf>

REYNOSO. Carlos. “Métodos heterodoxos en desarrollo de software”. http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.asp

MAMANI. Luís. “Sistema Informático para el Seguimiento y Supervisión de Proyectos de Desarrollo de Software”. Universidad Peruana de Ciencias Aplicadas. Perú 2002. <http://espejos.unesco.org.uy/simplac2002/Ponencias/Inforedu/IE020%20Luis%20Mamani.doc>

OKTAVA. Hanna. “Modelo de Procesos para la Industria de Software MoProSoft” Versión 1.1. México 2003. <http://www.lania.mx/biblioteca/manuales/moprosoft/V%201.1%20DocumentoBase.pdf>

HERNÁNDEZ. Orallo. Enrique. “El lenguaje Unificado de Modelado (UML)” Departamento de informática y computadores. UPV. 2000. <http://dark.disca.upv.es/enheror/pdf/ActaUML.pdf>

BPMWin. “What is Business Process Management (BPM)?”. 2004 <http://www.bpmtutorial.com/BPM/BPM.aspx>

SQS-SA. “SQS – RequirementsWORKFLOW. Herramienta que implementa eXtreme Programming para la gestión de requisitos”. 2004. <http://www.sqs.es/documentos/eXtreme.pdf>

PIKATZA. Juan M. “Modelo jerárquico de roles para organizaciones de pequeño tamaño”. Departamento de Lenguajes y Sistemas Informáticos. UPV/EHU. 2002. <http://www.sc.ehu.es/jiwdocoj/remis/adis2002.htm>

GARCÍA, Felix. “Gestión Integrada de modelado y de la medición del proceso software”. Grupo Alarcos. Escuela Superior de Informática. Universidad de Castilla – La Mancha. 2002. <http://lsi.ugr.es/~{}gedes/actividades/Dolmen4/a5.pdf>

ABIAN. Miguel Ángel. “J2EE Y .NET: LA RIVALIDAD PERMANENTE”. <http://www.javahispano.org/articles.article.action?id=55>

MAÑAS. José A. “Pruebas de programas”. <http://www.upv.es/protel/usr/jotrofer/pascal/testing.htm>

RUIZ G. Francisco. “MANTIS: Definición de un Entorno para la Gestión del Mantenimiento de Software”. Departamento de Informática. Universidad de Castilla – La Mancha. Junio de 2003. Documento Digital.

PUMAREJO. Johanna. “Descripción del sector del software - análisis de mercado”. Departamento de Inteligencia de Mercados. FEDESOFT 2002. Documento Digital.

RUIZ. Francisco. “Mantenimiento Avanzado de Sistemas de Información”. Universidad de Castilla – La Mancha. Ciudad Real. España 2004. Documento Digital.

GRUPO DE MEJORA DEL PROCESO DE SOFTWARE. “Resumen del libro Workflow and Process Automation”. Departamento de Informática. Universidad Carlos III de Madrid. 1998. Documento Digital.

DON. Wells. “Extreme Programming: A gentle introduction”. <http://www.extremeprogramming.org/>

ANTARESX. “Aprendiendo Struts”. http://www.programacion.com/blogs/46_aprendiendostruts

INSTITUTO TECNOLÓGICO DE MORELIA. “III Enfoques Metodológicos del Desarrollo de Sistemas de Información.” <http://antares.itmorelia.edu.mx/cursos/file.php?file=/3/enfoquesmetodologicos.pdf>

UNIVERSIDAD DE JAEN. “Ingeniería de la programación” <http://www.di.ujaen.es/asignaturas/ingprog/Tema1.pdf>

SPILLNER. Andreas. “The W-MODEL – Strengthening the Bond Between Development and Test”. University of Applied Sciences Bremen. Germany. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>

GILB. Tom. “10 Evolutionary Project Management (Evo) Principles”. Abril de 2002. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>

GILB. Kai. “Evolutionary Project Management & Product Development (or The Whirlwind Manuscript)”. 2003. <http://www.iti.upv.es/~{}squac/JTS/JTS2004/docs/Wmodel.pdf>

COCKBURN. Alistair. “Balancing Lightness with Sufficiency”. Septiembre de 2000. <http://alistair.cockburn.us/crystal/articles/blws/balancinglightnesswithsufficiency.html>.

AMBLER. Scott. “Agile Modeling and the Unified Process”. Enero de 2005. <http://www.agilemodeling.com/principles.htm>

Braude Eric: Ingeniería de Software: una perspectiva orientada a objetos, Alfaomega (2003) Center for Software Engineering of the USC.

Fraunhofer Institute for Software Engineering Germany)

Grady Booch: Análisis y Diseño Orientado a Objetos, Segunda Edición, Addison Wesley Longman, Pearson

Ian Sommerville: Ingeniería de Software, 6^a Edición, Addison Wesley

Ivar Jacobson, Grady Booch, James Rumbaugh: El Proceso Unificado de Desarrollo de Software, Rational Software Corporation, Addison Wesley, Edición en Español (2000)

Joseph Schmuller: Aprendiendo UML en 24 horas, Prentice Hall

Pankaj Jalote: “An Integrated Approach to Software Engineering”, Springer (1997)

Roger S. Pressman: Ingeniería de Software, un enfoque práctico, Sexta Edición, McGraw Hill (2006)

Shari Lawrence Pfleeger: Ingeniería de Software, Teoría y Práctica, Primera Edición, Prentice Hall (2002)

Software Engineering Institute of Carnegie-Mellon University

Stephen R. Schach: Ingeniería de Software Clásica y Orientada a Objetos, Sexta Edición, McGraw Hill (2006)

Mengunzzato, M.; Renau, J.J. 1991. La Dirección Estratégica de la Empresa. Un enfoque innovador del management. Ediciones Ariel economía pp.87,212

Becker, E et al. 2001. El cuadro de mando de RR.HH vinculado a las personas, la estrategia y el rendimiento de la empresa, Ediciones Gestión 2000, pp. 19.

Pierre Louart. Gestión de los Recursos Humanos. Ediciones gestión 2000 S.A. Pág. 55.

Kaplan,R. ; Norton, D. 2006. Incrementando los resultados mediante el alineamiento estratégico en toda la organización. Ediciones Gestión 2000, pp. 43-47

Ulrich, D. et al, 2003. El futuro de la dirección de recursos humanos, Ediciones Gestión 2000

Harper y Lynch, 1992. Motivación de personal y clima laboral. Ediciones de Publicaciones Económicas. España

Stoner, J. et al, 1996. Administración. Ediciones Pearson Educación, 6^a edición, México

Werther, W; Davis, K. 1996. Administración de personal y recursos humanos. Ediciones MacGraw-Hill.

Ansoff, I. et al, 1976. From strategic planning to strategic management. Ediciones Wiley, 1976. Michigan

Besseyre, 1997. Gestión estratégica de los recursos humanos. Ediciones Deusto

Cuesta, A. 2005. Tecnología de gestión de Recursos Humanos. Ediciones Academia, 2^a edición.

Resumen de Currículos

César Eduardo Velázquez Amador estudio la Ingeniería en Sistemas Computacionales, la Maestría en Informática y Tecnologías Computacionales y el Doctorado en Ciencias Exactas Sistemas y de la Información en el área de Ciencias de la Computación-Ingeniería de Software. Ha trabajado en el desarrollo de Sistemas tanto en la iniciativa privada como en el gobierno. Actualmente es profesor de tiempo completo en la Universidad Autónoma de Aguascalientes, impartiendo materias como Análisis y Diseño Orientado a Objetos e Ingeniería de Software, tanto a nivel licenciatura como a nivel de maestría. En el área de investigación ha participado en Artículos, Capítulos de Libro y en diversos congresos tanto a nivel nacional como internacional con trabajos relacionados a la Determinación de la Calidad y Evaluación de Objetos de Aprendizaje, Objetos de Aprendizaje Multiculturales, Adopción de Tecnologías de Información y Teoría de Servicios.

Yosly Caridad Hernández Bieliukas. Estudiante del Doctorado en Educación en la Universidad de los Andes, Venezuela. Magister en Ciencias de la Computación y Licenciada en Computación Universidad Central de Venezuela (UCV). Docente e Investigadora en la Categoría de Agregado de la Unidad de Educación a Distancia y Escuela de Computación, Profesora del Postgrado en Ciencias de la Computación. Profesora-Facilitadora de programas y Cursos de formación docente a Distancia de la UCV. Publicaciones en revistas arbitradas y congresos nacionales e internacionales. Investigadora activa en categoría B, acreditado en el Programa de Estímulo a la Investigación del ONCTI del MPPCTI. Trabajando en las líneas de investigación: Sistemas, Repositorios y Calidad de Objetos de Aprendizaje de Contenidos Abiertos (OACA), Ambientes de Aprendizaje con el uso de las TIC y Evaluación en la Educación a Distancia. Coordinadora del Proyecto Grupal PEII auspiciado por Fonacit Programa de Integración de las TIC en los Procesos de Enseñanza y Aprendizaje para Personas con Discapacidad Visual, Auditiva y Cognitiva, a través del uso de Objetos de Aprendizaje Web. Participando como investigadora en proyecto ALFA LATIn (iniciativa para la creación de libros abiertos en Latinoamérica) y en el Proyecto Autoevaluación de Programas a Distancia en la UCV auspiciado por el CALED.

Jaime Muñoz Arteaga es profesor investigador del centro de ciencias básicas de la Universidad Autónoma de Aguascalientes (UAA), el obtuvo su doctorado en ciencias computacionales en Francia en el 2000, así como un postdoctorado de 2 años con sello CONACYT. El doctor Muñoz Arteaga es miembro del SNI I, realizando para ello investigaciones y docencia en las áreas tecnologías educativas, interacción humano-computadora y de ingeniería web, en colaboración con otros grupo de investigación de otra instituciones nacionales (tales como la UV, el INAOE, CENIDET y la UNAM) e Internacionales (tal como la ESPOL de Ecuador y la Université Catholique de Louvain en Bélgica). El doctor Muñoz actualmente es líder de un cuerpo académico con un nivel consolidado PROMEP, en el tema de objetos de aprendizaje y de ingeniería de software, para ellos ha dirigido diversos trabajos postgrado a nivel de maestría y a

nivel doctorado. El doctor ha liderado proyectos de investigación de CONACYT de SEP-UNAM, Fondos Mixtos y ALFA III. Por último, el doctor ha publicado un libro en el área de ingeniería de software y dos libros en tecnologías de objetos de aprendizaje.

Francisco Javier Alvarez Rodríguez, Profesor de Ingeniería de Software adscrito al Departamento de Ciencias de la Computación, Universidad Autónoma de Aguascalientes. Ha sido Jefe de Departamento de Sistemas Electrónicos y miembro de núcleos académicos de diversos posgrados de la UAA. Autor de libros y artículos sobre la línea Objetos de Aprendizaje y Procesos de Desarrollo de Software. Evaluador internacional para programas de Ingeniería en Centro America, miembro del CENEVAL para exámenes de egreso de Ing. de Software, y miembro del comité de Acreditación del CONAIC. Responsable del proyecto de certificación de procesos académicos bajo la norma ISO 900-2001 para Centro de C. Básicas de la UAA.

Margarita Mondragón Arellano Ingeniero en Sistemas Computacionales (1994), Maestra en Informática y Tecnologías Computacionales (1998), Doctora en Ciencias Exactas, Sistemas y de la Información área Ingeniería de Software (2013), egresada de la Universidad Autónoma de Aguascalientes. Con 20 años de experiencia profesional y 18 como docente. Ha Desempeñado diversos puestos: Analista, Diseñador, Desarrollador, Probador, Capacitador y Líder en una serie de Proyectos Informáticos. Cuenta con una amplia trayectoria en la capacitación de personal en el ramo de la Informática. Como consultor Externo sus áreas de interés son: Ingeniería de Software, Procesos de Negocio, Dispositivos Móviles, Bases de Datos, Lenguajes de Programación, Desarrollo de Material y Software Didáctico.



Edición: Marzo de 2014.

Este texto forma parte de la Iniciativa Latinoamericana de Libros de Texto abiertos (LATIn), proyecto financiado por la Unión Europea en el marco de su [Programa ALFA III EuropeAid](#).



Los textos de este libro se distribuyen bajo una Licencia Reconocimiento-CompartirIgual 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/deed.es_ES