

RALPH R. YOUNG



THE
**REQUIREMENTS
ENGINEERING
HANDBOOK**

La ingeniería de requisitos Manual

Para obtener una lista de títulos recientes en la Biblioteca de Desarrollo Profesional y Gestión Tecnológica de Artech House, consulte la parte posterior de este libro.

La ingeniería de requisitos Manual

Ralph R. joven



Artech House
Boston • Londres
www.artechhouse.com

Datos de catalogación en publicación de la Biblioteca del Congreso
Un registro de catálogo para este libro está disponible en la Biblioteca del Congreso de EE. UU.

Catalogación de la Biblioteca Británica en datos de publicación
Un registro de catálogo para este libro está disponible en la Biblioteca Británica.

Diseño de portada de Igor Valdman.

© 2004 CASA ARTECH, INC.
685 calle cantón
Norwood, MA 02062

Los siguientes están registrados en la Oficina de Patentes y Marcas de EE. UU. por la Universidad Carnegie Mellon: Capability Maturity Model , CMM , y CMMI .

Reservados todos los derechos. Impreso y encuadrado en los Estados Unidos de América. Ninguna parte de este libro puede reproducirse ni utilizarse de ninguna forma ni por ningún medio, electrónico o mecánico, incluidas fotocopias, grabaciones o cualquier sistema de almacenamiento y recuperación de información, sin el permiso por escrito del editor.

Todos los términos mencionados en este libro que se conocen como marcas comerciales o marcas de servicio se escriben con mayúscula apropiada. Artech House no puede dar fe de la exactitud de esta información. No se debe considerar que el uso de un término en este libro afecta la validez de ninguna marca comercial o de servicio.

Número de libro estándar internacional: 1-58053-266-7

Un número de tarjeta de catálogo de la Biblioteca del Congreso está disponible en la Biblioteca del Congreso.

10 9 8 7 6 5 4 3 2 1

Para ti

¡Mejoremos la ingeniería de requisitos!

Machine Translated by Google

Contenido

Prólogo	xi
Prefacio.....	xv
Agradecimientos	xix
Importancia de los requisitos....	1
¿Qué son los requisitos y por qué son importantes?	1
¿Por qué planificar?	3
Una estrategia sugerida	3
Requisitos Actividades en el ciclo de vida del sistema	3
Inversión en el proceso de requisitos Un enfoque	5
de proceso El plan de	6
requisitos Factores que	7
afectan sus decisiones profesionales Un	10
comentario sobre proyectos pequeños	11
Resumen	11
Caso de estudio	12
Referencias	13
s funciones de la AR.	15
Funciones sugeridas de la AR	15
Resumen	23
Caso de estudio	24
Referencias	25

3 Habilidades y características de un RA eficaz.	27
Habilidades de la RA	27
Características de una AR efectiva	34
Resumen	42
Caso de estudio	43
Referencias	44
4 tipos de requisitos.	45
Vistas de tipos de requisitos	45
Definiciones y descripciones de tipos de requisitos	48
Requisitos comerciales	49
Requisitos declarados versus requisitos reales	50
Requisitos de usuario	50
Requisitos de alto nivel o de sistema	50
Reglas del negocio	50
Requerimientos funcionales	51
Requerimientos no funcionales	52
Requisitos derivados	52
Requisitos de diseño y restricciones de diseño	52
Requisitos de desempeño	53
Requisitos de interfaz	53
Requisitos verificados	53
Requisitos validados	53
Requisitos de calificación	53
Las "llidades" y los requisitos de ingeniería especializada	53
Requisitos incognoscibles	54
Requisitos del producto	54
Requisitos del proceso	54
Requisitos de soporte logístico	54
Requisitos medioambientales	55
Requisitos del sistema, subsistema y componentes	55
Terminologías a evitar	55
Requisitos de fuente o cliente	55
Requisitos no negociables versus requisitos negociables	55
Requerimientos clave	56
Requisitos de origen	56
Otras pautas	56
Ejemplos de tipos de requisitos	56
Resumen	57
Caso de estudio	57
Referencias	60

5 Requisitos de reunión.	61
Planifique el enfoque	62
Resumen	104
Caso de estudio	104
Referencias	105
6 mejores prácticas para el desarrollo de requisitos y Gestión.	109
Resumen	123
Caso de estudio	123
Referencias	126
7 Habilidades especializadas de la RA.	127
Resumen	159
Caso de estudio	163
Referencias	164
8 Un enfoque de calidad integrado.	169
Impulsores empresariales para la calidad	170
El papel de la gerencia	170
Principios rectores	171
Gestión de prioridades	172
Los componentes de un enfoque integrado de calidad	172
Técnicas de mejora de la calidad	173
El ciclo PDCA	179
Cómo diseñar un proceso	180
Trabajo en equipo	187
Resumen	189
Estudio de caso: un ejemplo de mejora de la calidad desviada	189
Referencias	191
9 Una visión para la ingeniería de requisitos.	193
¿Cómo debemos apoyar a los PM?	197
¿Cómo debemos apoyar a los clientes?	198
¿Cómo deberíamos apoyar a los desarrolladores?	198
Resumen del	199
estudio de caso	200
Referencias	202

10 Avanzando: requisitos conocibles,	
Riesgo manejable.	205
A dónde ir desde aquí	207
Avanzando	209
Un mandala de requisitos	212
Resumen	213
Caso de estudio	213
Referencias	215
Glosario.	217
Lista de acrónimos	227
Bibliografía.	233
Sobre el Autor	243
Índice.	245

Prefacio

Hace algunos años, una empresa de éxito consiguió un contrato por valor de cincuenta millones de dólares. El sistema del producto tenía seis consolas de operador, otros seis bastidores de equipos electrónicos y un sofisticado conjunto de radios y computadoras remotas. Las disciplinas de desarrollo incluyeron ingeniería de software, electrónica digital, electrónica de comunicaciones e ingeniería mecánica.

La adquisición de clientes y los grupos de usuarios sabían qué capacidad operativa querían, pero aún no había requisitos técnicos. Al principio del proyecto, la empresa desarrolló y entregó una especificación técnica.

Los revisores de los clientes proporcionaron docenas de cambios, incluidos seis requisitos adicionales que interpretaron a partir de declaraciones vagas de capacidad operativa. Sin embargo, en una discusión posterior, los clientes estuvieron de acuerdo en que estos requisitos adicionales, aunque agradables de tener, eran demasiado costosos para agregarlos.

Pasó el tiempo. El proyecto tuvo problemas políticos y de financiación, y saltó de arriba a abajo durante un período de cuatro años como un avión lanzadera de corto recorrido.

El personal cambió varias veces; En uno de esos cambios, el proyecto aparentemente terminó y todo el grupo de adquisición fue reasignado. Después de dos meses de pausa, un nuevo grupo de adquisiciones reanudó el proyecto.

La especificación técnica pasó por varias revisiones más. De alguna manera, el registro de esos seis requisitos permaneció. Sin embargo, el registro del acuerdo se perdió repetidamente. Los revisores de los clientes los reinseraron repetidamente. Después de cada revisión, nuevamente se consideró que agregar esos seis requisitos era demasiado costoso. Pero la especificación nunca fue aprobada del todo para reflejar el acuerdo. Mientras respondían a los frecuentes trastornos, los desarrolladores se centraron en completar el diseño y la producción. La especificación se desvaneció en un estante polvoriento.

Las cosas en un estante polvoriento tienen una manera de volver a atormentarnos. Esos seis requisitos salieron a la luz por última vez. Durante las pruebas de aceptación del sistema, los monitores de los clientes quitaron el polvo de la especificación y comenzaron una verificación formal.

El resultado de la falta de seis requisitos fue un sobrecoste de tres millones de dólares.

El libro de Ralph Young proporciona las herramientas que esa empresa necesitaba y no tenía. Basándose en las prácticas efectivas de requisitos y en sus años de experiencia práctica, Ralph ofrece un conjunto de herramientas y técnicas que son esenciales para el análisis de requisitos moderno, escritas en un formato de manual para

referencia continua. Este libro describe tanto la filosofía como la práctica del análisis de requisitos, con un pragmatismo realista que puede ayudar a hacer el trabajo frente a los complejos desafíos de los sistemas actuales.

Las comunicaciones humanas son imprecisas. Es un aspecto de la naturaleza de la humanidad que no logramos entendernos completamente.

Recuerda los juegos de fogata de tu infancia. En el juego "Susurros", alguien comienza una breve declaración alrededor del círculo de la fogata susurrándole a su vecino. Por turnos, cada jugador transmite la declaración, siempre en un susurro. Despues de sólo unas pocas transferencias, la declaración original se modifica hasta quedar irreconocible. En el juego, las diferencias son tan sorprendentes que hacen reír a todos.

Recuerde la última vez que fue a un restaurante y pidió un menú.

Escrito frente a usted hay una descripción completa del plato principal que desea. Con confianza, le dices al camarero el plato principal. El camarero suspira en silencio y comienza la perenne secuencia de preguntas sobre los acompañamientos. "¿Ensalada o sopa, señora?" "¿Papas nuevas o papas fritas?" "¿Judías verdes o succotash?" Todas estas opciones estaban claramente escritas en el menú, pero de alguna manera te las perdiste.

Los requisitos también son una forma de comunicación humana, un intento de transmitir ideas complejas de una mente a otra. Los requisitos también son una forma escasa de comunicación, que utiliza palabras escritas simples para esforzarse por lograr precisión. Al igual que las descripciones del menú, los requisitos siempre se quedan cortos. Es literalmente imposible escribir cualquier requisito, por simple que sea, que no pueda ser malinterpretado honestamente por algún destinatario.

Incluso la palabra "requisito" es en sí misma un error de comunicación, ya que los requisitos individuales con frecuencia son más flexibles que obligatorios. Si una compensación promete un beneficio significativo para un parámetro clave de desempeño, los especificadores con gusto cambiarán "requisitos" menores para acomodar la compensación.

Y, sin embargo, los requisitos siguen siendo el mejor método que conocemos para transmitir la complejidad de una idea técnica. Para manejar esta complejidad, utilizamos requisitos para desempeñar tres roles importantes, todos los cuales se mejoran con las herramientas y técnicas de este libro.

Primero, los requisitos son una herramienta contractual. Este es el propósito más comúnmente entendido. En esta función, una especificación define el alcance técnico de un contrato de desarrollo. El impacto legal de este papel está lejos de ser pequeño. Una demanda reciente entre un contratista principal y un subcontratista dependía de la gramática de una única declaración de requisitos, lo que resultó en un acuerdo multimillonario. Para proteger tanto a los adquirentes como a los proveedores, los requisitos contractuales deben ser lo más claros posible.

En segundo lugar, los requisitos son una herramienta de gestión de la configuración. La forma exacta y la relación de las declaraciones de requisitos definen de forma única una configuración del sistema. Representan la funcionalidad y los límites válidos del sistema. Controlando los requisitos, controlamos la definición de la configuración. Vemos la importancia de la definición de la configuración cada vez que una nueva herramienta de software deja de funcionar con nuestra computadora personal de "sistema abierto".

En tercer lugar, los requisitos son una herramienta de ingeniería. Este papel esencial muchas veces no se comprende, quedando eclipsado por las relaciones contractuales y

roles de gestión de configuración. Sin embargo, es en la ingeniería donde los requisitos tienen su poder. Utilizamos requisitos durante los procesos de ingeniería para hacer lo siguiente:

- Comunicarse entre los miembros del equipo de desarrollo, adquirentes, usuarios, y otros;
- Describir y comprender la necesidad operativa;
- Captar decisiones sobre la solución técnica;
- Definir la arquitectura del producto;
- Verificar la finalización de los elementos del producto;
- Verificar la finalización del producto.

El problema de esos seis requisitos se debió a muchos factores: los cambios políticos en el programa, las ideas en competencia entre las facciones de clientes, las presiones inusuales de iniciar y detener el desarrollo y el enfoque del equipo de desarrollo en la finalización. Sin embargo, ese problema también ocurrió debido a la falta de los métodos de gestión de requisitos (RM) que contiene este libro. Un esfuerzo moderno de RM para todo el proyecto habría costado una fracción de los tres millones de dólares de sobrecosto experimentado. Aún mejor, también se habrían evitado muchos otros gastos.

Al igual que los niños alrededor de la fogata, la informalidad conduce a problemas de comunicación. Como juego de fogata, nos reímos de los problemas. Sin embargo, al construir los complejos productos de sistemas actuales ya no nos reímos.

eric honor

Ex presidente del Consejo Internacional de Ingeniería de Sistemas

Machine Translated by Google

Prefacio

Este libro pretende ser una referencia concisa pero completa para las necesidades analíticas de requerimientos (RA) aquellos que están asignados para determinar los requisitos para los sistemas y software planificados, tanto en informática como en ingeniería.

Es una guía/manual de escritorio que se centra en cómo los RA pueden realizar mejor su trabajo.

Los requisitos son clave para el éxito o el fracaso de los proyectos técnicos.

Son la base de todo el trabajo posterior. Según mi experiencia, la mayoría de los proyectos y organizaciones no utilizan prácticas de requisitos efectivas y un proceso de requisitos documentado, y también que aquellos asignados como RA son asignados al trabajo necesario sin la preparación, experiencia y capacitación adecuadas, y sin un buen manual que los asesore. sobre cómo desempeñar sus funciones y qué hacer.

Las AR se encuentran en una posición estratégica para influir en las actividades realizadas en un Tarea o proyecto de ingeniería de sistemas o software:

Los requisitos son vitales para el inicio, realización y finalización del trabajo necesario.

Son de gran importancia en la consecución de los objetivos de los clientes y usuarios.

Los RA capacitados y experimentados son asesores valiosos para el administrador del programa, proyecto o tarea y recursos invaluables para otros miembros del equipo.

Este libro aborda todas las áreas que necesitará conocer en su trabajo. Los temas clave incluyen los siguientes:

Tema

La importancia de los requisitos.

Aprovechar las actividades relacionadas con los requisitos para beneficiar su proyecto.

Identificar los requisitos reales.

- Controlar los cambios de requisitos y nuevos requisitos.
- Utilizar prácticas, procesos, métodos, técnicas y requisitos eficaces. y herramientas.
- Invertir en el proceso de requisitos.
- Evalúe sus requisitos según los criterios de un buen requisito.
- Documentar la justificación de cada requisito.
- Planificar actividades relacionadas con los requisitos.
- Utilice una herramienta de requisitos automatizada de potencia industrial.
- Trabajar para mejorar las comunicaciones. Utilice un glosario del proyecto y acrolista de nym.
- En colaboración con los líderes de la tarea o del proyecto, seleccione un conjunto de mejores prácticas y luego impleméntelas de manera efectiva.
- Desarrollar un plan de desarrollo profesional personal y potenciar tus habilidades y capacidades.
- Aprender y aplicar las habilidades especializadas necesarias del analista de requisitos.
- Definir y utilizar un enfoque de calidad integrado.
- Desarrolla tu propia visión personal de la ingeniería de requisitos.
- Abordar los riesgos de los requisitos.

El Capítulo 2 describe nueve funciones de la RA e identifica en qué parte del ciclo de vida del sistema se debe aplicar cada una. El trabajo de requisitos requiere muchos conocimientos y habilidades, quizás más de lo que la mayoría de la gente piensa. El Capítulo 3 identifica y describe las habilidades que puede necesitar utilizar y proporciona una referencia en este libro donde puede aprender más sobre cada habilidad.

Es importante que el RA y otras personas asignadas a la tarea o proyecto comprendan los diferentes tipos de requisitos. Estos se describen en detalle en el Capítulo 4 y se desglosan de la siguiente manera:

- Requisitos comerciales;
- Requisitos del usuario;
- Requisitos del producto;
- Requisitos medioambientales;
- Requisitos incognoscibles.

Las necesidades y expectativas de los clientes se analizan y describen de las siguientes maneras:

- Requisitos de alto nivel (o nivel de sistema);
- Requisitos funcionales (lo que debe hacer el sistema);

Requisitos no funcionales:

Propiedades del sistema (por ejemplo,

seguridad); Los "requisitos de ingeniería especializada/llidades";

Requisitos derivados y restricciones de diseño;

Requisitos de desempeño (por ejemplo, ¿a qué velocidad?);

Requisitos de interfaz (relaciones entre elementos del sistema).

Luego, los requisitos del sistema se asignan a lo siguiente:

Subsistemas (agrupaciones lógicas de funciones);

Componentes del sistema (hardware, software, formación, documentación).

Se realizan comprobaciones para garantizar que el sistema haga lo que se supone que debe hacer, incorporando lo siguiente:

Requisitos verificados;

Requisitos validados;

Requisitos de calificación.

De esta descripción de las actividades requeridas se desprende lo siguiente:

Se necesita un entendimiento común y compartido sobre la tarea o proyecto;

Se debe brindar capacitación relacionada con los requisitos a tres grupos con necesidades algo diferentes, de modo que todos puedan beneficiarse de la experiencia de la industria y tomar conciencia de los métodos, técnicas y herramientas que funcionan mejor: (1) los RA, (2) los miembros de el equipo de desarrollo, y (3) los clientes y usuarios.

Estos temas se abordan en el Capítulo 5.

Las actividades de recopilación de requisitos son complejas y pueden volverse ineficaces fácilmente, como probablemente ya sabrá por su experiencia.

Consulte la Tabla 5.1 para obtener una lista de verificación que le ayudará en sus funciones vitales.

El Capítulo 5 proporciona una discusión de cada actividad, explicando por qué es importante y cómo realizarla. Las referencias le permiten acceder a más información, si la desea. La intención del libro es empoderarlo para que haga una contribución valiosa y también para que se sienta satisfecho en sus actividades laborales.

Escuchamos y leemos mucho sobre las "mejores prácticas". Desafortunadamente, con muy poca frecuencia se implementan, implementan e institucionalizan en proyectos reales, por una variedad de razones, pero la más importante es que es un trabajo duro.

El Capítulo 6 proporciona información sobre las mejores prácticas para los RA, según mi propia experiencia y la de la industria. Evidentemente no se puede hacer todo, al menos no al mismo tiempo. La información del Capítulo 6 le permitirá iniciar discusiones con su equipo de tarea o proyecto y seleccionar las mejores prácticas.

que mejor respalden las necesidades, actividades y objetivos de su proyecto o de su organización.

Existe un conjunto de habilidades especiales de la RA que se requieren en diferentes momentos de su trabajo. El Capítulo 7 describe estas habilidades especializadas y lo dirige a la sección de este libro donde se analiza cada una de ellas.

El Capítulo 8 explica la importancia de un enfoque de calidad integrado.

Es necesario un proceso de requisitos implementado efectivamente para tener un enfoque de calidad integrado, y se requiere un enfoque de calidad integrado para que el proceso de requisitos funcione mejor. El capítulo 8 explica qué se entiende por estos términos y, también, cómo lograrlos.

El capítulo 9 proporciona una visión de la ingeniería de requisitos. Se dará cuenta (si aún no lo ha hecho) de que el progreso en ingeniería de requisitos ha sido lento. ¡Este libro está dedicado a usted, con el desafío de mejorar la práctica de la ingeniería de requisitos! Este libro está lleno de ideas, sugerencias, enfoques y recomendaciones prácticas y le resultará muy útil como manual en su trabajo diario. Pero solo si (1) lo usa y (2) está decidido a trabajar para mejorar las cosas. Esto sugiere que debe comprometerse a realizar cambios que mejoren la forma en que desarrollamos sistemas y software. Con demasiada frecuencia no practicamos lo que predicamos. Sabemos lo que debemos hacer, pero no lo hacemos. La conclusión es que su compromiso y el de sus pares y gerentes es necesario si queremos lograr mejoras. Utilice este libro para guiarle en su trabajo vital. Además, podrá crear e implementar su propio plan de desarrollo profesional basado en las características y rasgos que elija fortalecer y mejorar aún más. Le animo a trabajar en conjunto con su gerente para desarrollar un plan de acción que le permita comprender integralmente sus funciones y, a través de la experiencia y el estudio, desarrollar la experiencia necesaria para impactar las tasas de éxito del proyecto de manera significativa y positiva.

El capítulo 10 proporciona un resumen del libro y sugerencias para seguir adelante. El subtítulo “Requisitos cognoscibles, riesgo manejable” sugiere que realmente podemos hacer un trabajo encomiable cuando estamos capacitados y aplicamos las pautas proporcionadas en este libro. Al hacerlo, ayudará a mejorar las profesiones de informática e ingeniería.

Reconozcamos que nos queda un largo camino por recorrer. Pero hay orientación disponible sobre cómo realizar mejoras (en nuestras políticas, prácticas, procesos, métodos, técnicas y herramientas, por ejemplo). Para que la profesión mejore, los profesionales deben tomar medidas en su trabajo diario que sean diferentes a las que estamos haciendo hoy. Sólo a través de acciones incrementales graduales, pero comprometidas, la profesión avanzará para lograr una visión positiva de la ingeniería de requisitos.

¿Estás listo?

Comparta sus propias reacciones a este libro, ideas, sugerencias y críticas constructivas conmigo en ryoungrr@aol.com. Sin duda trabajaré arduamente en otro proyecto y sus comentarios mejorarán cualquier contribución que pueda hacer.

¡Buena suerte y recuerda divertirte mientras haces todo esto!

Expresiones de gratitud

Sigo estando muy agradecido con mi esposa, Judy, por su increíble paciencia, comprensión y amor durante todo el proceso de escritura del libro. A veces, mis familiares y amigos se sorprenden de la alegría y la energía que obtengo al escribir; para mí, es como lo caracteriza Maryanne Radmacher-Hershey:

Escribir es el proceso que uno sigue para aprender lo que ya sabe en lo más profundo de su ser: agudiza el espíritu, disciplina la mente y conduce a soluciones. En los espacios entre las palabras y la soledad observa lo que sucede cuando las palabras y el silencio se encuentran. Las palabras importan. Prestar atención. Escribe para aprender lo que saber.

En cuanto a aquellos que me han apoyado, ¿cómo puedo reconocer adecuadamente a los críticos y contribuyentes que han aportado tanto y tan generosamente para hacer de este un libro mejor? Baste decir que muchas personas se han convertido en amigos íntimos y valiosos asesores.

Hablando de asesores, ¡hay uno cuyo nombre ni siquiera sé! Artech House Publishers contrató a un asesor para revisar mis borradores finales. Esta persona, obviamente un experto en ingeniería de requisitos, optó por permanecer en el anonimato y proporcionó comentarios constructivos y útiles sobre cada capítulo.

Hablando de expertos en ingeniería de requisitos, Ian Alexander brindó generosamente comentarios e ideas reflexivas en innumerables áreas, respondiendo casi en tiempo real a preguntas, consultas y solicitudes de comentarios de revisión. Randall Iliff, mentor de ingeniería y gestión de proyectos, también aportó excelentes conocimientos en varias áreas. Jeff Grady, Earl Hoovler, Capers Jones, John Moore, Rich Raphael y Doug Smith contribuyeron con ideas y textos reflexivos y útiles.

Los analistas de requisitos que son miembros del Grupo de Trabajo de Requisitos de Soluciones Empresariales de Defensa de Tecnología de la Información de Northrop Grumman proporcionaron valiosos comentarios de revisión, contribuciones y aportaron sus experiencias y conocimientos: Terry Bartholomew, Michael Davis, David Ebenhoeh, Bob Ellinger, Jim Faust, Graham Meech, , Dick Pederson, rico

Raphael, Dave Reinberger, Ron Rudman, Charlie Rynearson y John Waters, en particular.

Otros analistas de requisitos que hicieron valiosas contribuciones incluyen a Dorothy Firsching, Chris Fowler, Heather Gray, Skip Jensen, Wayne O'Brien, Joy van Helvert, Charlie Wight y Don Young.

Los gráficos, ilustraciones, tablas y figuras son componentes críticos de cualquier trabajo porque transmiten ideas y resumen información. Gracias a Rich Raphael por crear muchos de los contenidos de este libro, por su experiencia en su elaboración y refinamiento y por su constante disposición a ayudar en todo lo posible. Olga Rosario también aportó mucho.

Muchos de los "artefactos" del libro se beneficiaron de adiciones, correcciones y comentarios de revisión aportados por los participantes en mis cursos de ingeniería de requisitos, tutoriales, presentaciones y talleres que me encanta presentar. Gracias a todos ustedes, particularmente a Pat Little.

Otros amigos y asociados que prestaron su mano y su mente incluyen a Barry Boehm, Grady Booch, Dennis Buede, Pete Carroll, Tom Gilb, Ellen Gottes-diener, Eric Honor, Alice Hill-Murray, Craig Hollenbach, Ivy Hooks, Ray Huber, Charles Markert., Andy Meadow, Larry Pohlmann, Olga Rosario, Penny Waugh y Beth Werner.

Los revisores, incluidos muchos de los mencionados anteriormente, han fortalecido mis escritos. Otros críticos incluyen a Randy Allen, Jim Hay-den y Karl Wiegers. ¡Escribir un libro es claramente un esfuerzo de equipo!

Nuestro presidente de Northrop Grumman Information Technology Defense Enterprise Solutions, Kent R. Schneider, y mi gerente, Alan Pflugrad, tienen talento para crear y mantener un entorno de TRABAJO EN EQUIPO (¡lea sobre esto en el Capítulo 8!). Es muy gratificante y estimulante ser miembro de varios equipos de alto rendimiento a través de Northrop Grumman. Agradezco a Kent y Al por su liderazgo, apoyo y orientación.

Sigo siendo consciente, desde mi camino de fe, de que la oración funciona. Gracias a nuestros queridos amigos Art Banks, Tom Foss, Craig Hollenbach y Joe Matney.

Gracias también a la familia por su apoyo: Kimberly y Mike Wallace, Ann y Jeff Young, Matt Young y Jan y Don Hoffer.

CAPÍTULO

1

La importancia de los requisitos

Contenido

¿Qué son los requisitos y por qué son importantes?

¿Por qué planificar?

Una estrategia sugerida

Requisitos Actividades en el Ciclo de vida del sistema

Inversión en el Proceso de requisitos

Un enfoque de proceso

El plan de requisitos

Factores que afectan su carrera Decisiones

Un comentario sobre los pequeños Proyectos

Resumen

Caso de estudio

Referencias

El propósito de este libro es ayudarle a mejorar la práctica de la ingeniería de requisitos. La ingeniería de requisitos es difícil. No se trata simplemente de escribir lo que el cliente dice que quiere. Un problema fundamental en los negocios es que los requisitos son inherentemente dinámicos; cambiarán con el tiempo a medida que cambie nuestra comprensión del problema que estamos tratando de resolver. La importancia de unos buenos requisitos y la naturaleza dinámica subyacente del proceso significan que debemos ser lo más precisos posible y, al mismo tiempo, ser flexibles. Flexible no significa “débil”, sino que tenemos un proceso para desarrollar requisitos y adaptarnos a los cambios a medida que aclaramos los requisitos reales de los clientes. Las prácticas de requisitos ineficaces son un problema en toda la industria. Esta es un área en la que usted puede tener un impacto positivo importante. Se necesita un enfoque más disciplinado para el desarrollo y la gestión de requisitos a fin de mejorar las tasas de éxito de los proyectos. Un alarmante 53% de la inversión de la industria en proyectos de desarrollo técnico es víctima de sobrecostos y proyectos fallidos.

Este capítulo define el término “requisito”, explica por qué los requisitos son importantes y recomienda la planificación para definir la estrategia y las actividades de requisitos. Sugiere el uso de un proceso de requisitos definido y documentado, que invertir más en el proceso de requisitos tendrá una gran rentabilidad y que los requisitos desempeñan un papel crucial en los negocios en la gestión del riesgo. Recomienda que considere ciertos factores al tomar sus decisiones profesionales. Sugiere que muchos de los consejos proporcionados en el libro son aplicables a proyectos de todos los tam-

¿Qué son los requisitos y por qué son importantes?

Un requisito es un atributo necesario en un sistema, una declaración que identifica una capacidad, característica o factor de calidad de un

sistema para que tenga valor y utilidad para un cliente o usuario. Los requisitos son importantes porque proporcionan la base para todo el trabajo de desarrollo que sigue. Una vez establecidos los requisitos, los desarrolladores inician el otro trabajo técnico: diseño, desarrollo, prueba, implementación y operación del sistema.

Con demasiada frecuencia, existe una tendencia a querer iniciar lo que a menudo se denomina "el trabajo real" (desarrollar o programar el software) demasiado rápido.

Muchos clientes y gerentes de proyectos (PM) parecen creer que la realidad El trabajo de programación ("codificación") indica que se están logrando avances.

Según la experiencia del sector, no se dedica suficiente tiempo y esfuerzo a las actividades relacionadas con los requisitos asociadas con el desarrollo del sistema.

La experiencia del sector confirma que un mejor enfoque es invertir más tiempo en actividades de recopilación, análisis y gestión de requisitos. La razón es que, normalmente, el trabajo de codificación se inicia mucho antes de lo que debería porque se necesita tiempo adicional para identificar los requisitos "reales" y planificar las actividades relacionadas con los requisitos (descritas a continuación).

Existe una diferencia significativa entre los requisitos "declarados" y los "reales". Los requisitos indicados son aquellos proporcionados por un cliente en el comienzo de un esfuerzo de desarrollo de sistema o software, por ejemplo, en una solicitud de información, propuesta o cotización o en una declaración de trabajo (SOW). Los requisitos reales son aquellos que reflejan las necesidades verificadas de usuarios para un sistema o capacidad particular. A menudo existe una gran diferencia entre los requisitos declarados y los requisitos reales. Se requiere un análisis de los requisitos establecidos para determinar y refinar los requisitos reales.

necesidades y expectativas del cliente y usuario sobre el sistema entregado. El Los requisitos deben filtrarse mediante un proceso de clarificación de su significado e identificación de otros aspectos que deben considerarse. Para citar un

En un ejemplo simple, los analistas de requisitos (RA) están más familiarizados con la necesidad indicar claramente los requisitos (consulte los criterios para un buen requisito que se proporcionan a continuación). Hay muchas maneras en que la capacidad, la comprensión, y la comunicación del significado de todos y cada uno de los requisitos puede ser diferente para un usuario que para un desarrollador. Por lo tanto, es vital (y ahorra tiempo) que todos los requisitos se aclaren a través del mecanismo de un esfuerzo conjunto de cliente/usuario y RA. Los clientes y usuarios necesitan el apoyo de profesionales técnicamente capacitados y con experiencia, y viceversa, para garantizar comunicación efectiva. Los desarrolladores deben tener la misma comprensión para que la solución que definen aborde las necesidades de la manera en que todos espera. Los malentendidos de los requisitos resultan en un desperdicio de esfuerzo y rehacer. Otra idea importante es que a veces los requisitos son incognoscible al comienzo de un esfuerzo de desarrollo porque se ven afectados por las nuevas capacidades que proporcionará el nuevo sistema. Esto sugiere la necesidad de planificar requisitos nuevos y modificados, para proporcionar un grado de flexibilidad.

Identificar los requisitos reales requiere un proceso de requisitos interactivo e iterativo , respaldado por prácticas, procesos, mecanismos, métodos, técnicas y herramientas eficaces. Este libro proporciona una descripción de cómo la RA puede utilizarlos para realizar el trabajo necesario. en un anterior

En el libro Prácticas efectivas de requisitos [1], describo lo que se debe hacer y proporciono un amplio conjunto de referencias a muchas de las mejores publicaciones de la literatura de la industria. Este libro tiene como objetivo proporcionar un manual conciso que sirva como guía de referencia para el RA o ingeniero y gerente de requisitos en ingeniería e informática. También proporciona referencias a

El proceso de requisitos no tiene por qué ser complicado ni costoso. Sin embargo, se requiere un proceso de requisitos para un proyecto de cualquier tamaño. Es muy importante que un proyecto u organización tenga un proceso de requisitos definido y documentado. La naturaleza de los componentes específicos del proceso definido se puede mejorar basándose en la experiencia.

¿Por qué planificar?

La mayoría de la gente sabe y entiende bien que un poco de planificación ayuda mucho. Por ejemplo, antes de emprender un viaje en automóvil, consultar un mapa para localizar el destino y, tal vez, incluso planificar una ruta puede ser tiempo bien empleado. Sin embargo, con frecuencia avanzamos con poca o ninguna planificación, ¿no es así? Es parte de la naturaleza humana querer continuar con el trabajo necesario sin hacer mucha planificación.

Los gerentes y profesionales de desarrollo de sistemas y software están familiarizados con varios tipos de planes: plan de proyecto, plan de gestión de ingeniería de sistemas (SEMP), plan de garantía de calidad (QA), plan de gestión de configuración (CM), plan de desarrollo de software (SDP).), plan de prueba, etc. Sin embargo, el concepto de plan de requisitos puede resultar nuevo para usted. Aprovechar las actividades relacionadas con los requisitos tiene un gran poder y efecto. Escribir un plan de requisitos maximiza el valor. Un plan de requisitos define cómo evolucionarán los requisitos reales y cómo se abordarán las actividades de requisitos.

Escribir un plan de requisitos (RP) facilita la comprensión de las actividades y esfuerzos que deben realizarse para implementar un proceso de requisitos eficaz para un esfuerzo de desarrollo particular. A continuación se proporcionan detalles adicionales sobre el plan de requisitos.

Una estrategia sugerida

Sugiero una estrategia que incluya (1) redactar un plan de requisitos, (2) diseñar o adaptar un proceso de requisitos para su proyecto, (3) invertir en las actividades relacionadas con los requisitos en el ciclo de vida del sistema y (4) utilizar los requisitos efectivos, prácticas, mecanismos, métodos, técnicas, herramientas y capacitación que se describen en este libro.

Requisitos Actividades en el ciclo de vida del sistema

Los gerentes a menudo piensan que las actividades relacionadas con los requisitos consisten principalmente en recopilar requisitos y gestionar cambios en ellos.

requisitos a lo largo del ciclo de vida. En realidad, existen varias otras actividades relacionadas con los requisitos que deben abordarse en el ciclo de vida del sistema:

Identificar a las partes interesadas: Esto incluye a cualquiera que tenga interés en el sistema o en que posea cualidades que satisfagan necesidades particulares.

Obtener una comprensión de las necesidades de los clientes y usuarios para el sistema planificado y sus expectativas sobre el mismo: esto a menudo se denomina obtención de requisitos. Tenga en cuenta que los requisitos pueden incluir varios tipos. Los tipos de requisitos se analizan en el Capítulo 4. Las técnicas de recopilación de requisitos se analizan en el Capítulo 5.

Identificar requisitos: esto implica establecer requisitos en oraciones simples y proporcionarlos como un conjunto. Las necesidades o requisitos comerciales son las actividades esenciales de una empresa. Se derivan de las metas comerciales (los objetivos de la empresa). Los escenarios comerciales se pueden utilizar como técnica para comprender los requisitos comerciales. Un factor clave en el éxito de un sistema es la medida en que respalda los requisitos del negocio y facilita que una organización los alcance.

Aclarar y reformular los requisitos: Esto se hace para garantizar que describan las necesidades reales del cliente y estén en una forma que pueda entenderse, mantenida y utilizada por los desarrolladores del sistema.

Analizar los requisitos: Esto se hace para garantizar que estén bien definidos y que se ajusten a los criterios de un buen requisito (proporcionados a continuación).

Definir los requisitos de manera que signifiquen lo mismo para todas las partes interesadas: tenga en cuenta que cada grupo de partes interesadas puede tener una perspectiva significativamente diferente del sistema y de sus requisitos.

A veces, esto requiere invertir mucho tiempo en aprender un vocabulario especial o un léxico de proyecto. A menudo es necesario dedicar mucho tiempo y esfuerzo para lograr un entendimiento común.

Concretar los requisitos: Esto requiere incluir todo el detalle preciso de cada requisito para que pueda incluirse en un documento de especificaciones u otra documentación, dependiendo del tamaño del proyecto.

Priorizar los requisitos: No todos los requisitos tienen la misma importancia para los clientes y usuarios del sistema planificado. Algunos son críticos, otros de prioridad relativamente alta, algunos de prioridad normal o media y algunos incluso de menor prioridad. Es importante priorizar todos los requisitos porque nunca hay suficiente tiempo o dinero para hacer todo lo que nos gustaría hacer en nuestros sistemas desarrollados. Priorizar los requisitos brinda la oportunidad de abordar primero la prioridad más alta y posiblemente lanzar más adelante una versión de un producto que aborde

necesidades de menor prioridad. Priorizar ayuda a garantizar que se realice una cantidad adecuada de inversión para satisfacer las diversas necesidades de los clientes.¹

Requisitos derivados: Hay algunos requisitos que surgen debido al diseño de un sistema, pero no proporcionan un beneficio directo al usuario final. La necesidad de almacenamiento en disco podría deberse, por ejemplo, a la necesidad de almacenar una gran cantidad de datos.

Requisitos de partición: categorizamos los requisitos como aquellos que pueden cumplirse mediante hardware, software, capacitación y documentación, por ejemplo. A menudo, este proceso resulta más complejo de lo que anticipamos cuando más de una categoría satisface algunos requisitos.

Asignación de requisitos: Asignamos requisitos a diferentes subsistemas y componentes del sistema. Es posible que las asignaciones no siempre sean satisfechas por un solo subsistema o componente.

Seguimiento de requisitos: Necesitamos la capacidad de rastrear o rastrear en qué parte del sistema se cumple cada requisito, de modo que podamos verificar que se esté abordando cada requisito. Esto suele lograrse mediante el uso de una herramienta de requisitos automatizada.

Gestión de requisitos: debemos poder agregar, eliminar y modificar requisitos durante todas las fases de diseño, desarrollo, integración, prueba, implementación y operación del sistema. El repositorio de requisitos consta de un conjunto de artefactos y bases de datos. Se describe en el Capítulo 5.

Prueba y verificación de requisitos: este es el proceso de verificar requisitos, diseños, códigos, planes de prueba y productos del sistema para garantizar que se cumplan los requisitos.

Validación de requisitos: este es el proceso para confirmar que los requisitos reales se implementan en el sistema entregado. Se debe priorizar el orden de validación de los requisitos ya que hay una cantidad limitada de financiación disponible.

Inversión en el proceso de requisitos

La inversión promedio de la industria en el proceso de requisitos para un sistema típico es del 2% al 3% del costo total del proyecto. Debería ser evidente a partir de la

1. Véase AM Davis, "The Art of Requisitos Triage", Computer (marzo de 2003), para una discusión sobre el concepto de "clasificación de requisitos". Davis define la clasificación de requisitos como el proceso de determinar qué requisitos debe satisfacer un producto dado el tiempo y los recursos disponibles. Proporciona amplia orientación y sugerencias que ayudan a priorizar los requisitos. Se proporcionan tres estudios de casos de desarrollo de productos y 14 recomendaciones.

Según la información ya presentada, esta cantidad de inversión es inadecuada y, de hecho, es la causa fundamental del fracaso de muchos proyectos. Los datos de la Administración Nacional de Aeronáutica y del Espacio (NASA) de EE. UU. descritos en [2] brindan un mensaje claro y poderoso: proyectos que gastaron el promedio de la industria del 2% al 3% del costo/efuerzo total del proyecto en los requisitos (ciclo de vida completo) El proceso experimentó un sobrecoste del 80% al 200%, mientras que los proyectos que invirtieron del 8% al 14% del costo/efuerzo total del proyecto en el proceso de requisitos tuvieron sobrecostos del 0% al 50% [2, p. 9]. (Obviamente, nuestro objetivo no es tener excesos en absoluto; sin embargo, ¡es preferible un exceso menor que uno mayor!) Este libro describe cómo lograr un nivel apropiado de inversión en el proceso de requisitos y los beneficios asociados.

Un enfoque de proceso

Durante las últimas dos décadas, ha habido un debate considerable sobre el valor de un “enfoque de proceso”. Por enfoque de proceso me refiero a desarrollar y utilizar una descripción documentada (un diagrama de flujo de proceso y una descripción de proceso (DP) que lo acompaña) de un conjunto de actividades que resultan en la realización de una tarea o el logro de un resultado. Según mi experiencia, el uso de un enfoque basado en procesos tiene un gran valor:

Quienes apoyan la actividad documentan las acciones o actividades involucrados en hacer algo.

Una vez documentado, existe un entendimiento común (compartido) de lo que está involucrado.

El proceso documentado puede ser entendido por todos los involucrados.

Los involucrados, al tener un entendimiento común, pueden sugerir mejoras al proceso (permitiendo la mejora continua y empoderando a los involucrados para que aporten ideas para mejorar el proceso).

Se han desarrollado varios modelos de procesos generales. Por ejemplo, el Modelo de Madurez de Capacidad (CMM) [3] desarrollado por el Instituto de Ingeniería de Software (SEI) de la Universidad Carnegie Mellon a finales de los años 1980 proporciona un marco estándar de la industria para evaluar la madurez/ capacidad de un proceso de desarrollo. . La versión actual de este modelo se llama Capability Maturity Model Integration (CMMI) [4]. Su éxito se debe a la capacidad del modelo para discernir si el software se está desarrollando de forma más eficaz. Se puede saber si el esfuerzo de desarrollo es “mejor” o “peor” con el tiempo. Algunos PM pueden cuestionar el valor de la mejora de procesos, creyendo que desvía recursos de su responsabilidad principal de satisfacer las necesidades del cliente y que la mejora de procesos cuesta demasiado dinero. Los datos de la industria mantenidos por el SEI reflejan un retorno de la inversión (ROI) del proceso de 7:1.

mejora.² Otros datos de la industria reportan consistentemente entre 40% y 50% de retrabajo en proyectos de desarrollo. Reducir el retrabajo es un objetivo lucrativo para los esfuerzos de mejora de procesos. Reducir el retrabajo puede proporcionar los recursos para emprender iniciativas de mejora de procesos.

Además, se encuentran disponibles modelos de procesos de requisitos; por ejemplo, en mi libro anterior proporciono uno y está disponible en mi sitio web (www.ralphyoung.net); el modelo espiral para ingeniería de requisitos; y se proporciona un modelo en Dominar el proceso de requisitos [5].

El plan de requisitos

La AR debe desarrollar un plan de requisitos temprano, ya sea durante la fase de preparación de la propuesta o poco después de que se tome la decisión de continuar con un proyecto o tarea de desarrollo. El propósito del plan de requisitos es determinar y documentar cómo evolucionarán los requisitos reales y cómo se abordarán las actividades relacionadas con los requisitos en el ciclo de vida del sistema (enumeradas y descritas anteriormente). A continuación se muestra una lista de temas sugeridos para este plan y una descripción de cada tema:

Propósito (del plan de requerimientos): Fue definido en el punto anterior. párrafo.

Resumen del contrato/proyecto: se debe proporcionar un resumen de alto nivel de los objetivos del sistema o software. Esta sección se puede extraer de otros documentos, como un documento de visión y alcance , que puede haberse escrito previamente para describir la intención general.

Antecedentes: esta sección debe describir la situación que llevó a la decisión de desarrollar el sistema o software. Debe identificar los principales grupos de partes interesadas: aquellos que tienen interés en el sistema, como el cliente (la persona u organización que proporciona los fondos para pagar el proyecto o sus productos finales), diversas categorías de usuarios, desarrolladores y proveedores principales. .

Evolución de los requisitos: Se debe acordar un mecanismo entre los clientes/ usuarios y el equipo de desarrollo para revisar los requisitos establecidos y evolucionar los requisitos reales. Los clientes pueden resistirse a este esfuerzo, creyendo que ya tienen un “buen” conjunto de requisitos. La AR debe estar familiarizada con la experiencia de la industria en cuanto a cuántos proyectos han fracasado y cuántos más se han visto afectados grave y negativamente por no invertir en este paso crítico [1, p. 48]. Un mecanismo es una forma de hacer algo o de lograrlo.

2. Véase “Effects of Process Maturity on Development Effort” de BK Clark, Centro de Ingeniería de Software, Universidad del Sur de California, 1999, en www.ralphyoung.net/goodarticles, para obtener un excelente resumen de los beneficios de la mejora de procesos.

un resultado. El mecanismo recomendado para desarrollar los requisitos reales es un equipo cooperativo o conjunto compuesto por uno o varios representantes de los usuarios y un número similar de desarrolladores técnicamente competentes. Los miembros del equipo conjunto deben revisar los requisitos para garantizar que cumplan con los criterios de un buen requisito proporcionado en la Tabla 1.1. También se debe documentar la justificación de cada requisito (por qué es necesario). La experiencia de la industria es que al dar este único paso, se pueden eliminar hasta la mitad de los requisitos.

Roles y responsabilidades del personal del proyecto involucrado en actividades relacionadas con los requisitos: incluso en un proyecto pequeño, es probable que más de una persona participe en actividades relacionadas con los requisitos. Es útil aclarar y documentar estos roles, para que todos comprendan sus responsabilidades únicas y comunes. Por ejemplo, se debe designar a alguien para que proporcione capacitación sobre requisitos (el contenido de esta capacitación se describe en el Capítulo 5). Otra persona será responsable de la herramienta automatizada de requisitos. Otra persona más puede tener la responsabilidad de los procesos clave que se utilizarán en el proyecto, incluido el proceso de requisitos. Otro más puede ser responsable del diseño de la arquitectura (la estructura subyacente del sistema o software). Dado que los requisitos y la arquitectura se impactan entre sí, una práctica de requisitos recomendada es iterar los requisitos y la arquitectura repetidamente; esto da como resultado requisitos más sólidos y una arquitectura más robusta [1, págs. 131-158].

Tabla 1.1 Criterios de un buen requisito

Cada requisito individual debe ser
Necesario: Si el sistema puede satisfacer necesidades reales priorizadas sin el requisito, no es necesario.
Factible: El requisito es factible y se puede cumplir dentro del presupuesto y el cronograma.
Correcto: Los hechos relacionados con el requisito son exactos y es técnica y legalmente posible.
Conciso: el requisito se establece de forma sencilla.
Inequívoco: El requisito sólo puede interpretarse de una manera.
Completo: Se establecen todas las condiciones bajo las cuales se aplica el requisito, y expresa un todo idea o afirmación.
Consistente: No está en conflicto con otros requisitos.
Verificable: Se puede demostrar la implementación del requisito en el sistema.
Trazable: se puede rastrear el origen del requisito y se puede rastrearlo en todo el sistema. (por ejemplo, al diseño, código, prueba y documentación).
Asignado: El requisito se asigna a un componente del sistema diseñado.
Diseño independiente: No plantea una solución de implementación específica.
No redundante: no es un requisito duplicado.
Escrito usando la construcción estándar: el requisito se expresa como un imperativo usando "deberá".
Asignado un identificador único: Cada requisito tendrá un número de identificación único.
Sin cláusulas de escape: el lenguaje no debe incluir frases como "si", "cuándo", "pero", "excepto". "a menos que" y "aunque". El lenguaje no debe ser especulativo ni general (es decir, evitar expresiones como "normalmente", "generalmente", "a menudo", "normalmente" y "típicamente").

Definición del proceso de requisitos a utilizar: Como se señaló anteriormente, un proceso de requisitos documentado es esencial. Un proceso puede considerarse como un diagrama de flujo (que indica los pasos realizados y la persona u organización que realiza cada paso) acompañado de un PD narrativo que indica, por ejemplo, el nombre del proceso, sus clientes, los insumos para el proceso, los resultados del proceso, las tareas realizadas en el proceso, la persona u organización que realiza cada tarea y algunas medidas (métricas) que pueden usarse para evaluar la calidad de los productos producidos por el proceso y el desempeño del proceso. La experiencia demuestra que es una buena práctica involucrar a los principales actores de un proceso en su construcción. Este enfoque fomenta la comprensión, la integridad y la aceptación del proceso definido, así como el compromiso de utilizarlo.

Mecanismos, métodos, técnicas y herramientas a utilizar: a lo largo de este libro se describirán varios ejemplos de cada categoría. Obviamente, algunos son más apropiados en algunos casos que otros, y algunos son particularmente útiles en situaciones específicas. Se deben determinar y documentar los mecanismos, métodos, técnicas y herramientas específicos, y el equipo del proyecto debe estar familiarizado con los seleccionados y la justificación de su selección.

Integración de prácticas de requisitos efectivas y comprobadas: la experiencia ha demostrado que el uso de un conjunto de prácticas de requisitos efectivas y comprobadas puede marcar una gran diferencia en un proyecto [1]. Por ejemplo, ya se ha recomendado la práctica de invertir tiempo y esfuerzo en definir las necesidades reales del cliente. Las "mejores" prácticas de requisitos recomendadas se describirán a lo largo de este libro y se resumen en el Capítulo 6.

Seleccione y documente un conjunto de prácticas de requisitos que serán de gran utilidad para su proyecto.

Referencias: Existirá un conjunto de documentos que son referencias claves para el proceso de requisitos. Los ejemplos incluyen documentos que describen metas y objetivos del sistema, listas de requisitos de diferentes usuarios, estándares que el cliente ha especificado que se apliquen, políticas aplicables, etc. Se deben enumerar estas referencias y se debe indicar el lugar donde se puede acceder a cada una.

Estrategia recomendada: Con base en el análisis de la información anterior, se debe desarrollar y establecer una estrategia para aprovechar de manera óptima los aspectos del proyecto relacionados con los requisitos. Los elementos de la estrategia podrían incluir los siguientes:

La estrategia de asociación;3

3. El término "asociación" se utiliza a menudo para sugerir una relación de trabajo estrecha, coordinada y eficaz. Aquí me refiero a un proceso definido de esfuerzo de asociación en un proyecto. Le recomiendo que se familiarice con las referencias en [6] y que considere el uso del proceso de asociación. Puedes descubrir (como yo) que contiene uno de los secretos para proyectar éxito.

El "proceso inicial" que se utilizará (para comprender las necesidades reales del cliente y el entorno, comprender y documentar el alcance del proyecto, definir interfaces externas, definir los componentes del sistema y definir el esquema para la especificación del sistema);

Determinar qué impulsa los requisitos (regulaciones; especificaciones de nivel superior; estándares; políticas; sistemas y procesos existentes; restricciones, como costo, cronograma, viabilidad técnica; necesidades y expectativas de clientes y usuarios);

Definición de una política de requisitos del proyecto;

Definición del proceso de requisitos (diagrama de flujo y PD) (Se proporciona un proceso de requisitos de muestra en [1] y en mi sitio web (www.ralphyoung.net). Es posible que pueda utilizarlo para adaptar un proceso de requisitos para su entorno o proyecto.);

Mecanismos a utilizar (por ejemplo, el equipo conjunto y otros que se recomiendan en este libro);

Formación sobre los requisitos para el equipo del proyecto (incluido el cliente);

Selección de una herramienta de requisitos automatizada adecuada y cómo se utilizará;

Definición de la arquitectura de destino;

Planes para abordar requisitos nuevos y modificados (por ejemplo, uso de un mecanismo para controlarlos, así como versiones, lanzamientos y compilaciones);

Comprendión de los riesgos inherentes a los requisitos, ya que es probable que la falta de comprensión total de algunos requisitos cree riesgos importantes para el proyecto;

Definición de lineamientos para el desarrollo del sistema en base a los requerimientos. consideraciones de comentarios.

Apéndices: Estos podrían incluir lo siguiente:

Proceso de requisitos (diagramas de flujo y PD);

Enfoque del proceso de asociación [6];

Borrador de política de requisitos del proyecto;

Planes de acción y cronogramas para los esfuerzos necesarios (por ejemplo, selección de una herramienta de requisitos).

Factores que afectan sus decisiones profesionales

Le recomiendo que se reúna con su PM muy temprano, tal vez incluso antes de que finalice su asignación al proyecto. Discuta con él o ella las perspectivas relativas a los requisitos. Despues de digerir este libro y mi anterior

En primer lugar, debe tener un conocimiento suficiente de las prácticas de requisitos que le permitan concluir si puede ser eficaz en su función.

¿El PM cree que los requisitos, las prácticas de requisitos, invertir en el proceso de requisitos, controlar los cambios de requisitos y los nuevos requisitos y minimizar el retrabajo son importantes?

¿Sientes que él o ella te apoyará en las muchas funciones en las que potencialmente puedes contribuir al proyecto (ver Capítulo 2)?

¿Parece preocupado por las personas, por motivarlas, reconocer sus esfuerzos, empoderarlas y apoyarlas?

¿Tiene él o ella una buena reputación en la organización como PM?

¿Le preocupa el crecimiento personal y profesional?

¿Está dispuesto a delegar responsabilidades?

La cuestión es que estás a punto de dedicar una parte de tu vida profesional a un proyecto. Tómese el tiempo y el esfuerzo para asegurarse de que su tiempo estará bien empleado. Debe percibir que un nuevo puesto le brindará experiencias de aprendizaje, oportunidades para realizar contribuciones valiosas y necesarias, trabajar con pares a quienes respeta, obtener autosatisfacción y realización, y divertirse en el trabajo.

Un comentario sobre proyectos pequeños

Mucha gente siente que el enfoque que se utiliza en proyectos medianos y grandes es una guía inapropiada para proyectos pequeños: que las prácticas, políticas, mecanismos, métodos, técnicas y herramientas no se pueden aplicar. Mi experiencia es que el juicio profesional se puede utilizar para reducir y aplicar prácticas clave para lograr buenos resultados en proyectos pequeños. Animo a los miembros de proyectos, tareas o equipos pequeños a beneficiarse de lo que pueden aprender de las experiencias de proyectos más grandes adaptando el enfoque, en lugar de utilizar el tamaño pequeño como excusa para no aprovechar las lecciones de la industria. Consulte [7] para obtener información adicional.

Resumen

Este capítulo se ha centrado en la importancia de los requisitos y ha proporcionado una introducción al papel crítico del RA (las funciones del RA se detallan con más detalle en el próximo capítulo, y las habilidades y características de un RA eficaz se describen en el Capítulo 3). Del material presentado ya debería resultar evidente que existe un gran poder y efecto en aprovechar las actividades relacionadas con los requisitos en ingeniería e informática. Un alarmante 53% de la inversión de la industria en proyectos de desarrollo técnico es una víctima

de sobrecostos y proyectos fallidos. Los principales factores que contribuyen son la falta de aportes de los usuarios (13%), requisitos incompletos (12%) y requisitos cambiantes (12%).⁴ La comunidad de usuarios y, en particular, la gestión de proyectos no se dan cuenta del valor de invertir en el proceso de requisitos. . Sugiero que no está "bien" que un RA sea consciente de esto y no discuta las implicaciones con su primer ministro. Como profesional preocupado, usted tiene la responsabilidad de presentar estos hechos y su enfoque recomendado a su PM y pedirle que respalde un proceso de requisitos efectivo que incorpore prácticas de requisitos efectivas. Los RA, ingenieros y gerentes se encuentran en una posición estratégica para mejorar el desempeño de la industria. Este libro proporciona orientación enfocada y específica que puede tener enormes beneficios. Al aplicar el enfoque recomendado en este libro, puede tener un impacto muy positivo en su proyecto y organización.

Caso de estudio

Este primer estudio de caso informa sobre un taller que involucró discusiones facilitadas entre un grupo de PM sobre las principales razones por las que creían que los sistemas y proyectos de software tenían dificultades, según su experiencia. Estas son las principales razones informadas por un conjunto de PM:

1. Los requisitos del proyecto no son explícitos.
2. Los cambios en los requisitos se realizan/aceptan sin abordar los impactos concomitantes en costo, cronograma y calidad.
3. No se utiliza un proceso de requisitos.
4. No existe ningún mecanismo (como un equipo conjunto) para llegar a un acuerdo sobre la definición de los requisitos y gestionarlos a lo largo del ciclo de vida del proyecto.
5. Las necesidades "reales" de los clientes no están definidas.
6. No existe ningún mecanismo para mantener la comunicación entre los partes involucradas en el proyecto.
7. Los métodos, técnicas y herramientas conocidos, familiares y probados no son utilizado.
8. El cliente no participa como socio durante toda la vida del proyecto.

Le recomiendo que tenga en cuenta estas razones mientras digiere este libro. Averigüe qué podría hacer o recomendar para ayudar a superar estos problemas.

4. The Standish Group, The CHAOS Report (Dennis, MA: The Standish Group International, 1995). Consulte <https://secure.standishgroup.com/reports/reports.php?rid=1>.

Referencias

- [1] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.
- [2] Hooks, IF y KA Farry, Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos, Nueva York: AMACOM, 2001.
- [3] Paulk, MC, et al., Modelo de madurez de capacidad para software, versión 1.1, febrero de 1993, SEI, Universidad Carnegie-Mellon, Pittsburgh, PA, 1993.
- [4] Sitio web de CMMI en www.sei.cmu.edu/cmmi.
- [5] Robertson, S. y J. Robertson, Dominar el proceso de requisitos, Harlow, Reino Unido: Addison-Wesley, 1999.
- [6] Markert, C., "Partnering: Unleashing the Power of Teamwork", 2002, informe informativo disponible en Markert@facilitationcenter.com. Véase también Frank Carr et al., Partnering in Construction: A Practical Guide to Project Success, Chicago: American Bar Association Publishing, 1999.
- [7] Véase Paulk, MC, "Using the Software CMM with Good Judgment", ASQ Software Quality Professional 1(3) (junio de 1999): 19–29, en www.sei.cmu.edu/publications/articles/paulk/juicio.html.

Machine Translated by Google

CAPÍTULO

2

Contenido

[Funciones sugeridas de la AR](#)[Resumen](#)[Caso de estudio](#)[Referencias](#)

Las funciones de la RA

El Capítulo 1 enfatizó la importancia de los requisitos. Se observó que los clientes, gerentes y desarrolladores subestiman la ingeniería de requisitos. La RA se encuentra en una posición estratégica para mejorar las prácticas en uso en los proyectos y en la organización. El analista puede tener un impacto positivo en el éxito del proyecto y también facilitar los resultados de mejora de la organización al desempeñar varios roles. Hacer explícito el papel de la AR contribuye a que el proceso sea más fluido. El papel del RA puede vincularse fácilmente a los objetivos comerciales, como aumentar la satisfacción del cliente con los productos de trabajo entregados; reducir el tiempo de comercialización de los productos; cumplir con los objetivos de costos, cronograma y calidad; y utilizar los recursos humanos de la empresa de forma más eficaz. El papel de la RA debe ser comprendido y valorado en la mente de los PM y de las comunidades técnicas (tanto de informática como de ingeniería). La Tabla 2.1 resume las funciones de la AR, señalando las fases del ciclo de vida en las que se desempeña cada función.

Funciones sugeridas de la AR

1. Trabajar en colaboración con clientes, usuarios y arquitectos y diseñadores de sistemas para identificar los requisitos reales de un sistema planificado o esfuerzo de desarrollo de software para definir el problema que debe resolverse.

El concepto de requisitos reales se explicó en el Capítulo 1. La experiencia ha demostrado que el problema número uno en ingeniería de requisitos es no identificar los requisitos reales antes de iniciar las actividades de desarrollo del sistema. Cualquiera que haya tenido alguna experiencia en el desarrollo de sistemas o software estará de acuerdo en que identificar los requisitos reales es un problema importante. Con respecto a esta función, la AR debe crear conciencia sobre el problema y también proporcionar una

estrategia sugerida para superar el problema. Este es un ejemplo concreto de una situación que sabemos que se puede mejorar, pero la mayoría de las veces no actuamos en base a este conocimiento. Estamos impacientes por empezar con el llamado trabajo real de la programación. Nos contentamos con permitir que el esfuerzo de desarrollo continúe sin realizar un esfuerzo adicional para desarrollar los requisitos reales. Tenga en cuenta que he usado la palabra "evolucionar". Este trabajo implica más que identificar requisitos. La tarea esencial es utilizar los requisitos declarados articulados por los clientes y usuarios como base, combinar esto con una comprensión profunda de los objetivos comerciales e iterar para desarrollar requisitos que cumplan con los criterios de un buen requisito y aborden las necesidades reales priorizadas para el sistema o software. Las actividades involucradas en la realización de este trabajo incluyen las siguientes:

Identificar las necesidades manifestadas por los clientes y usuarios. Esto implica revisar lo escrito previamente sobre el sistema propuesto, entrevistar a clientes y usuarios, estudiar la legislación relevante, etc.

Estudiar los objetivos de negocio del esfuerzo propuesto.

Colaborar con clientes y usuarios en un entorno conjunto o cooperativo para analizar los requisitos establecidos, desarrollar mejores requisitos y priorizarlos (consulte las técnicas sugeridas a continuación).

Involucrar a los arquitectos de sistemas en el desarrollo de requisitos. La iteración del borrador o los requisitos propuestos dará como resultado una arquitectura candidata con mejores requisitos y una arquitectura más sólida. Por ejemplo, los sistemas deben poder adaptarse a las necesidades comerciales cambiantes. La arquitectura debe diseñarse y desarrollarse en consecuencia; de lo contrario, el sistema entregado pronto quedará obsoleto.

Utilizar una herramienta de requisitos automatizada sólida en la industria para respaldar este trabajo.

La AR debe trabajar dentro de la organización del proyecto para ganar el apoyo del PM y lograr el compromiso de invertir más tiempo y esfuerzo para desarrollar los requisitos reales. Esta es una gran oportunidad para que la AR asuma la responsabilidad y, aprovechando la experiencia de la industria, convenza a la gerencia de proyectos y a los desarrolladores para que inviertan más tiempo y esfuerzo en el proceso de requisitos. Afortunadamente, hay datos disponibles para ayudarnos a gestionar por hechos y no por intuición o la forma en que siempre hemos hecho las cosas. Consulte Prácticas de requisitos eficaces [1, p. 62] para estos datos.

Considere el uso de técnicas colaborativas de obtención de requisitos que funcionen bien en sesiones grupales. Ejemplos de buenas técnicas de obtención de requisitos son los talleres de requisitos, el software colaborativo electrónico o las herramientas electrónicas de desarrollo colaborativo, los diagramas de flujo de datos de alto nivel, los diagramas IDEF0 de alto nivel (especialmente para el modelado de negocios) y los diagramas de casos de uso de alto nivel (especialmente para distinguir requisitos que son

fueras del sistema versus el comportamiento esperado del sistema). Todos estos funcionan bien en una pizarra, son fáciles de entender y permiten que todos presentes participen. Consulte Gestión de requisitos de software: un enfoque unificado [2] de Dean Leffingwell y Don Widrig para obtener buenas discusiones sobre estos y Otras técnicas y cómo utilizarlas. David Hay proporciona una útil comparación de técnicas que se pueden utilizar en el análisis de requisitos: desde el negocio.

Vistas a la Arquitectura (ver [3, p. 194] y la discusión anterior).

2. Trabajar eficazmente con clientes y usuarios para gestionar requisitos nuevos y modificados para que el proyecto permanezca bajo control. Instalar un mecanismo para controlar cambios.

El siguiente problema más serio en ingeniería de requisitos (después de no identificar los requisitos reales) es la falta de control de los requisitos que se identifican después de que comienza el desarrollo (programación) del sistema, tanto nuevos requisitos y cambios a los requisitos existentes. Aquí distinguimos entre los requisitos críticos (aquellos que tendrían un impacto en el costo, programa, o el esfuerzo de desarrollo si se modifica) y requisitos no críticos, como un requisito derivado que define aún más el sistema que se está construido, pero sirve para aclarar un requisito de nivel superior y no afecta costo, cronograma o funcionalidad. Todas las partes interesadas deberían acoger con agrado un requisito de “no impacto” que aclare aún más el sistema.

Nuevamente, contamos con datos de la experiencia de la industria para guiar nuestras acciones: un cambio del 20% en los requisitos resultará en una duplicación de los costos de desarrollo del proyecto [4]. Por lo tanto, es fundamental que se establezca un mecanismo lugar para evaluar y adjudicar cambios a los requisitos. Sin un mecanismo eficaz para controlar los cambios en los requisitos, el proyecto pronto se verá interrumpido.

fuera de control en términos de cronograma y costo. Se deben hacer varias cosas:

Se debe resaltar la importancia de controlar los cambios en los requisitos. explicado a los clientes, usuarios y desarrolladores para que la asociación Se mantiene el compromiso con el éxito del proyecto.

Los desarrolladores deben estar capacitados para no aceptar requisitos no autorizados. cambios. Todas las solicitudes de cambios, por triviales que sean, deben canalizarse a través del mecanismo de control de cambios.

El mecanismo de control de cambios debe ser un equipo conjunto que incluya tomadores de decisiones empoderados que representan al cliente y al desarrollador. El equipo conjunto debe reunirse con suficiente frecuencia para tener un número razonable de solicitudes de cambio para considerar. Una métrica objetivo del 0,5% Se recomienda la volatilidad de los requisitos para guiar las decisiones tomadas por el equipo conjunto una vez que se haya establecido una línea base de requisitos validados.¹ “¡Vaya!”, dirás, “¡eso no es mucho!” ¡Bien! Este es otro

1. El Capítulo 10 de Prácticas efectivas de requisitos proporciona varias ideas, sugerencias y recomendaciones para controlar los cambios de requisitos.

razón para invertir el tiempo necesario para desarrollar los requisitos reales antes de iniciar las actividades de desarrollo.

Al asociarse con su cliente, desarrolle formas de afrontar el cambio. Sabemos que el mundo está cambiando mientras desarrollamos el sistema. ¿Cuáles son algunas formas de abordar esto sin poner en peligro el éxito del proyecto?

Considere la posibilidad de utilizar lanzamientos, versiones y actualizaciones. Incrementos de paquetes de actualizaciones de requisitos y cambios en versiones posteriores o actualizaciones del sistema.

Asegúrese de que su contrato prevea tiempo y presupuesto adicionales para todos los cambios. Este es un mecanismo para mantener buenas relaciones durante todo el trabajo por contrato: asociarse para lograr el éxito. Los cambios cuestan tiempo y dinero. Esto debe reconocerse desde el principio y reflejarse en el contrato.

3. Esté alerta a las nuevas tecnologías que pueden ayudar.

Una función que a menudo no se utiliza es la de asesorar a nuestros clientes sobre la evolución de la tecnología. Si bien esto no es responsabilidad exclusiva de los analistas o ingenieros de requisitos, muchos involucrados en el desarrollo de sistemas para clientes harían bien en dedicar tiempo y esfuerzo adicionales a aprender sobre nuevas tecnologías y cómo se pueden aplicar a nuestro trabajo.

Los clientes suelen centrarse en lo que debe hacer el sistema. Podemos brindarles un mejor servicio si estamos familiarizados con las tecnologías en evolución que mejoran la forma en que se diseña el sistema necesario. Esto sugiere que los RA se beneficiarán de que los diseñadores de sistemas revisen sus productos de trabajo. Simultáneamente con la elaboración de requisitos, involucre a un pequeño equipo de diseñadores para revisar los requisitos reales en cuanto a costos, cronograma, tecnología e impactos de riesgo. Utilice estudios comerciales (el proceso de resolución de análisis de decisiones (DAR) en la terminología CMMI) para desarrollar alternativas. Mantenga al cliente involucrado en estas actividades, de modo que cuando surjan oportunidades, el cliente esté allí para colaborar con usted y hacer recomendaciones para tomar decisiones. Una excelente referencia que describe el proceso de utilización de nuevas tecnologías es Dif-fusion of Innovations de Everett M. Rogers (4^a ed.) [5].

4. Facilitar la reutilización de artefactos del proyecto y lograr la repetibilidad.

Ha habido mucha discusión en la literatura de la industria sobre la reutilización.

Reutilizar tiene dos significados: (1) tomar el objeto X (por ejemplo, un objeto, subrutina o software COTS) que fue realizado por Y y usarlo directamente en otro proyecto, y (2) adaptar2 un producto de trabajo desarrollado (una especificación , un plan o proceso, por ejemplo). Muchas organizaciones han invertido en estrategias de reutilización sólo para concluir que no son viables ni prácticas. Otros desconfían de

2. Por "adaptación" nos referimos a modificar, extraer piezas, elaborar o adaptar un proceso o documento para otro uso. La reutilización de artefactos personalizados ahorra tiempo y dinero y es una ventaja de un enfoque orientado a procesos.

reutilizar porque creen que excluye soluciones sin precedentes e incorpora los errores de los productos de trabajo reutilizados.

Podemos considerar los requisitos en sí mismos como artefactos reutilizables. Los libros que analizan patrones de requisitos reutilizables incluyen Patrones de modelo de datos: convenciones de pensamiento [6] (para un punto de vista relacional) de David C. Hay, Patrones de análisis (para un punto de vista orientado a objetos) de Martin Fowler [7] y Patrones de diseño de Eric Gamma, et al. [8]. Los marcos de problemas de Michael Jackson (descritos en su libro del mismo nombre [9]) son en esencia patrones de requisitos altamente abstractos que pueden conectarse, anidarse e integrarse en modelos del mundo real. La cuestión es que muchos requisitos no son únicos; ya han sido identificados en el entorno y el espacio problemático de otra persona.

En mis actividades de escritura he descubierto que comenzar con un producto de trabajo de ejemplo me da ideas sobre el formato, la estructura, el contenido y los recursos para consultar o contactar. Un ejemplo de producto de trabajo que quizás desee considerar es un plan de requisitos. Como se enfatizó en el Capítulo 1, propongo el desarrollo de un plan de requisitos para cualquier esfuerzo de desarrollo de sistema o software. Esta idea puede ser nueva para usted y sería muy útil e instructivo revisar una desarrollada anteriormente para considerar su valor potencial para su trabajo. Otro ejemplo de mi experiencia es la reutilización de procesos documentados. Si la organización u otro proyecto tiene un proceso documentado para hacer algo, ¿por qué no adaptarlo según sea necesario y luego reutilizarlo, en lugar de crear su propio proceso? Otros que han realizado el proceso en la práctica han incorporado su experiencia y las lecciones que han aprendido al utilizarlo. Relacionado con esto está el valor de las revisiones por pares. Abogo por una revisión por pares de cada producto de trabajo. (El alcance de la revisión por pares (el número de personas solicitadas para revisar el producto de trabajo y el tiempo invertido para realizar la revisión por pares e informar sobre defectos y hacer sugerencias) es una función de la importancia del producto de trabajo.) Si uno puede reutilizar el proceso de revisión por pares y las listas de verificación de otra organización, lo que proporciona un punto de partida para diseñar, aceptar, implementar, implementar e institucionalizar el proceso.

Un ejemplo de reutilización de

procesos Al enseñar cursos y tutoriales sobre requisitos, siempre me interesa saber cuántos de los participantes están utilizando un proceso de requisitos documentado en su proyecto o en su organización. Normalmente, esto resulta ser entre el 15% y el 20% de los participantes. En Prácticas efectivas de requisitos [1, págs. 110-118] se proporciona un proceso de requisitos de muestra . Este proceso se ha adaptado, implementado e implementado en más de 50 proyectos. Su integración con el proceso de arquitectura del sistema se describe más adelante en el libro [1, págs. 136-146].

Sugerencia: Adapte este proceso de requisitos de muestra para su proyecto u organización. Involucrar a las partes interesadas para realizar los cambios que mejor satisfagan sus necesidades. Proporcione diagramas de flujo y PD narrativos como se describe en Prácticas de requisitos eficaces. Actualizar periódicamente el proceso documentado con ideas y sugerencias de mejora continua.

5. Ayudar al proyecto y a sus clientes a visualizar una ruta de crecimiento desde el primer lanzamiento o versión de un producto a través de un conjunto de lanzamientos por etapas hasta el sistema o producto definitivo.

Esta función está relacionada con la función 3. La RA puede desempeñar una función importante y valiosa al ayudar a los clientes a visualizar y desarrollar una serie de lanzamientos o versiones de productos. Este enfoque es particularmente apropiado en la situación en la que los requisitos no se comprenden bien desde el principio o los requisitos están cambiando rápidamente. Esto sugiere que se debe utilizar un "enfoque de desarrollo incremental", en el que el sistema completo se implemente durante un período de tiempo mediante incrementos de la funcionalidad entregada. En cierto sentido, ningún sistema está terminado, por lo que tenemos que ayudar a todos a ver el desarrollo del sistema como un viaje. Independientemente de la metodología de desarrollo del sistema utilizada (en cascada, incremental, en espiral, evolutiva, etc.), tiene que haber un proceso acordado para gestionar los cambios y determinar el alcance de los proyectos individuales. No importa cuánta discusión y pruebas se realicen, faltan algunos requisitos que no se descubrirán hasta que el sistema esté en producción.

6. Asesorar al proyecto (y al cliente) sobre los métodos, técnicas y herramientas automatizadas que están disponibles para respaldar mejor el trabajo y las actividades del proyecto relacionados con los requisitos.

Éste es un papel importante. La experiencia ha demostrado que los métodos y técnicas varían en su aplicabilidad y eficacia y que a menudo las herramientas automatizadas adquiridas por proyectos y organizaciones no se utilizan o están subutilizadas. El Capítulo 11 de Prácticas efectivas de requisitos [1] informa sobre la experiencia de la industria y proporciona varias recomendaciones. El Capítulo 8 de Prácticas efectivas de requisitos [1] recomienda que los métodos y técnicas que se utilizan en un proyecto sean familiares para los participantes del proyecto y estén probados en sus respectivas industrias. No es aconsejable emprender un proyecto con métodos y técnicas no probados y desconocidos. El trabajo de desarrollo es lo suficientemente desafiante sin introducir la complejidad de métodos o técnicas que no son familiares y que no se han utilizado con éxito en proyectos anteriores de la organización. A nivel de proyecto, el equipo debe apegarse a las herramientas, procesos y técnicas con las que sus miembros están familiarizados. A nivel organizacional, el proyecto debe intentar utilizar las herramientas, procesos y técnicas que son conocidos y probados en la organización.

Cuando los contratistas participan en un esfuerzo existente, deben adaptarse a las herramientas que el cliente ya tiene implementadas (suponiendo que estén trabajando de manera efectiva). Si los últimos cinco proyectos se realizaron con la herramienta X y todos están satisfechos con la utilidad de la herramienta, cuando llegue, habrá buenas razones para usarla. Tenga en cuenta que puede estar involucrado un problema de recursos. Idealmente, un RA sería un recurso aprovechado, que pasaría de un proyecto a otro y llevaría consigo su experiencia. Sin embargo, a menudo en la práctica, se crea un equipo de proyecto (o ya existe) y alguien del equipo actual con conocimiento del dominio tiene la tarea de ser el RA. Si bien existen técnicas y herramientas probadas y verdaderas, es posible que esta persona no esté familiarizada con ellas, lo que requiere una curva de aprendizaje larga y a veces dolorosa, con desventajas significativas para el

proyecto. Esto aboga por que la organización proporcione un conjunto de AR experimentados que proporcionen un alto retorno de la inversión realizada para identificarlos, capacitarlos y brindarles experiencia.

También recomiendo desafiar las instrucciones del cliente para utilizar métodos o técnicas específicas que no sean familiares para el equipo del proyecto o que no hayan sido probadas previamente en la práctica. Por ejemplo, un cliente podría indicar que se emplee un enfoque de desarrollo orientado a objetos (OO) (consulte [10] para obtener pautas detalladas sobre este tema) o que se utilice una herramienta automatizada o un conjunto de herramientas en particular. Es valioso estar en condiciones de poder asesorar a su proyecto y a su cliente sobre los métodos, técnicas y herramientas automatizadas que mejor respaldarán la situación de desarrollo específica. Aproveche la experiencia de la industria y no pretenda que “todo saldrá bien”.

7. Utilice métricas para medir, rastrear y controlar las actividades de trabajo del proyecto relacionadas con los requisitos, actividades y resultados.

La literatura de la industria sobre métricas es amplia. Calculo que quizás el 20% proporciona consejos útiles. Es fácil caer en una situación de realizar actividades de medición por sí mismas, en lugar de ayudar a evaluar el trabajo del proyecto y tomar acciones correctivas. Recomiendo utilizar algunas métricas útiles. He desarrollado el siguiente axioma en mi trabajo durante el

años:

Las cosas que se miden y rastrean y a las que la gerencia presta atención son las que mejoran.

Esto sugiere que no es suficiente tener algunas métricas útiles: se debe realizar un seguimiento de ellas y la dirección debe utilizarlas para guiar las decisiones del proyecto.

Existe un conjunto de medidas o métricas que todos los proyectos deben utilizar. Consulte Prácticas efectivas de requisitos [1, págs. 255–261] para obtener sugerencias específicas.

Existe otro nivel de sofisticación que deberían utilizar los proyectos y organizaciones maduros. Tal como se utiliza aquí, “maduro” significa que los procesos han sido definidos, documentados, implementados, utilizados, institucionalizados y mejorados continuamente durante un período de al menos dos a cuatro años. Esto implica la gestión cuantitativa (QM) de costos, cronograma, calidad y métricas y líneas de base de procesos en apoyo de objetivos comerciales específicos. Es gratificante ver cómo los proyectos y las organizaciones pasan de una situación en la que la calidad de la calidad no se comprende bien a una en la que la calidad de la calidad se utiliza eficazmente para lograr los objetivos empresariales. Esto resulta especialmente satisfactorio para los ingenieros de procesos, porque los ejecutivos pueden ver de primera mano el valor de la mejora de procesos para satisfacer las necesidades del negocio.

8. Ser capaz de facilitar discusiones y mediar en conflictos.

Este papel enfatiza las “habilidades interpersonales” de la RA. Hemos aprendido que estar bien cualificado técnicamente es importante, pero que también es necesario tener habilidades fuertes,

Habilidades interpersonales bien refinadas. La experiencia ha demostrado que dos cabezas piensan mejor más de uno, cada vez que nos tomamos el tiempo para explorar ideas y enfoques con otros, ¡obtenemos ideas y enfoques aún mejores! Ergo, podemos dejar el punto de vista de que "sabemos más". Y podemos hacer un gran uso de este principio convirtiéndonos en buenos facilitadores y mediadores. Hay cursos disponibles para ayudar (por ejemplo, habilidades de negociación, formación de equipos, comunicaciones, relaciones y liderazgo). Se puede ganar mucho practicando estas habilidades en nuestro trabajo diario. Tener una perspectiva en la que todos salgan ganando es útil; de hecho, Barry Boehm et al. hemos desarrollado un desarrollo de requisitos beneficioso para todos enfoque en el trabajo realizado en la Universidad del Sur de California. Ver http://sunset.usc.edu/research/WINWIN/winwin_main.html.

9. Estudiar el dominio del área en la que se está utilizando el sistema o software.

Ser capaz de captar, abstraer y expresar ideas rápidamente en el idioma de los usuarios. Si la RA no comprende el dominio del usuario casi tan bien como los usuarios, corre el riesgo de limitar su papel al de un tomador de órdenes. he visto diferentes van y vienen grupos cuya especialidad era la comunicación, la creación de consenso, etc. Poblar esos grupos era un conjunto de personas que estaban facilitadores capacitados, pero que no eran técnicamente competentes. Se movieron de proyecto en proyecto con tanta frecuencia que nunca lograron ningún logro profundo. comprensión del dominio. Por ejemplo, ¿qué pasa si, en un proyecto de comunicaciones de red, la única manera en que un RA puede explicar cualquier concepto a los usuarios es mediante ¿Dando analogías con la construcción de aviones militares? Respuesta: eficacia y credibilidad reducidas.

Resumen

La AR desempeña varias funciones importantes en un proyecto y en una organización. En este capítulo se identificaron y describieron nueve roles importantes. El Los dos primeros son primordiales y esenciales para el éxito del proyecto. En consecuencia, estudiar estos, dominarlos y ayudar a su proyecto y organización a adoptar, implementar e institucionalizar prácticas relacionadas. Las organizaciones deberían considerar tomar medidas específicas para desarrollar y aprovechar sus AR, tales como (1) garantizar que se asignen AR experimentados a cada proyecto; (2) proporcionar formación adecuada a los RA; (3) asignar AR experimentados a asesorar a nuevos empleados, RA junior y pasantes; y (4) tener un grupo de trabajo sobre requisitos organizacionales para compartir experiencia y proporcionar una recurso a la organización. El RA debe ser una persona capacitada, experimentada y intérprete fuerte. Lamentablemente, he visto muchos casos en los que el nuevo Se envía al empleado o al pasante de verano "para cumplir con los requisitos". El El papel de la AR debe ser comprendido y valorado en la mente de los PM y la comunidad técnica. En este punto, es posible que se sienta abrumado sus responsabilidades, como sugiere la Figura 2.1. Tenga la seguridad de que con estudio y experiencia, usted brindará una contribución muy positiva a los esfuerzos que ¡tu apoyo!



Figura 2.1 Los desafíos de la AR.

Caso de estudio

Esta es la historia de un proyecto que fracasó porque ni el cliente ni el contratista supieron cómo gestionar los requisitos. Es un ejemplo negativo.

Aunque las personas involucradas eran profesionales y tenían buenas intenciones, las cosas salieron terriblemente mal porque no se aplicaron prácticas de requisitos efectivas.

El enfoque del proyecto se utilizó con una frecuencia alarmante y se puede caracterizar como "comience a programar y descubriremos lo que quieren a medida que avanzamos". El cliente, una organización militar, le entregó al contratista un montón de reglas y regulaciones ("regs") diciendo: "Estos son los requisitos". Los programadores, todos empleados in situ de un contratista, estaban listos para empezar, y lo hicieron. Los representantes de la organización militar conocían el proyecto, pero no participaron en él hasta que llegó el momento de revisar el código terminado.

Mientras se escribía el código, el contratista se comprometió a convertir las normas en un conjunto de declaraciones obligatorias. Esto se hizo fiel y minuciosamente, pero a medida que surgió el código, se descubrió que la verificación de las declaraciones debe (compararlas con partes de los diferentes módulos de código) era prácticamente imposible. Simplemente no hicieron mapas.

Otra complicación encontrada fue una completa interrupción de la comunicación entre el contratista y el subcontratista que elaboraba el código. No es que no se entendieran; simplemente no se comunicaron. El subcontratista consideró cualquier consulta como una interferencia y, en una atmósfera de hostilidad, la comunicación simplemente se cortó.

Cuando el cliente revisó el código, se escuchó a los representantes militares decir una y otra vez: "No, eso es lo que hacemos, pero eso es lo que hacemos".

no cómo lo hacemos". Y cuando el contratista probó los módulos, fallaron repetidamente. Las cosas correctas se habían escrito de manera incorrecta y no funcionaron.

Después de meses de lucha, el primero de unos 20 módulos estaba casi listo para su lanzamiento. Todavía estaba un poco inestable y mucha gente no estaba contenta con él, pero estaba cerca de ser aceptado desde la perspectiva del contratista. Sin embargo, el cliente se dio por vencido porque las relaciones entre las dos partes se habían roto durante el período de desarrollo del código. No sólo el proceso estaba roto, sino que también la plataforma y el sistema operativo estaban obsoletos e inadecuados. Se abandonaron tres millones de líneas de código, se desechó el hardware y se empezó de nuevo todo el proyecto.

¿Qué había salido mal?

No hubo ninguna asociación entre el cliente, el contratista y subcontratista.

No hubo comunicación.

No había una atmósfera de respeto mutuo.

No existía un plan de requisitos viable.

No existía ningún mecanismo para la resolución conjunta de problemas.

El proyecto se inició de nuevo, en otra plataforma, en otro idioma y con una combinación diferente de subcontratistas. Se utilizó un conjunto mejorado de prácticas de requisitos, que incluían lo siguiente:

Mecanismos (similares al equipo conjunto analizado en el Capítulo 1) para facilitar la asociación, identificar necesidades y requisitos reales y priorizar los requisitos;

Un enfoque que involucró a los usuarios en el esfuerzo de desarrollo, proporcionó colaboración con ellos y obtuvo la aceptación de los usuarios en el enfoque del proyecto;

Uso de métodos que incluyan casos de uso que facilitaron la comprensión y comunicación efectiva de las necesidades y requisitos de los usuarios;

Incorporar tecnología adecuada y actualizada que sirva mejor el cliente y los usuarios.

Referencias

[1] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.

[2] Leffingwell, D. y D. Widrig, Gestión de requisitos de software: un enfoque unificado, Reading, MA: Addison-Wesley, 2000.

[3] Hay, DC, Análisis de requisitos: de la visión empresarial a la arquitectura, Upper Saddle River, Nueva Jersey: Prentice Hall, 2003.

- [4] Hooks, I., "Redacción de buenos requisitos: un tutorial de un día", patrocinado por el capítulo del Área Metropolitana de Washington (WMA) del Consejo Internacional de Ingeniería de Sistemas (INCOSE), McLean, VA, Compliance Automation, Inc. , Junio de 1997.
- [5] Rogers, EM, Difusión de innovaciones, 4^a ed., Nueva York: The Free Press, 1995.
- [6] Hay, DC, Patrones de modelos de datos: convenciones de pensamiento, Nueva York: Dorset House, 1996.
- [7] Fowler, M., Patrones de análisis: modelos de objetos reutilizables, Reading, MA: Addison-Wesley, 1996.
- [8] Gamma, E., et al., Patrones de diseño, Reading, MA: Addison-Wesley, 1995.
- [9] Jackson, M., Marcos de problemas: análisis y estructuración del desarrollo de software Problemas, Londres, Reino Unido: Addison-Wesley, 2001.
- [10] Webster, BF, Errores del desarrollo orientado a objetos, Nueva York: Hungry Minds, Inc., 1995.

CAPÍTULO

3

Contenido

[Habilidades de la RA](#)[Características de un eficaz](#)[Resumen](#)[Caso de estudio](#)[Referencias](#)

Habilidades y características de un RA efectiva

Este capítulo describe las habilidades y características deseadas de un RA eficaz.

Como se destacó en el capítulo anterior, la AR cumple varias funciones críticas en el proyecto. En muchos proyectos, el RA es una persona a tiempo parcial que también se desempeña como PM, gerente de producto, ingeniero de sistemas, desarrollador o en alguna otra capacidad. En otros proyectos, puede haber un RA a tiempo completo o incluso varios RA y un administrador de requisitos. El tamaño del proyecto y la complejidad percibida de las actividades necesarias relacionadas con los requisitos, así como la financiación disponible, son los principales determinantes del número de RA y sus niveles de habilidades necesarios. Las funciones del RA pueden dividirse entre aquellos disponibles para realizar el trabajo necesario, considerando también las habilidades, los intereses y las necesidades de desarrollo deseadas actuales. Cualquiera que sea la situación, las AR deben considerarse recursos clave, capaces de contribuir al proyecto en las funciones descritas en el capítulo anterior. La RA requiere una combinación única de habilidades que refleje el conocimiento y la orientación del mundo real, así como la capacidad de interpretar y satisfacer las intenciones de los clientes, usuarios y directivos. Algunas de las habilidades son intrínsecas a la forma en que trabaja un individuo (como las habilidades analíticas e interpersonales) y otras se aprenden (por ejemplo, las habilidades de facilitación).

Habilidades de la RA

Como marco para este capítulo, consulte la Tabla 3.1, Matriz de habilidades de RA.¹ Se proporciona una lista de habilidades de RA. Tres niveles de RA son mostrado:

1. Nuestro agradecimiento al RA Michael Davis de Northrop Grumman IT por proporcionar este artefacto. La responsabilidad por las modificaciones al original. La versión es mía.

Tabla 3.1 Matriz de Habilidades de RA

Línea Matriz de habilidades de Number RA	Referencia	Entrada/ Júnior- Nivel			Nivel medio Analista	Nivel superior Analista
		Analista	Nivel medio	Nivel superior		
1 Tipos de requisitos	Cap. 4	k	X		X	
2 Criterios de un buen requisito.	Cap. 1	k	X		X	
3 Implicación del cliente/usuario con equipo conjunto de requisitos	Cap. 1	k	X		X	
4 Identificar necesidades reales (de los requisitos indicados)	Cap. 1	k	X		X	
5 Anticipar y controlar cambios de requisitos	Cap. 1	k	X		X	
6 Herramientas de ofimática	Tutoriales	X	X		X	
7 Referencias relativas requisitos (libros, artículos, estándares)	Ver Bibliografía		KXX			
8 Atributos de requisitos	Cap. 5	k	X		X	
9 Línea base de requisitos	Cap. 6	k	X		X	
10 Formación en ingeniería de sistemas. (por ejemplo, ciclos de vida, gestión de riesgos)	Cap. 5	k	X		X	
11 Requisitos justificación/razonamiento	Cap. 5	k	X		X	
12 Herramientas de gestión de requisitos (p. ej., PUERTAS, RequisitePro)	Cap. 5	k	X		X	
13 Requisitos para revisión/inspección/recorrido	Cap. 5	k	X		X	
14 Sintaxis de requisitos	WBR, cap. 7K			X		X
15 Trazabilidad de requisitos	Cap. 5	k	X		X	
detalles Verificación de requisitos y validación (V&V)	Cap. 5	k	X		X	
17 Nivel de sistema/subsistema/software requisitos	Cap. 5	k	X		X	
18 Desarrollar y utilizar métricas para requisitos actividades/procesos	Cap. 2	k	X		X	
19 Redacción técnica de requisitos. entregables (RTM, SRS, IRS)	Cap. 4	k	X		X	
20 Desarrollo, implementación y uso de procesos de requisitos	Cap. 5		k		X	
21 Tutorial de familiaridad con Microsoft Project			k		X	
22 control de calidad de los requisitos	Cap. 9		k		X	
23 Asignación de requisitos (a componentes, aplicaciones, paquetes)	Cap. 4		k		X	
24 Requisitos de control de cambios y notificación de cambio	Cap. 6		k		X	
25 Repositorio de requisitos	Cap. 5		k		X	
26 Errores de requisitos (faltantes, incorrecto, inviable, fuera de alcance)	Cap. 6		k		X	
27 Notificación de defectos de requisitos	Cap. 6		k		X	
28 Difusión de requisitos a clientes/usuarios/desarrolladores/probadores	Cap. 4		k		X	

Tabla 3.1 Matriz de Habilidades de RA (continuación)

Línea			Entrada/ Júnior- Nivel	Nivel medio	Nivel superior
	Matriz de habilidades de Number RA	Referencia	Analista	Analista	Analista
29	Elicitación de requisitos	Cap. 5	k	X	
30	Identificación de requisitos	Cap. 5	k	X	
31	Desarrollo de casos de uso (con cliente/usuario y basado en el usuario guías)	Cap. 7	k	X	
32	Requisitos en cliente/usuario proceso de toma de decisiones	Cap. 1	k	X	
33	Interacción de requisitos con CM Ch. 6		X	X	
34	Negociación de requisitos	SL, EG1	X	X	
35	Propiedad de requisitos	WBR, EG2	X	X	
36	Priorización de requisitos	Cap. 5	X	X	
37	junta de revisión de requisitos (RRB)/junta de revisión de configuración (CRB)/tablero de control de configuración (CCB)	Cap. 7	X	X	
38	Requisitos orden aproximado de magnitud (ROM) costos	Cap. 7	X	X	
39	Especificaciones de requisitos	Cap. 7	X	X	
40	Evaluación de requisitos para riesgos.	Cap. 7		X	
41	Entrenando los requisitos procesos	Cap. 5		X	
42	Estimación del impacto de los requisitos. (IE) tabla	Gilb		X	

Conocimiento de = K Experiencia con = X

Referencias:

REH = Young, RR, Manual de ingeniería de requisitos, Norwood, MA: Artech House, 2004.

WBR = Alexander, IF y R. Stevens, Redactando mejores requisitos, Boston: Addison-Wesley, 2002.

SL = Lauesen, S. Requisitos de software: estilos y técnicas, págs. 346–347.

EG1 = Gottesdiener, E., Requisitos por colaboración: talleres para definir necesidades, Reading, MA: Addison-Wesley, 2002, págs. 122-128.

EG2 = Gottesdiener, E., Requisitos por colaboración: talleres para definir necesidades, Reading, MA: Addison-Wesley, 2002, págs. 89–94.

Gilb: Consulte el material en www.result-planning.com.

1. Analista de nivel inicial/junior;
2. Analista de nivel medio;
3. Analista de nivel superior.

En esta tabla se utiliza una “K” para sugerir que se necesita conocimiento de la habilidad en un nivel particular de experiencia del analista. Una “X” sugiere que la experiencia en usar la habilidad es necesario en un nivel particular. Se proporciona un mapeo para

secciones de este libro o de otras fuentes donde se aborda cada habilidad. Addi-

En esos lugares se proporcionan referencias adicionales.

Como ocurre con cualquier marco o modelo, utilice esta matriz como guía, no como especificación. Le ayudará a evaluar su idoneidad para un puesto en el proyecto y Proporcionar una guía y recursos para fortalecer y mejorar sus habilidades.

Evidentemente se podrían añadir otras habilidades. Sin embargo, la matriz sirve como guía y sugiere que hay mucho que aprender para ser un AR eficaz.

Una cosa es leer sobre (adquirir conocimientos sobre) habilidades y otra muy distinta.

Lo importante es adquirir experiencia en la aplicación de las habilidades en un entorno de proyecto. que involucra a clientes y usuarios reales.

Un analista junior o principiante (aquellos con menos de dos años de experiencia) debe estar familiarizado con lo siguiente:

Los tipos de requisitos (descritos en detalle en el próximo capítulo);

Los criterios de un buen requisito (establecidos en el Capítulo 1);

Herramientas de automatización de oficina [por ejemplo, Microsoft (MS) Office o Corel WordPer-suite perfecta];

El concepto de utilizar un proceso de requisitos;

Algunas de las referencias relativas a actividades relacionadas con requisitos;

El propósito de la verificación de requisitos, etc.

Ella debe entender que se debe proporcionar una justificación para cada Requisito (por qué el requisito es necesario en el sistema o software).

Un analista de nivel medio (aquellos con dos a cuatro años de experiencia) debería tener conocimiento de más aspectos y actividades de la ingeniería de requisitos, junto con experiencia adicional en la aplicación de este conocimiento.

En lo alto de esta lista se encuentran las actividades de requisitos que involucran a clientes y usuarios. (como el concepto de un equipo conjunto), utilizando un proceso de requisitos, y familiaridad con una herramienta de requisitos sólida en la industria. El analista de nivel medio debe ser competente en revisiones e inspecciones por pares y debe asegurarse de que todos los productos de su propio trabajo sean revisados por pares. Ella debería comprender el valor de la trazabilidad bidireccional de los requisitos y ser aprender a desarrollar una matriz de trazabilidad de requisitos (RTM).

Un analista de nivel superior (aquellos con cinco o más años de experiencia realizando actividades relacionadas con requisitos) debe tener conocimientos de y experiencia en el uso de todas las habilidades de la matriz. Ella debería ser familiarizado con todos los roles descritos en el capítulo anterior y Habilidades y características interpersonales bien desarrolladas, como se describe más adelante en Este capítulo. Debe comprender el valor y la importancia del control de calidad independiente y tener un conocimiento profundo de las actividades de CM. Ella debería ser capaz de recomendar y utilizar métricas de requisitos y poder aplicar métricas a los procesos de requisitos. Debería poder impartir sesiones de capacitación para RA más jóvenes y para otros miembros del equipo del proyecto. Ella debe tener una buena familiaridad con la ingeniería de sistemas y la vida útil del sistema.

ciclo de vida y una comprensión de las numerosas actividades relacionadas con los requisitos que deben realizarse a lo largo del ciclo de vida del sistema.

La figura 3.1 resume la progresión de la AR.

Otro documento importante es el puesto o descripción del trabajo del RA, que se proporciona en la Tabla 3.2.2.

Este es un resumen conciso y útil del papel de la AR. Describe el puesto, resume las habilidades que se necesitan (aunque no con tanto detalle o precisión como la Figura 3.1), indica los conocimientos que se necesitan, sugiere varias responsabilidades, indica algunas medidas de desempeño y proporciona tres referencias útiles sobre las cuales se basa este trabajo. se basa el artefacto. Le sugiero que utilice este artefacto para aclarar su función y desarrollar solicitudes de puesto para posibles RA. Adáptelo para reflejar sus responsabilidades. Utilícelo en sus evaluaciones de desempeño para discutir actividades de desarrollo profesional que mejorarán sus habilidades con su gerente.

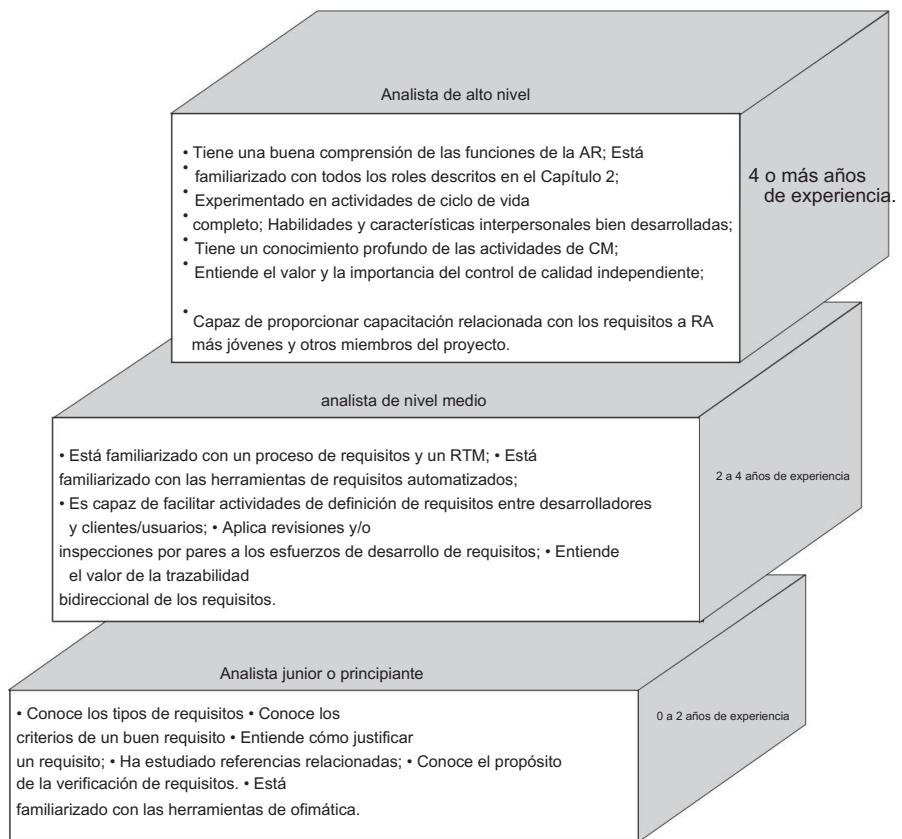


Figura 3.1 El crecimiento profesional de la RA se basa en experiencias acumuladas.
(Adaptado de: Michael Davis.)

Tabla 3.2 Descripción del puesto de RA

Descripción	<p>El RA o ingeniero es la persona que tiene la responsabilidad principal de obtener, analizar, validar, especificar, verificar y gestionar las necesidades reales de las partes interesadas del proyecto, incluidos los clientes y usuarios finales. El RA/ingeniero también se conoce como administrador de requisitos, analista de negocios, analista de sistemas o, simplemente, analista.</p> <p>La RA sirve como conducto entre la comunidad de clientes y el equipo de desarrollo de software a través del cual fluyen los requisitos.</p> <p>Una RA participa en algún nivel durante todo el ciclo de vida del desarrollo del sistema o software. Una vez establecida la línea base de requisitos, la atención se desplaza hacia la gestión de la especificación de requisitos y la verificación del cumplimiento de todos los requisitos.</p> <p>La función de ingeniería de requisitos es una función de proyecto, no necesariamente un puesto de trabajo. La función puede ser desempeñada por un RA dedicado o dividirse entre varios miembros del equipo que tienen otras funciones laborales principales, como PM o desarrollador de productos. La RA es responsable de garantizar que las tareas se realicen correctamente.</p>
Habilidades necesarias	<p>Habilidades de entrevista para hablar con individuos y grupos sobre sus necesidades y hacer las preguntas correctas para obtener información sobre los requisitos esenciales.</p> <p>Habilidades de escucha para comprender lo que dicen las personas y detectar lo que podrían dudar en decir.</p> <p>Habilidades analíticas para evaluar críticamente la información recopilada de múltiples fuentes, conciliar conflictos, descomponer información de alto nivel en detalles, abstraer información de bajo nivel para una comprensión más general, distinguir las solicitudes presentadas de los usuarios de las verdaderas necesidades subyacentes y distinguir ideas de solución. de los requisitos.</p> <p>Habilidades de facilitación para liderar talleres delicitación de requisitos.</p> <p>Habilidades de observación para validar datos obtenidos mediante otras técnicas y exponer nuevas áreas de obtención.</p> <p>Habilidades de redacción para comunicar información de forma eficaz a los clientes, marketing, gerentes y personal técnico.</p> <p>Habilidades organizativas para trabajar con la amplia gama de información recopilada durante la obtención y el análisis y para hacer frente a información que cambia rápidamente.</p> <p>Habilidades interpersonales para ayudar a negociar prioridades y resolver conflictos entre las partes interesadas del proyecto (como clientes, gestión de productos e ingeniería).</p> <p>Habilidades de modelado para representar información de requisitos en formas gráficas que aumentan las representaciones textuales en lenguaje natural, incluido el uso de lenguajes de modelado ya establecidos en la organización de desarrollo.</p>
Conocimiento necesario	<p>Comprensión de los requisitos contemporáneos, obtención, análisis y prácticas de especificación, verificación y gestión y la capacidad de aplicarlas en la práctica.</p>
Responsabilidades	<p>Familiaridad con herramientas de ingeniería de requisitos y otros recursos.</p> <p>Comprensión de cómo practicar la ingeniería de requisitos de acuerdo con varios ciclos de vida de desarrollo de software en un entorno de equipo.</p> <p>Conocimiento de los conceptos de gestión de productos y cómo se posicionan y desarrollan los productos de software empresarial.</p> <p>El conocimiento del dominio de la aplicación es una ventaja para tener credibilidad con los representantes de los usuarios y poder trabajar eficazmente con ellos.</p> <p>Trabaje con el PM, el gerente de producto o el patrocinador del proyecto para documentar la visión y el alcance del producto.</p> <p>Identifique las partes interesadas del proyecto y las clases de usuarios, documente las características de las clases de usuarios e identifique a los representantes adecuados para cada clase de usuarios y negocie sus responsabilidades.</p>

Tabla 3.2 Descripción del puesto de RA (continuación)

	<p>Responsabilidades Obtenga requisitos mediante entrevistas, análisis de documentos, talleres de requisitos, guiones gráficos, encuestas, visitas al sitio, descripciones de procesos comerciales, casos de uso, escenarios, listas de eventos, análisis comerciales, análisis de productos competitivos, análisis de tareas y flujo de trabajo, y puntos de vista.</p> <p>Redactar especificaciones de requisitos según plantillas estándar, utilizando lenguaje natural de forma sencilla, clara, inequívoca y concisa.</p> <p>Descomponer los requisitos comerciales y de usuario de alto nivel en requisitos funcionales y requisitos de calidad, especificados con un nivel apropiado de detalle adecuado para que lo utilicen quienes deben basar su trabajo en los requisitos.</p> <p>Definir atributos de calidad, interfaces externas, restricciones y otros requisitos no funcionales.</p> <p>Representar requisitos utilizando vistas alternativas, como modelos de análisis (diagramas), prototipos o escenarios, cuando corresponda.</p> <p>Liderar el análisis y la verificación de requisitos, asegurando que las declaraciones de requisitos sean completas, consistentes, concisas, comprensibles, rastreables, factibles, inequívocas y verificables y que se ajusten a los estándares.</p> <p>Participar en la priorización de requisitos.</p> <p>Participar en revisiones por pares e inspecciones de documentos de requisitos.</p> <p>Participar en revisiones por pares de productos de trabajo derivados de especificaciones de requisitos para garantizar que los requisitos se interpretaron correctamente.</p> <p>Ingrese, manipule e informe sobre los requisitos almacenados en una herramienta de requisitos comerciales.</p> <p>Defina los atributos de los requisitos y facilite su uso durante todo el proyecto.</p> <p>Administre la información de trazabilidad de los requisitos y realice un seguimiento del estado de los requisitos durante todo el proyecto.</p> <p>Identifique errores y defectos de requisitos y redacte informes de notificación e identificación de defectos de requisitos.</p> <p>Gestione los cambios en los requisitos básicos mediante la aplicación eficaz de procesos y herramientas de control de cambios.</p> <p>Establecer e implementar prácticas de requisitos efectivas, incluido el uso y la mejora continua de un proceso de requisitos.</p> <p>Ayudar con el desarrollo de las políticas, procedimientos y herramientas de ingeniería de requisitos de la organización.</p> <p>Implementar formas de reutilizar los requisitos en todos los proyectos.</p> <p>Identificar formas de ayudar a la gestión de productos en la planificación de productos mediante el desarrollo y análisis de requisitos.</p> <p>Proponer nuevas características y actualizaciones del producto.</p>
Medidas de Actuación	<p>Evaluación de la gestión de productos y proyectos sobre la calidad general del producto y la eficacia en el mercado de los requisitos una vez desarrollado el producto.</p> <p>Comentarios de clientes clave o representantes de marketing sobre la forma en que se llevó a cabo el proceso de ingeniería de requisitos.</p> <p>Medidas de satisfacción del cliente.</p> <p>Satisfacer o superar los cronogramas de desarrollo de requisitos, las limitaciones de recursos y los objetivos de calidad.</p> <p>Control del aumento de requisitos atribuible a requisitos incumplidos y filtración de requisitos "no oficiales" al proyecto.</p>

Tabla 3.2 Descripción del puesto de RA (continuación)

Referencias	Ferdinandi, Patricia L., <i>Un patrón de requisitos: tener éxito en la economía de Internet</i> , Boston: Addison-Wesley, 2002, Capítulo 8. Wiegers, Karl, "Los hábitos de los analistas eficaces", <i>Desarrollo de software</i> 8(10) (Octubre de 2000): 62–65. Young, RR, <i>Prácticas de requisitos eficaces</i> , Boston, MA: Addison-Wesley, 2001, Capítulos 4 y 5.
-------------	---

Notas:

Cada equipo que utiliza esta descripción de trabajo necesita sopesar las diversas habilidades y conocimientos que son pertinentes para su trabajo. Ciertas habilidades enumeradas pueden ser críticas para un trabajo de ingeniero de requisitos y no importantes para otro.

Cada persona que esté considerando contratar a una persona para que sea ingeniero de requisitos debe considerar cuáles de estas habilidades son intrínsecas a la forma en que trabaja el individuo (por ejemplo, habilidades analíticas e interpersonales) y cuáles se pueden aprender (por ejemplo, habilidades de facilitación y escucha).

Los usuarios de esta descripción genérica de trabajo necesitarán modificar parte de la terminología para reflejar sus entornos específicos (por ejemplo, desarrollo de sistemas de información corporativos, desarrollo de productos comerciales, desarrollo de contratos).

Esta descripción del puesto debe adaptarse para que coincida con el nivel de experiencia del puesto.

Fuente: Karl Wiegers et al.

Características de una AR efectiva

Además de las habilidades aprendidas o “duras”, existe un conjunto de características personales que serán de gran utilidad para el RA. Es posible que sienta que algunas de estas características son en sí mismas habilidades. No voy a discutir esto contigo; aceptemos que todas las habilidades y características señaladas son útiles y útiles. La Tabla 3.3 resume las características deseadas que se describen a continuación como contramedidas. puede postularse para superar las barreras que probablemente encontrará.

El Cuadro 3.4 ofrece sugerencias sobre cómo fortalecer estas características.

Considere las siguientes características, que puede optar por seguir perfeccionando, así como las sugerencias y recursos que se ofrecen para ayudarle.

1. Participar en educación continua para adquirir conocimientos expertos en ingeniería de requisitos y prácticas de requisitos. El Capítulo 1 describió los muchos componentes del proceso de requisitos, un conjunto de actividades que se realizan a lo largo del ciclo de vida del sistema de un proyecto. Mi libro anterior [1] proporciona un conjunto completo de referencias en la literatura sobre requisitos a partir de 2001 (en este libro se proporcionan muchas referencias más recientes). Para cada una de las diez prácticas recomendadas descritas en el libro anterior, se proporcionan algunas referencias clave al final de cada capítulo, junto con un breve resumen de la información proporcionada por la referencia. Es útil realizar estudios continuos, como asistir a seminarios de capacitación en áreas relacionadas con la experiencia, la asignación y las actividades. Revistas como IEEE Software, CrossTalk, Software Development Magazine e INSIGHT proporcionan artículos informativos y reseñas de libros relacionados que puede comprar y estudiar. Estos son a la vez informativos y motivadores:

Tabla 3.3 Características deseadas de la AR como contramedidas a posibles barreras

Barreras que puede encontrar	Características como contramedidas
[A] La falta de un conocimiento profundo de la ingeniería de requisitos, los procesos de requisitos y las metodologías de errores de requisitos puede hacer que la RA sea menos efectiva de lo necesario.	[1] Participar en educación continua para adquirir conocimientos expertos en ingeniería de requisitos y prácticas de requisitos. [7] Iniciar el aprendizaje, la aplicación y el uso de prácticas efectivas; buscar patrocinio del PM para actividades relacionadas con los requisitos; estar comprometido con el éxito del proyecto. [13] Mantener un buen conocimiento de la tecnología en evolución y cómo se puede aplicar para satisfacer las necesidades de los clientes.
[B] El producto final no satisface las necesidades del cliente. Hay diferentes ideas y opiniones sobre cuáles son los requisitos reales.	[2] Sea un buen oyente, comunicador y escritor. Documente cuidadosamente las decisiones y los elementos de acción. [3] Tener buenas habilidades de facilitación y negociación. [4] Ser persistentes y perseverantes. [5] Sea proactivo a la hora de involucrar a clientes y usuarios, compañeros de trabajo y gestión de proyectos. [15] Deseo de marcar la diferencia en su labor profesional. [6] Desarrollar la capacidad de comunicarse eficazmente con la dirección.
[C] La gerencia no siempre comprende lo que se está construyendo y qué recursos se necesitan para lograrlo. el producto final.	[10] Desarrollar la capacidad de estimar el tiempo y otros recursos necesarios para realizar el trabajo técnico. [8] Desarrollar y mantener una actitud de mejora continua.
[D] El proceso de requisitos no respalda las necesidades del proyecto.	[14] Establezca metas alcanzables y cumplas. definir y describir métodos para lograr los objetivos del proyecto en un plan de requisitos. [16] Desarrollar su capacidad para contribuir al proceso de riesgo del proyecto.
[E] Las personalidades y opiniones fuertes pueden descarrilar la eficacia de un buen desarrollo y gestión de requisitos.	[2] Sea un buen oyente, comunicador y escritor. Documente cuidadosamente las decisiones y los elementos de acción. [9] Asumir la responsabilidad de sus puntos de vista, actitudes, relaciones y acciones, y mantener el respeto por los demás.
[F] Las personas a menudo intentan hacer más de lo necesario y realizar cambios ad hoc durante el desarrollo del producto de trabajo.	[11] Manténgase enfocado en mantener lo principal como lo principal. Instalar un mecanismo para controlar nuevos requisitos y cambios. No invente requisitos de forma independiente y evite el baño de oro. Evite el aumento de requisitos.
[G] El personal del proyecto puede dedicarse demasiado a la solución del producto de trabajo para analizar y descomponer los requisitos de manera efectiva.	[12] Desarrollar la capacidad de pensar fuera de lo común para proporcionar enfoques creativos que quizás no se les ocurrían a las personas cercanas al problema y al sistema heredado.

Fuente: Richard Rafael.

proporcionar estímulo para fortalecer la propia comprensión. Además, hay varios sitios web que ofrecen reseñas de libros relacionados con requisitos (ver, por ejemplo, el sitio web de Ian Alexander [2]) y "objetos" (artefactos reutilizables relacionados con requisitos) disponibles en el sitio web de Karl Wiegers [3]. Asistir a conferencias como la Conferencia anual sobre Ingeniería de Requisitos del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) [4] o la co-

Cuadro 3.4 Características de una AR efectiva y actividades sugeridas para fortalecerlas

Característica	Actividades sugeridas
1. Participar en la educación continua.	Leer literatura sobre ingeniería de requisitos, asistir a reuniones y conferencias profesionales, visitar sitios web, ocupar cargos en asociaciones profesionales.
2. Buen oyente, comunicador y escritor.	Asistir a seminarios sobre habilidades para escuchar, comunicarse y escribir; Practica hacer presentaciones y escribir.
3. Buena facilitación y habilidades de negociación.	Practique facilitando reuniones, coordinando talleres y gestionando sesiones de diseño de procesos.
4. Persistente y perseverante.	Practique la evolución de los requisitos reales a partir de los requisitos establecidos.
5. Proactivo en involucrar a otros.	Al realizar las tareas diarias, piense deliberadamente en (1) sugerencias para mejorar las cosas y (2) lugares y enfoques apropiados para hacerlo. Práctica. Solicite comentarios y actúe en consecuencia.
6. Capacidad de comunicarse efectivamente con la gerencia.	Practique analizando sus responsabilidades desde la perspectiva de su gerente y la alta dirección. Escriba su perspectiva y la perspectiva de la gerencia. Trabaje para comprender las diferencias y modifique sus comunicaciones en consecuencia.
7. Aprender, aplicar y utilizar prácticas efectivas.	Seleccione una práctica que crea que mejorará una situación laboral. Reúna apoyo para probarlo ("ponerlo a prueba"). Considere los pasos que usted y el proyecto u organización pueden tomar para darle al piloto la mayor probabilidad de éxito. Implementar la práctica. Realice un seguimiento para asegurarse de que sea necesario. Evaluar el valor de implementar la práctica después de un mes.
8. Desarrollar y mantener una actitud de continua mejora.	Practique el ciclo Planificar-Hacer-Verificar-Actuar (PDCA) al finalizar las reuniones. Documento las sugerencias que se ofrecen. Dar seguimiento a las sugerencias en la medida de lo factible y posible. Si esto funciona, explore otras oportunidades para inculcar una actitud de mejora continua, como documentar los procesos y mejorarlo.
9. Asuma la responsabilidad de sus puntos de vista, actitudes, relaciones y acciones.	Dar a conocer a tus compañeros de trabajo algún error que hayas cometido, con el ánimo de contribuir a hacer las cosas mejor. Transmite que sus intenciones eran buenas y que se ha esforzado por aprender del error. Además, trabaje para reconocer el valor aportado por todos los compañeros de trabajo.
10. Desarrollar la capacidad de estimar el trabajo.	Calcule el tiempo que cree que necesitará para realizar las tareas laborales que se le asignen. Realice un seguimiento del tiempo real consumido y observe las distracciones.
requisitos.	Considere los cambios que podría realizar en sus hábitos de trabajo para ser más productivo. Con el tiempo, trate de que las estimaciones se acerquen más a los reales.
11. Mantenga la concentración.	Adopte el concepto de requisitos reales. Comprenda en qué se diferencian de los requisitos establecidos. Sugiera priorizar los requisitos de su proyecto y desarrolle un enfoque para priorizar de forma colaborativa un conjunto de requisitos. Evaluar el impacto de este enfoque.
12. Fortalece tu capacidad de pensar fuera de lo común.	Reunirse con las partes interesadas para considerar posibles soluciones a problemas desconcertantes que no se habían considerado previamente. Utilice la técnica de lluvia de ideas para obtener tres ideas de cada participante. Voto múltiple sobre las ideas sugeridas. Considere el valor potencial de perseguir seriamente una o más ideas.
13. Fortalece tu conocimiento de la tecnología disponible.	Programe una bolsa marrón para considerar las posibilidades tecnológicas. Invite a un arquitecto de sistemas y a otros "tecnólogos". Discuta posibles formas de lograr algunos objetivos del sistema mediante la incorporación de nuevas tecnologías.
14. Establezca metas alcanzables y cumplas.	Planifica tu trabajo para el próximo mes. Establezca algunos objetivos específicos para algunas cosas que crea que son realmente importantes de lograr. Tenga estos objetivos específicos en mente durante el próximo mes. Gestionar los objetivos específicos (es decir, asegurarse de lograrlos).
15. Esfuércese por marcar la diferencia en sus situaciones laborales.	Explore con su gerente cómo las tareas de las cuales es responsable en el trabajo podrían marcar una diferencia para el proyecto u organización. Identifique algunos logros específicos y luego persigalos con seriedad. Solicite el apoyo de sus compañeros de trabajo y de su superior para lograrlos.

Cuadro 3.4 Características de una AR eficaz y actividades sugeridas para fortalecerlas (continuación)

Característica	Actividades sugeridas
16. Contribuya al proceso de riesgo de su proyecto.	Ofrézcase como voluntario para formar parte del equipo de gestión de riesgos del proyecto. Si su proyecto no tiene uno, sugiera que el proyecto considere iniciar un proceso de riesgo. Identifique los principales riesgos, priorícelos y desarrolle planes de mitigación de riesgos para aquellos que se consideren los más graves. Monitorear los riesgos.

Fuente: Richard Rafael.

del Consejo Internacional de Ingeniería de Sistemas (INCOSE) [5] es otra forma de fortalecer el conocimiento, aprender sobre técnicas nuevas y antiguas, conocer a otras personas que están trabajando en esta área y conocer las últimas ofertas disponibles de los proveedores. Considere convertirse en miembro y participante activo en asociaciones y sociedades profesionales como INCOSE, IEEE, la Sociedad Estadounidense para la Calidad (ASQ), la Sociedad para la Calidad del Software (SSQ), la Asociación Internacional de Facilitadores (u organizaciones locales asociadas) y la Grupo de especialistas en ingeniería de requisitos (RESG) en Europa [6].

A menudo, las organizaciones profesionales ofrecen reuniones nocturnas, sesiones a la hora del almuerzo o tutorías los sábados que brindan oportunidades para aprender y conocer colegas. Por ejemplo, el capítulo de INCOSE en Washington, DC ofrece magníficas oportunidades para compartir experiencias, recoger lecciones aprendidas y encontrar fuentes de información, al igual que muchos otros capítulos locales y regionales [7]. Busque activamente oportunidades para escribir artículos y realizar presentaciones.

El viejo dicho de que nadie aprende más que el autor, el maestro o el presentador es cierto.

2. Sea un buen oyente, comunicador y escritor. Las buenas habilidades de comunicación son importantes. Es importante comprender las necesidades y expectativas de las diferentes partes interesadas. Aprenda a escuchar con atención para captar lo que los usuarios y clientes intentan decir, incluso si no son muy buenos expresándolo. Debe poder verificar la comprensión repitiendo su interpretación de las declaraciones. Debe poder escribir de manera clara y concisa para que los requisitos estén documentados de acuerdo con los criterios de un buen requisito proporcionados en el Capítulo 1. El sitio web de Steven Gaffney y sus seminarios han sido recursos valiosos para mí; consulte el trabajo tiendas y materiales disponibles en su sitio web [8].
3. Tener buenas habilidades de facilitación y negociación. Entre las técnicas de recopilación de requisitos más efectivas se encuentra el taller de requisitos. Consulte los Requisitos por colaboración de Ellen Gottesdiener [9] para un tratamiento exhaustivo de esta importante técnica. La AR a menudo puede encontrarse facilitando grupos de personas en dichos lugares. Es importante poder fomentar la identificación de ideas, sin permitir que una o unas pocas personas dominen la discusión. A menudo, necesitará negociar para lograr un consenso.

entre individuos con puntos de vista divergentes. Hay talleres a los que puede asistir para aprender y perfeccionar estas habilidades. Como se señaló anteriormente, existen recursos profesionales disponibles para fortalecer las habilidades de facilitación. Consulte el sitio web de la Asociación Internacional de Facilitadores de ideas [10].

4. Sea persistente y perseverante. Dado que los clientes y usuarios nos brindan sus requisitos declarados, es vital que los RA sean persistentes y perseverantes para que los requisitos reales evolucionen. No basta con depender de poder ofrecer la excusa de que "construimos el sistema que usted solicitó". Si los requisitos indicados no son aceptables, no espere hasta que haya completado el sistema y los usuarios lo rechacen. Reduzca los riesgos de su proyecto mejorando los requisitos lo antes posible. Los riesgos incluyen desperdiciar trabajo técnico y, por supuesto, que el sistema sea rechazado (con todos los riesgos legales y comerciales que ello conlleva). Identificar los requisitos reales es lo más importante que la AR puede hacer para contribuir al máximo a clientes.
5. Sea proactivo a la hora de involucrar a clientes y usuarios, compañeros de trabajo y gestión de proyectos. Pronto comprenderá que no basta con dejarse llevar. El desempeño de las funciones de la AR exige que usted sea proactivo. Los clientes y usuarios necesitan su iniciativa y perseverancia para ayudarlos a desarrollar los requisitos reales. Sus compañeros de trabajo necesitan su apoyo proactivo para ayudarlos a seleccionar y utilizar procesos, prácticas, métodos, técnicas y herramientas eficaces. La gestión de proyectos necesita que usted hable sobre los enfoques que mejor servirán al proyecto, por ejemplo, invertir más en el proceso de requisitos, identificar los requisitos reales y proporcionar un mecanismo para controlar los nuevos requisitos y los cambios en los mismos.
6. Desarrollar la capacidad de comunicarse eficazmente con la dirección. Con demasiada frecuencia, las diferencias de perspectiva impiden una buena comunicación. La gerencia ve la tecnología de la información (TI) como un medio para lograr los objetivos comerciales. Los ingenieros de sistemas y software ven su trabajo en términos de productos de trabajo que deben cumplir requisitos específicos. Como se señaló anteriormente, decir sí no promueve ni su carrera ni el proyecto u organización, cuando los compromisos imposibles sólo garantizan el fracaso en el futuro. Dorothy McKinney ofrece sugerencias en su artículo "Seis traducciones entre el lenguaje del software y el lenguaje de la gestión" [11]. Otra idea es que a veces los AR deben preocuparse no sólo de su propia gestión, sino también de la gestión del cliente.
7. Iniciar el aprendizaje y la aplicación de prácticas efectivas y comprometerse con el éxito del proyecto. Es necesario estar dispuesto a aprender y utilizar prácticas eficaces. El aprendizaje proviene de la experiencia y el estudio. La aplicación de prácticas en proyectos requiere capacitación para familiarizar a las personas con ellas; asesorar a las personas en su uso; seguimiento de su eficacia; y garantizar que su implementación y uso sean efectivos. Un serio

El problema es que, en la mayoría de los casos, a las prácticas nuevas y mejoradas no se les da una oportunidad real porque es parte de la naturaleza humana volver a utilizar prácticas que ya existen.³ Es necesario comprometerse con el éxito del proyecto y con asesorar cuando las cosas sucedan. hay que hacerlo de otra manera o mejor.

8. Desarrollar y mantener una actitud de mejora continua.

Relacionada con el compromiso descrito anteriormente está la idea de manteniendo una actitud de mejora continua. La AR debería alentar al proyecto a adoptar mecanismos para inculcar la actitud de "mejorar cada vez más". En mi empresa finalizamos cada reunión con una evaluación de cómo fue: qué funcionó y qué se podría haber hecho mejor. A esto lo llamamos "hacer PDCA" en honor a las contribuciones del Dr. Deming y Walter Shewhart a la mejora continua de los procesos y nuestra adopción de sus enseñanzas. De manera similar, al final de cada ciclo de actividades, considere realizar un taller para recopilar comentarios sobre cómo fueron las cosas. Utilice las ideas y sugerencias generadas para mejorar la forma en que se realiza el trabajo (es decir, para mejorar el proceso que se está utilizando). Estos mecanismos (además de proporcionar buenas ideas) sirven para ayudar a todos a aceptar los procedimientos utilizados, porque los participantes ayudan a darles forma: "¡Yo ayudé a mejorar ese proceso!"

9. Asuma la responsabilidad de sus puntos de vista, actitudes, relaciones y acciones. Al asumir la responsabilidad, uno establece un sentido de rendición de cuentas.

Tenderás a mostrar orgullo por tu trabajo. No permitirás que las personalidades y las características individuales te impidan tener buenas relaciones con todos. Tus acciones harán una valiosa contribución. Estarás dando ejemplo a los demás. Serás un líder.

10. Desarrollar la capacidad de estimar el tiempo y otros recursos necesarios para realizar el trabajo técnico. Una de las dificultades al hacer estimaciones del trabajo técnico es que estas estimaciones se necesitan temprano para desarrollar proyecciones del número de personal necesario para completar el proyecto. (Para elaborar una estimación del costo del proyecto se requiere la cantidad de personal, su antigüedad y sus funciones.) La dificultad se ve agravada por el hecho de que aún no se conocen las necesidades reales. Por eso, a menudo nos encontramos haciendo estimaciones sin una base precisa para ellas. Esto puede generar mucho trabajo que es

no productivos y también a la confusión causada por la incapacidad de las empresas de cumplir con las estimaciones.

El RA puede contribuir al proceso de estimación (1) trabajando con los usuarios en el entorno del equipo conjunto para identificar el valor real.

3. Consulte "¿Por qué no practican lo que predicamos" de Watts Humphrey para obtener ideas sobre este problema y sugerencias sobre cómo abordarlo? Consulte www.sei.cmu.edu/publications/articles/practice-preach/practice-preach.html.

requisitos, y (2) trabajar con los PM y el personal de desarrollo para hacer estimaciones del tiempo y otros recursos necesarios para realizar el trabajo técnico. Lo mejor es utilizar datos basados en experiencias previas (“gestión basada en hechos”).

El compañero de trabajo John E. Moore en la unidad de negocios Defense Enterprise Solutions (DES) de Northrop Grumman IT es un recurso valioso. Como “propietario del proceso” de gestión de proyectos (planificación de proyectos, seguimiento de proyectos y gestión integrada de productos), el Dr. Moore ha desarrollado una capacidad de “Brickchart” dentro de MS Project que facilita el seguimiento del progreso de las tareas.⁴ Otro compañero de trabajo, Rich Raphael, desarrolló el Asistente del Gerente de Riesgos (RMA), una sencilla herramienta de base de datos que respalda los procesos y programas estándar de gestión de riesgos.⁵ Ambas son herramientas de gestión de proyectos útiles y fáciles de aprender. Véase la discusión en el Capítulo 5 sobre la gestión de riesgos. A medida que una AR adquiere más experiencia en la realización de análisis de requisitos, la AR también debe evaluar cada requisito nuevo o modificado para detectar cualquier riesgo que pueda agregar al proyecto. A medida que los proyectos se vuelven más complejos y los clientes se vuelven más maduros a la hora de especificar sus necesidades, cada requisito nuevo o cambiante corre el riesgo de tener un impacto adverso en el proyecto. Tenga en cuenta que CMMI[®], como parte de la introducción al área de proceso (PA) de gestión de requisitos (REQM), especifica que se debe “consultar el área de proceso de gestión de riesgos (RI) para obtener más información sobre cómo identificar y manejar los riesgos asociados con”.

11. Manténgase enfocado en mantener lo principal como principal. Uno de los obstáculos al desarrollar sistemas y software es que intentamos hacer demasiado; otra es que intentamos incorporar cambios a medida que trabajamos. Los clientes y usuarios preguntarán: ¿Puedes hacer esto? ¿El nuevo sistema hará eso? No nos gusta decir que no. Participamos en la creación de la percepción de que el nuevo sistema será todo para todas las partes interesadas. Al hacerlo, ponemos en peligro nuestra capacidad de cumplir estos compromisos y el éxito del esfuerzo.

La RA puede desempeñar un papel fundamental en este sentido. Desde muy temprano, la AR debería facilitar el establecimiento del concepto de que no todos los requisitos son igualmente importantes y que es responsabilidad de todas las partes interesadas priorizar las necesidades de manera colaborativa y enfocar la intención del proyecto (mantener lo principal como principal). Como sugiere Neal Whitten en “Cumplir con los requisitos mínimos: cualquier cosa más es demasiado” [12], la RA debería trabajar para identificar el conjunto mínimo de requisitos necesarios para lograr los objetivos comerciales. Este objetivo se puede facilitar haciendo lo siguiente:

4. Comuníquese con el Dr. Moore en john.moore@ngc.com.

5. Comuníquese con el Sr. Raphael en RRaphael@ngc.com.

Establecer y seguir un proceso de priorización de todos los requisitos;
Establecer el concepto de lanzamientos o versiones posteriores que abordarán los requisitos de menor prioridad y los requisitos identificados durante el desarrollo de incrementos de funcionalidad más adelante en el proceso de desarrollo;
Garantizar que se establezca y utilice un mecanismo para controlar los nuevos requisitos y los cambios en los mismos.

Estas actividades pueden tener un enorme impacto positivo para mantener el tren en las vías. Una de las principales causas de retrabajo son los cambios introducidos después de que el trabajo técnico se ha completado o está en marcha.

Otro aspecto de esta característica deseada es que la RA no debe inventar requisitos de forma independiente y debe evitar el “dorado”, es decir, agregar características y capacidades a los sistemas y software cuando los requisitos reales no los requieren.

El RA o el desarrollador podrían pensar que saben algo que será “muy interesante” para los usuarios y que podría resultar no deseado o muy perjudicial para el proyecto (por ejemplo, si se incurre en mayores costos para proporcionarlo).

12. Desarrollar la capacidad de pensar fuera de lo común para proporcionar enfoques creativos que quizás no se les ocurran a las personas cercanas al problema y al sistema heredado. Una de las ventajas que aporta un RA a una nueva asignación es que no tiene los mismos intereses creados que tiene un usuario o cliente y, por lo tanto, puede actuar como un agente imparcial o imparcial. La AR llega sin expectativas, sin necesariamente tener mucho conocimiento del ámbito y sin estar apegada a ningún resultado en particular. Sin los obstáculos de años de asociación con un dominio problemático y sin las limitaciones del sistema heredado, usted es libre de pensar más libremente sobre lo que se debe hacer y cómo abordarlo mejor. Aproveche estas oportunidades para pensar en formas nuevas y diferentes de abordar los objetivos del sistema.
13. Mantener un buen conocimiento de la tecnología en evolución y cómo se puede aplicar para satisfacer las necesidades de los clientes. Algunos RA experimentados creen que una sólida formación técnica es muy útil para un RA. Como se mencionó anteriormente, comprender las tecnologías actuales no es responsabilidad exclusiva de la RA, pero podemos contribuir al diseño del sistema involucrando a los arquitectos en las revisiones de los requisitos y ayudándolos en el desarrollo de soluciones técnicas. Otra razón por la que esto es importante es que la incorporación de algunas tecnologías nuevas crea nuevos requisitos que deben tenerse en cuenta. Otros RA experimentados creen que una sólida formación técnica no es tan importante para el RA como otras características, especialmente a la hora de suscitar requisitos y comprender las necesidades y expectativas reales de los clientes y usuarios. Estas personas creen que una técnica fuerte

La perspectiva puede en realidad inhibir la AR y que utilizar a una persona con antecedentes más generales es un mejor enfoque.

14. Establezca metas alcanzables y cúmplalas. Esto está relacionado con la característica de mantener el enfoque. La AR debería fijar objetivos alcanzables y cumplirlos. Tener un plan y proceso de requisitos documentados y seguirlos será útil.

15. Deseo de marcar la diferencia en su trabajo profesional. No deberíamos contentarnos con ir a trabajar o dedicar un número determinado de horas a nuestro trabajo. Más bien, debería ser uno de nuestros valores querer marcar la diferencia en nuestro trabajo profesional. Es vital que se mantenga el control del proyecto. Tener este valor afecta a los demás y nos inspira a ser cada vez más eficaces en nuestros propios roles.

De vez en cuando nos encontramos en una situación en la que no podemos marcar la diferencia. Por ejemplo, participé en un proyecto durante un período de varios meses durante los cuales sentí que había hecho una contribución importante, necesaria y valorada. De repente, el primer ministro pareció retirar su apoyo a mi papel. Hablé de la situación con él y no pude cambiarla. Ya era hora de pasar a un proyecto diferente. A veces necesitamos asumir la responsabilidad del cambio y actuar en consecuencia.

16. Desarrollar su capacidad para contribuir al proceso de riesgo del proyecto. Cada proyecto debe tener un proceso de riesgo para identificar, evaluar, priorizar y mitigar los riesgos existentes o potenciales. Considere participar en el equipo y proceso de gestión de riesgos de su proyecto. Los riesgos relacionados con los requisitos son importantes para el proyecto. Usted puede contribuir al diálogo que ayudará a su proyecto a afrontar sus riesgos con éxito.

Resumen

Las habilidades sugeridas para el RA se enumeran y clasifican en la matriz de habilidades de un RA (Tabla 3.1) según las que necesita un analista de nivel junior, medio o senior. Esta matriz le ayudará a evaluar su idoneidad para un puesto específico en el proyecto. Puede utilizarlo como guía para fortalecer y mejorar aún más sus habilidades o como referencia a fuentes de información sobre cada habilidad. La Tabla 3.2 proporciona una descripción del puesto o puesto de RA que debería ayudarle a aclarar las muchas maneras en que se puede aprovechar el papel de la RA para beneficiar tanto a su proyecto como a su organización. Hacer explícito el papel de la AR ayuda a que un proyecto se ejecute sin problemas. El papel de la AR debe ser comprendido y valorado en la mente de los PM y de la comunidad técnica. nity: ¡esta descripción del trabajo debería ayudar! Se presentaron y describieron dieciséis características de una AR eficaz. Se proporcionan sugerencias sobre cómo fortalecer estas características. Considérelos en el contexto de su propio desarrollo personal y profesional, así como de sus asignaciones y responsabilidades actuales, y seleccione una o algunas características para fortalecer cada año. Sí, ser un RA eficaz implica aprender muchos

habilidades y tener muchas características personales deseadas. Este capítulo, junto con una reflexiva introspección, debería proporcionar una hoja de ruta útil.

Caso de estudio

Se invitó a un consultor de ingeniería de requisitos para ayudar a un particular ubicación de una gran organización del gobierno de EE. UU. Alta dirección en ese lugar indicó, "se habían desperdiciado millones de dólares" en repetidas esfuerzos para desarrollar sistemas y soluciones de software internamente. El consultor se reunió con la alta dirección, gerentes, usuarios y desarrolladores para reunir obtener información y comprender la situación. Siguiente análisis y el desarrollo de un curso de requisitos personalizado, presentó Capacitación para todas las partes interesadas que abordó los problemas existentes en la organización desde la perspectiva de la experiencia relevante de la industria. En la superficie, Parecía haber un deseo sincero por parte de todos los interesados de mejorar la situación, aunque existían muchos problemas. La capacitación abordó cómo estos problemas podrían resolverse. Al finalizar la capacitación, el senior El gerente concluyó que la situación no podía mejorarse. Mucho de Otros participantes en la formación quedaron perplejos ante esta conclusión: sintieron que habían comenzado de nuevo.

Análisis: El propio alto directivo fue la cuestión clave que impidió la situación mejore. Aunque hubo problemas relacionados con todos partes, la administración estaba dispuesta a permitir los intereses provincianos de los usuarios, un proceso de desarrollo excesivamente burocrático y luchas de poder de algunos sectores clave. partes interesadas paralizar los esfuerzos y hacer que la situación mejore. imposible. En el marco del Dr. Deming, había "demasiadas cuentas rojas".⁶ Los usuarios y la organización de desarrollo no tenían poder para mejorar la Situación sin el apoyo y las expectativas de la dirección. para mejores resultados. La dirección debe habilitar y empoderar a sus trabajadores (todos el resto de nosotros) para que el trabajo sea productivo y eficaz. Los estudios de la industria informan que la falta de apoyo adecuado de la alta dirección es un factor en la mayoría de las fallas de TI. Esta viñeta tiene mucho que ofrecer a los altos directivos. La experiencia de la industria es que la alta dirección debe patrocinar y apoyar las iniciativas de desarrollo de TI y sistemas/software si se quiere que sean exitosas. exitoso. Véase la discusión en el Capítulo 8 y un informe reciente de Harvard Business Revise el artículo "Seis decisiones que su personal de TI no debería tomar" [13] para obtener más información y sugerencias específicas. La RA puede ser útil aquí ofreciendo estos conocimientos, sugerencias y experiencia en la industria a su equipo directivo y ayudando a aclarar las funciones específicas que la alta dirección debe desempeñar.

6. Asegúrese de familiarizarse con las enseñanzas del Dr. Deming. Véase, por ejemplo, *The Deming de Mary Walton*.

Método de gestión (Nueva York: The Putnam Publishing Group, 1986). En el Capítulo 4, Walton explica cómo el Dr. Deming utilizó "La parábola de las cuentas rojas" en sus seminarios para recalcar que los trabajadores de cualquier La organización [la mayoría de nosotros] somos impotentes sin el apoyo de la dirección.

Referencias

- [1] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.
- [2] Alexander, IF y Richard Stevens, Redactando mejores requisitos, Boston: Addison-Wesley, 2002, www.easyweb.easy.net.co.uk/~iany/index.htm.
- [3] Wiegers, K., sitio web, en www.processimpact.com (para información relacionada con los requisitos). “obsequios” y otra información útil).
- [4] Conferencia de ingeniería de requisitos IEEE .org/RE, Sitio web, en conferencias.computadora
- [5] Sitio web de la organización nacional INCOSE, en www.incose.org/se-int.
- [6] Grupo de especialistas en ingeniería de requisitos (en el Reino Unido), sitio web, en www.resg.org.uk.
- [7] Capítulo INCOSE WMA, sitio web, en www.incose-wma.org/info.
- [8] Gaffney, S., sitio web, en www.StevenGaffney.com.
- [9] Gottesdiener, E., Requisitos por colaboración: talleres para definir necesidades. Lectura, MA: Addison-Wesley, 2002.
- [10] Asociación Internacional de Facilitadores, sitio web, en www.iaf-world.org.
- [11] McKinney, D., “Seis traducciones entre el lenguaje del software y el lenguaje de la gestión”, IEEE Software 19(6) (2002): 50–52. Consulte www.computer.org/software.
- [12] Whitten, N., “Cumplir con los requisitos mínimos: cualquier cosa más es demasiado”, PM Network (septiembre de 1998), p. 19.
- [13] Ross, JW y P. Weill, "Seis decisiones de TI que su personal de TI no debería tomar", Harvard Business Review (noviembre de 2002): 85–91.

CAPÍTULO

4

Contenido

[Vistas de tipos de requisitos](#)[Definiciones y descripciones de Tipos de requisitos](#)[Terminologías a evitar](#)[Ejemplos de requisitos](#)[Tipos](#)[Resumen](#)[Caso de estudio](#)[Referencias](#)

Tipos de requisitos

Es importante que el RA o el ingeniero de requisitos establezca definiciones de los tipos de requisitos que utilizará de manera consistente. Debería defender significados consistentes para estos tipos en su proyecto y en su organización. Se puede evitar mucha confusión acordando un conjunto de definiciones y no utilizando ciertos términos. En este capítulo, revisaremos varios tipos de requisitos y sugeriremos definiciones para ellos. Sugeriremos por qué no se deben utilizar algunos términos y brindaremos otras pautas. Una razón importante para llegar a un acuerdo sobre las definiciones de los tipos de requisitos es evitar debates largos y acalorados sobre la terminología mientras trabajamos juntos.

Establezca un glosario de proyectos con el que todos puedan vivir (incluso si algunas definiciones no son las favoritas de todos) y utilícelo en su trabajo. Considere el glosario proporcionado con este libro como punto de partida y adáptelo según sea necesario.

Primero, recordemos nuestra definición simple y útil de requisito del Capítulo 1. Un requisito es una declaración que identifica una capacidad, característica o factor de calidad de un sistema para que tenga valor y utilidad para un usuario. Un requisito está bien definido y es más específico que una necesidad, que es una capacidad deseada por un usuario o cliente para resolver un problema o lograr un objetivo. Los autores del Modelo de Madurez de Capacidades de Ingeniería de Sistemas (SE-CMM) [1] fueron perspicaces cuando crearon el Área de Proceso 06, “Comprender las necesidades y expectativas del cliente”. El propósito de esta área de proceso es obtener, estimular, analizar y comunicar las necesidades y expectativas de los clientes y usuarios y traducirlas en un conjunto verificable de requisitos.

Vistas de tipos de requisitos A continuación, proporcionaremos tres formas diferentes de organizar los tipos de requisitos. Estas vistas le ayudarán a poner su trabajo en contexto y

perspectiva. Las opiniones son cortesía de Jeffrey O. Grady, autor e instructor de cursos de requisitos e ingeniería de sistemas. Las definiciones de los distintos tipos se proporcionan más adelante en este capítulo.

La tabla 4.1 proporciona la primera vista. Aquí los tipos de requisitos se dividen en hardware y software. Luego, los requisitos de hardware se caracterizan como requisitos de rendimiento o restricciones. Los requisitos de desempeño definen qué tan bien el sistema debe cumplir un requisito. Las restricciones se caracterizan además como requisitos de interfaz, requisitos de ingeniería especializada y requisitos ambientales. Los requisitos de software se caracterizan como funcionales o no funcionales. Los requisitos funcionales especifican una acción que un sistema debe poder realizar. Un requisito no funcional especifica propiedades del sistema, como confiabilidad y seguridad (consulte la discusión sobre "requisitos de ingeniería especializada y de capacidades" a continuación).

La Figura 4.1 proporciona un contexto más detallado para la AR. Los tipos de requisitos que se anotan son los requisitos del proceso de producción (por ejemplo, las instalaciones físicas necesarias), los requisitos de los productos que proporcionará el sistema o software, los requisitos de los procesos utilizados para producir los productos (por ejemplo, el proceso de prueba) y los requisitos de apoyo operativo y logístico (por ejemplo, equipo, capacitación y procedimientos). Todos estos requisitos deben identificarse antes de comenzar a trabajar en el diseño detallado del sistema. Mientras los ingenieros de producto desarrollan especificaciones para los elementos del producto, los ingenieros de fabricación deben definir los requisitos de fabricación, los ingenieros de logística los requisitos de logística y los ingenieros de verificación los requisitos de calificación. Al hacerlo, estos ingenieros deben comunicarse entre ellos y resolver conjuntamente la mejor expresión agregada de los requisitos desde la perspectiva del producto y del proceso.

Es importante señalar que se deben realizar varios pasos o recorridos de los requisitos reales identificados¹ para garantizar, por ejemplo, lo siguiente:

Tabla 4.1 Tipos de requisitos

Requisitos de hardware:
Requisitos de desempeño
Restricciones:
Requisitos de interfaz
Requisitos de ingeniería especializada
Requisitos medioambientales
Requisitos de Software:
Requerimientos funcionales
Requerimientos no funcionales

Fuente: Jeffrey O. Grady. Usado con permiso.

1. La asesora y practicante de la industria Ellen Gottesdiener, presidenta de EBG Consulting, Inc., recomienda tres o cuatro iteraciones del desarrollo de requisitos, cada una de las cuales incorpora una revisión formal o informal por parte de clientes internos y externos. Su experiencia enfatiza el valor y la importancia de identificar los requisitos reales antes de iniciar otros trabajos.

Vistas de tipos de requisitos

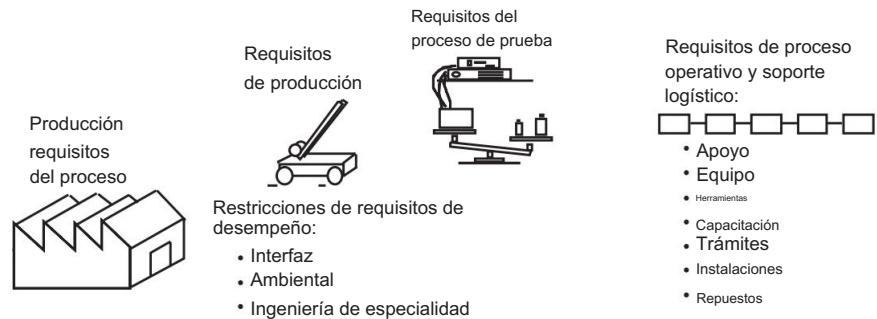


Figura 4.1 Otra vista de los tipos de requisitos.

Los requisitos son mutuamente consistentes;

Se priorizan los requisitos (nunca hay suficiente tiempo y dinero para hacer todo).

La Figura 4.2 proporciona una taxonomía de requisitos totales. Grady describe su figura de la siguiente manera:

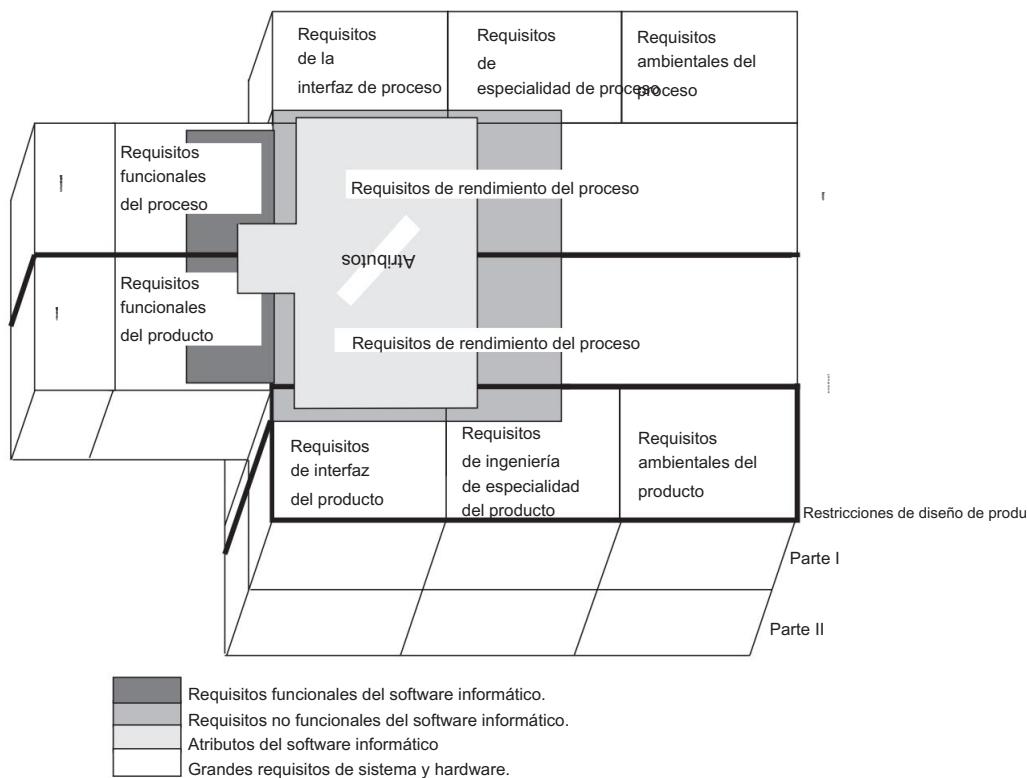


Figura 4.2 Taxonomía de requisitos totales. (Fuente: Jeffrey O. Grady.)

La capa superior corresponde a los requisitos de desarrollo, a menudo llamados requisitos de diseño que deben entenderse claramente antes del diseño. La capa inferior corresponde a los requisitos del producto, comúnmente llamados requisitos de fabricación . Los requisitos por encima de la línea media gruesa corresponden a los requisitos del proceso capturados en las declaraciones de trabajo y planes. Los requisitos del proceso se incluyen en los planes y procedimientos del programa. Los requisitos del producto se capturan en las especificaciones del programa comenzando con el requisito final, la necesidad del cliente. Necesitamos métodos efectivos para ampliar la declaración de necesidades a una visión más refinada de las necesidades del cliente, definiendo así requisitos de desempeño más detallados. Además, debemos identificar requisitos adicionales llamados restricciones de tres tipos que requieren modelos especiales. Es posible que las especificaciones que creamos deban publicarse en dos partes: (1) especificaciones de desarrollo o rendimiento, y (2) especificaciones de producto o detalles. El primero impulsa el diseño y la cualificación. Este último tipo impulsa la aceptación [8].

Otra vista con la que la RA debe estar familiarizada se proporciona en el Estándar 632 [2] de la Asociación de Industrias Electrónicas (EIA), Sección 4, (“Requisitos”), y en el Estándar IEEE 12207 [3], Sección 5.3.2 (“Sistema- Análisis de requisitos del sistema”) y Sección 5.3.4 (“Análisis de requisitos de software”). Debe asimilar estos estándares y considerar las categorías que se sugieren.

Otro enfoque es el Marco Zachman (ZF) [4, 5]. John Hay describe el ZF en su libro Análisis de requisitos: desde las vistas empresariales hasta la arquitectura [6]. Hay describe el trabajo del RA como un movimiento de las filas uno y dos a la fila tres en el ZF, y el libro está dedicado a las técnicas analíticas utilizadas en cada columna. Una revisión puede ayudar a aclarar y comprender los distintos tipos de requisitos.

También son compatibles con ZF las categorías CMMI [7] de requisitos de cliente, producto y componentes del producto. Tanto CMMI como ZF tratan el análisis de requisitos como una progresión continua desde las necesidades y expectativas del cliente hasta las especificaciones del sistema. Esta progresión es útil para comprender qué hace una AR.

Definiciones y descripciones de tipos de requisitos

El resto de este capítulo proporciona definiciones y descripciones de muchos tipos de requisitos. Reitero mi sugerencia anterior de que establezca un glosario de términos que utilizará en su proyecto. Trabajando con los otros desarrolladores, agregue palabras y definiciones que todos puedan respaldar a medida que avanzan con su trabajo. No pierda mucho tiempo llegando a un consenso sobre las definiciones: simplemente utilice la técnica en las reuniones del proyecto en la que cada persona indique su acuerdo con el pulgar hacia arriba, hacia abajo o hacia un lado para llegar a acuerdos con los que la gente pueda vivir. He sugerido que algunos de estos términos (tipos de requisitos) que se proporcionan a continuación no deberían usarse porque tienden a confundir a las personas y crear malentendidos.

poner en peligro la finalización exitosa de las actividades laborales. Estas son sólo mis propias opiniones y prejuicios basados en mi experiencia. Es posible que tengas opiniones diferentes y ¡está bien!

La Tabla 4.2 proporciona la visión de una RA de muchos de los tipos de requisitos. Puede ayudarle a obtener y aplicar una comprensión útil de los diferentes tipos de requisitos.

Requisitos comerciales Los

requisitos comerciales son la razón principal para desarrollar sistemas y software. Los requisitos comerciales son las actividades esenciales de una empresa. Los requisitos comerciales se derivan de las metas comerciales (los objetivos de la empresa u organización). Los escenarios comerciales se pueden utilizar como técnica para comprender los requisitos comerciales. Un factor clave en el éxito de un sistema es el grado en que el sistema respalda los requisitos del negocio y facilita que una organización los alcance. Si nuestros sistemas y software no soportan los requisitos del negocio de manera efectiva y eficiente, no tienen razón de ser. Las empresas existen para ganar dinero.

Tabla 4.2 Visión de una RA sobre los tipos de requisitos

Necesidades y expectativas del cliente:
Requisitos comerciales;
Requisitos de usuario;
Requisitos del producto;
Requisitos medioambientales;
Requisitos incognoscibles.
Estos son analizados por el analista de requisitos y descritos de diferentes maneras:
Requisitos de alto nivel (o nivel de sistema);
Requisitos funcionales (qué debe hacer el sistema);
Requerimientos no funcionales:
Propiedades del sistema (por ejemplo, seguridad);
Los "requisitos de ingeniería especializada/ilidades".
Requisitos derivados y restricciones de diseño;
Requisitos de desempeño (por ejemplo, ¿qué tan rápido?);
Requisitos de interfaz (relaciones entre elementos del sistema);
Los requisitos del sistema se distribuyen en:
Subsistemas (agrupaciones lógicas de funciones);
Componentes del sistema (hardware, software, formación, documentación).
Se realizan comprobaciones para garantizar que el sistema haga lo que se supone que debe hacer, incorporando:
Requisitos verificados;
Requisitos validados;
Requisitos de calificación.

para los accionistas; Las organizaciones existen para satisfacer las necesidades de sus miembros. Es vital que consideremos nuestro trabajo de desarrollo de software y sistemas totalmente dentro del contexto de los objetivos comerciales y organizacionales.

Requisitos declarados versus requisitos reales

Ya hemos aclarado la diferencia entre estos:

Los requisitos establecidos son proporcionados por el cliente al comienzo de un esfuerzo de desarrollo de sistemas o software.

Los requisitos reales reflejan las necesidades verificadas de un sistema o capacidad en particular. Tenga en cuenta que se pueden identificar algunos requisitos reales que el cliente y los usuarios omitieron en los requisitos establecidos. De hecho, identificar los requisitos omitidos es una tarea clave de la AR.

Requisitos de usuario

Los usuarios son los individuos o grupos que utilizan un sistema o software en su entorno. Los requisitos del usuario son sus necesidades verificadas para ese sistema o software.

Requisitos de alto nivel o de sistema

Para permitir comprender un sistema necesario, nos referimos a los requisitos de alto nivel o nivel de sistema. Este término se relaciona con aquellos requisitos que son de mayor importancia, capturan la visión del cliente, permiten definir el alcance del sistema y permiten estimar el costo y el cronograma requerido para construir el sistema. (Algunos arquitectos de sistemas creen que la especificación de requisitos debe contener todos los requisitos de rendimiento).

Se recomienda identificar una cantidad viable de requisitos (del orden de 50 a 200) a nivel de sistema para un sistema grande. En el Capítulo 8, analizaremos un conjunto de impulsores comerciales que pueden considerarse requisitos de alto nivel del cliente, que a menudo no se expresan.

Reglas del negocio

Las reglas de negocio² proporcionan la base para crear los requisitos funcionales.

Son los siguientes:

Las políticas, condiciones y limitaciones de las actividades comerciales respaldadas por el sistema;

Los procesos de decisión, directrices y controles detrás de los requisitos funcionales (por ejemplo, procedimientos);

2. Esta discusión se resume a partir de materiales desarrollados por Ellen Gottesdiener, que incluyen "Captura de reglas comerciales", "El valor de la estandarización de las reglas comerciales" y "Conversión de reglas en requisitos". Estos materiales están disponibles en su sitio web, www.ebgconsulting.com.

Definiciones utilizadas por la empresa;

Relaciones y flujos de trabajo en el negocio;

Conocimientos necesarios para realizar acciones.

Una pauta a seguir es documentar las reglas comerciales de manera correcta y temprana. Para lograr esto se requiere el patrocinio activo y el liderazgo de su cliente, porque probablemente habrá políticas y reglas comerciales “indiscutibles”, poco claras y contradictorias, que deben aclararse y resolverse para proporcionar una base adecuada para el desarrollo de los requisitos reales del sistema o software. Es importante identificar las reglas de negocio que son inconsistentes, en conflicto, inefficientes, redundantes, no estandarizadas, que no cumplen con las regulaciones o la política de la empresa, o que no tienen dueño.

Las reglas de negocio deben ser capturadas explícitamente por la RA y basarse en ellas durante el análisis de requisitos. Centrarse en las reglas de negocio como requisitos funcionales centrales acelera el análisis de requisitos y promueve la validación y verificación. La AR debe utilizar el proceso de “Documentación de reglas comerciales” descrito en la Figura 4.3 para seleccionar o adaptar una taxonomía y una plantilla de reglas comerciales para cualquier problema comercial determinado. La plantilla proporciona una sintaxis estándar para escribir reglas comerciales en lenguaje natural (inglés).

Si se encuentra en una situación en la que necesita ayuda, considere comunicarse con Ellen Gottesdiener (ellen@ebgconsulting.com) para facilitar un taller sobre requisitos de reglas comerciales.

Requisitos funcionales Los

requisitos funcionales son una categoría importante de los requisitos reales.

Los requisitos funcionales describen lo que debe hacer el sistema o software. Una función es una capacidad útil proporcionada por uno o más componentes de un sistema. Los requisitos funcionales a veces se denominan requisitos conductuales u operacionales porque especifican las entradas (estímulos) del sistema, las salidas (respuestas) del sistema y las relaciones conductuales entre ellos. El documento utilizado para comunicar los requisitos a los clientes, a los ingenieros de sistemas y de software se denomina documento funcional.

La RA colabora con:

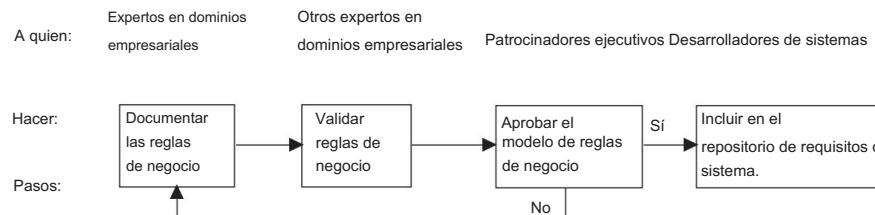


Figura 4.3 Documentación del proceso de reglas comerciales. (Adaptado de: Ellen Gottesdiener.)

documento (FD) o especificación. Esto se refiere a una colección completa de las características de un sistema y las capacidades que pondrá a disposición de los usuarios. Proporciona un análisis detallado de los datos que se espera que manipule el sistema. Puede incluir una definición detallada de las interfaces de usuario del sistema.

Requisitos no funcionales Los requisitos

no funcionales especifican propiedades del sistema, como confiabilidad y seguridad.

Requisitos derivados Un

requisito derivado es aquel que se refina aún más a partir de un requisito de nivel superior o un requisito que resulta de la elección de una implementación o elemento del sistema específico. En cierto sentido, todos los requisitos se derivan de la necesidad del sistema; por tanto, la distinción derivada tiende a tener poca significación. Sin embargo, muchos ingenieros de sistemas distinguen entre requisitos identificados externamente y requisitos que se derivan bajo el control del ingeniero.

Requisitos y restricciones de diseño Para la mayoría de los

esfuerzos de desarrollo de sistemas, los requisitos/restricciones de diseño aparecen justo al comienzo de la formulación del sistema. A continuación se muestran ejemplos de por qué es difícil separar la ingeniería de requisitos de las actividades de diseño:

A menudo se instalan nuevos sistemas en entornos que ya cuentan con otros sistemas. Los otros sistemas suelen limitar el diseño del nuevo sistema. Por ejemplo, un requisito (restricción de diseño) puede ser que el sistema a desarrollar deba obtener su información de una base de datos existente. La base de datos ya ha sido diseñada y partes de su especificación normalmente se incluirán en el documento de requisitos.

Para sistemas grandes, a menudo es necesario algún diseño arquitectónico para identificar subsistemas y relaciones. La identificación de subsistemas significa que el proceso de ingeniería de requisitos para cada subsistema puede continuar en paralelo.

Por razones de presupuesto, cronograma o calidad, una organización puede desear reutilizar algunos o todos los sistemas de software existentes en la implementación de un nuevo sistema. Esto limita tanto los requisitos del sistema como el diseño.

Si un sistema tiene que ser aprobado por un regulador externo (por ejemplo, sistemas en aeronaves civiles), puede ser necesario utilizar un diseño certificado estándar que haya sido probado en otros sistemas.

Requisitos de rendimiento Uno de los

desafíos más difíciles en el desarrollo de sistemas es definir y cumplir los requisitos de rendimiento (a veces denominados requisitos de confiabilidad). Los requisitos de desempeño definen qué tan bien deben funcionar los requisitos funcionales. Los componentes del análisis de requisitos de desempeño están más allá del alcance de este libro, pero están descritos por Jeffrey O. Grady en Systems Requests Analysis [8, págs. 238, 313 y 324]. Grady también proporciona un conjunto de pautas útiles para la identificación de requisitos de desempeño [8, págs. 323–324]. Los requisitos de confiabilidad corresponden a las necesidades a nivel del sistema en materia de disponibilidad, seguridad, rendimiento, confiabilidad y protección.

Requisitos de interfaz Otro desafío

difícil en el desarrollo de sistemas es encontrar y definir los requisitos de interfaz. El análisis de requisitos de interfaz identifica relaciones físicas y funcionales entre los elementos del sistema y entre los elementos del sistema y el entorno del sistema. A un miembro del equipo del proyecto se le debe asignar la responsabilidad principal de asegurar la coordinación de los requisitos de la interfaz. Véase [8, págs. 270–297] para obtener una buena discusión sobre técnicas y análisis de interfaces.

Requisitos verificados Los

requisitos verificados son requisitos reales que se cumplen o satisfacen en la solución de diseño.

Requisitos validados Los requisitos

validados son requisitos que se implementan en el sistema entregado. Consulte Validación y verificación del sistema de Jeffrey O. Grady [9] para obtener definiciones claras de estos términos, información detallada sobre cómo utilizar estas herramientas fundamentales de resolución de problemas y métodos prácticos para cada paso del proceso.

Requisitos de calificación La calificación

se refiere a la verificación o validación del desempeño del elemento en una aplicación específica y a los resultados de la revisión del diseño, la revisión de los datos de prueba y las auditorías de configuración.

Las “Ilidades” y los requisitos de ingeniería especializada

A menudo se escuchan referencias a las “Ilidades” de un sistema, a veces llamadas atributos de calidad, como las siguientes:

Designabilidad;

Eficiencia;

Ingeniería humana;

Modificabilidad;

Portabilidad;

Fiabilidad;

Comprobabilidad;

Comprensibilidad;

Capacidad;

Degrado del servicio;

Mantenibilidad;

Memoria;

limitaciones de tiempo;

Modificabilidad;

Usabilidad.

Estos son los requisitos no funcionales o no conductuales de un sistema o del software. Consulte Requisitos de software: objetos, funciones y estados de Alan Davis [10, págs. 307–340] para obtener una discusión detallada y técnicas sugeridas.

Requisitos incognoscibles La

experiencia ha demostrado que hay requisitos que son incognoscibles al comienzo de un esfuerzo de desarrollo de sistemas. Algunos requisitos se vuelven evidentes sólo a medida que el sistema evoluciona. Descubrimos que tenemos un requisito que antes no podíamos imaginar.

Estos requisitos incognoscibles pueden ser requisitos reales, que deben incluirse.

Requisitos del producto Estos

son requisitos de los productos que produce un sistema.

Requisitos del proceso Hay

requisitos que existen debido a los procesos que se utilizan para desarrollar el sistema o software.

Requisitos de soporte logístico Estos son

requisitos que existen debido a cosas como herramientas, capacitación, procedimientos, instalaciones y repuestos. A menudo se oye hablar de ellos en referencia a los requisitos del soporte logístico integrado (ILS).

Requisitos medioambientales

Son requisitos que resultan del entorno físico y social y condiciones culturales del esfuerzo de desarrollo del sistema y el entorno en el cual se utilizará el sistema o software.

Requisitos del sistema, subsistema y componentes

Esto se refiere a los requisitos asociados con diferentes niveles del sistema. El sistema es el nivel más alto y se divide en subsistemas; Los subsistemas se componen de componentes, como hardware, software, capacitación y documentación.

Terminologías a evitar

Requisitos de fuente o cliente

A veces se oye a la gente referirse a los requisitos de origen o a los requisitos del cliente. En cambio, prefiero especificar la fuente del requisito (es decir, de quién o dónde se identificó el requisito real) como un atributo³ de un requisito real. Tener identificada la fuente para cada requerimiento real nos permite acudir a la persona o documento para dudas y aclaraciones. Sugiero evitar el uso de los términos requisitos de origen o requisitos del cliente.

Requisitos no negociables versus requisitos negociables

Un requisito no negociable implica que si no se cumple, el sistema es de poca utilidad. Es evidente que el requisito es un requisito real. Un requisito negociable implica que está realmente bien si no se satisface en el sistema entregado. Es evidente que los requisitos negociables no son requisitos reales. Esta clasificación no es útil. El artículo de Neal Whitten, "Cumplir con los requisitos mínimos: Cualquier cosa más es demasiado" [11] es instructivo. Cumplimiento mínimo real requisitos es lo mejor para todos, porque este enfoque reduce riesgo, costo, cronograma, complejidad, etc. Ten presente el CAOS Informe [12] conclusión de que el 45% de las funciones del sistema proporcionadas en los países desarrollados ¡Los sistemas nunca se usan una vez!

3. Véase Prácticas efectivas de requisitos de Young [13, págs. 85–87] para una discusión de los atributos de un requisito y una matriz de atributos de requisitos de muestra que se podría usar en una herramienta de requisitos como Dynamic Sistema de requisitos orientado a objetos (DOORS). Ejemplos de atributos de cada requisito incluyen únicos ID, fuente, propietario, justificación (por qué se necesita el requisito), prioridad, estado (aprobado, pendiente de aprobación, rechazado, en reconsideración), costo, dificultad, estabilidad, asignado a, ubicación, autor, revisión, fecha, motivo, rastreado desde, rastreado hasta, número de etiqueta raíz, historial, verificación, validación, lanzamiento, módulo y otros que Dependerá de las necesidades específicas de su proyecto. Consulte también el paso 19 en el Capítulo 5.

Requisitos clave El

término requisitos clave se utiliza a veces para referirse a requisitos que son importantes para comprender las capacidades o funciones esenciales de un sistema.⁴ Es apropiado analizar los requisitos en términos de su relación beneficio-costo, riesgo o el tiempo y el esfuerzo estimados necesarios para abordarlos, de modo que podamos tener discusiones informales dentro del equipo conjunto para negociar los requisitos que se incluirán. Sin embargo, sugiero evitar el uso de este término porque no está claro.

Requisitos originarios En El diseño

de ingeniería de sistemas: modelos y métodos [14], Dennis Buede utiliza el término requisitos originarios para referirse a los requisitos establecidos inicialmente por las partes interesadas del sistema con la ayuda del equipo de ingeniería de sistemas. El término no es tan claro como el término requisitos reales y, por lo tanto, sugiero no utilizarlo.

Otras pautas

Evite el uso de terminología vaga, como “generalmente, a menudo, típicamente, generalmente, fácil de usar, versátil, flexible, confiable y actualizable”, al escribir los requisitos.

Evite incluir más de un requisito en un requisito (a menudo indicado por la presencia de la palabra “y”).

Evite cláusulas como “si eso fuera necesario”.

Evite las ilusiones: 100% de confiabilidad, ejecutándose en todas las plataformas, complaciendo a todos los usuarios, manejando todas las fallas inesperadas.

Consulte Writing Better Requests de Alexander y Stevens [15] para obtener excelentes pautas basadas en una amplia experiencia.

Ejemplos de tipos de requisitos

El siguiente escenario proporciona ejemplos de los tipos de requisitos discutidos anteriormente que facilitarán la comprensión de los diferentes tipos.

ABC, Inc. ha experimentado un crecimiento fenomenal en los últimos tres años debido a fusiones y adquisiciones de empresas de naturaleza similar o complementaria. Para mejorar la posición competitiva de ABC, la dirección

4. El experto en la industria Ian Alexander advierte que es práctica del Ministerio de Defensa (MoD) del Reino Unido utilizar muy pocos “requisitos clave”; podría haber cinco para un buque de guerra, por ejemplo. “Éstos se convierten en los objetivos más buscados por el sistema: si todo lo demás falla, estos objetivos permanecen y los demás requisitos pueden evaluarse a su luz. Puede que no sea adecuado para todos, pero contiene algo de la esencia de la priorización”.

Comunicación personal al autor, 18 de enero de 2003.

desea un sistema de recursos humanos para identificar a los empleados con las habilidades y la capacitación necesarias en todas las ubicaciones de la empresa.

La gerencia de ABC ha determinado que desarrollar un nuevo sistema integral de recursos humanos en toda la empresa tendría un costo prohibitivo. En cambio, la dirección desea dejar los sistemas heredados de cada empresa adquirida e integrar los datos contenidos en ellos en un marco de TI. El marco consistirá en una infraestructura de TI en la sede, la red telefónica pública y equipos de comunicaciones en cada ubicación de la empresa necesarios para respaldar los procesos del marco.

El marco, cuyo nombre en código SATURN (Red Unificada de Referencia de Habilidades y Capacitación), no reemplazará ninguno de los sistemas heredados de las empresas individuales que ABC ha adquirido. La presentación de los nuevos datos de los empleados de toda la empresa será transparente para el usuario local. En otras palabras, cada usuario local verá la información de los empleados seleccionados para toda la empresa en el mismo formato que si se almacenara localmente en su propio sistema heredado.

Además, SATURN podrá determinar si todos los sistemas heredados están disponibles en el momento en que se realiza una consulta. Si un sistema heredado en particular no está disponible por algún motivo, se notificará al usuario para que pueda realizar una consulta más adelante. Cada consulta debe estar completa y toda la información disponible debe devolverse al usuario en un minuto. el saturno

El sistema debería poder admitir hasta 20 usuarios simultáneos sin ninguna degradación del rendimiento.

La siguiente Tabla 4.3(a–d) proporciona ejemplos de tipos de requisitos para el desarrollo del sistema SATURN. La Tabla 4.3(a) proporciona ejemplos de requisitos para la vista de requisitos de hardware/software que se muestra en la Tabla 4.1. La Tabla 4.3(b) proporciona ejemplos del contexto más detallado descrito en la Figura 4.1. La Tabla 4.3(c) proporciona ejemplos de la vista de taxonomía de requisitos totales descrita en la Figura 4.2. Finalmente, la Tabla 4.3(d) proporciona ejemplos de la visión RA de los requisitos descritos en la Tabla 4.2.

Resumen

De esta discusión se desprende que existen muchos tipos diferentes de requisitos. Es útil aceptar utilizar unos pocos tipos seleccionados. Acuerde dentro de su equipo de proyecto los tipos que serán más útiles. Utilice el glosario de su proyecto, que proporciona terminología definida y acordada. Utilice palabras sencillas y comprensibles. Escriba requisitos que cumplan con los criterios de un buen requisito (consulte el Capítulo 1). Estudie las referencias proporcionadas para este capítulo si aún no está familiarizado con ellas.

Caso de estudio

Teníamos un requisito conocido para el rendimiento del sitio web. Implicaba un conjunto finito de escenarios de usuario que debían ser ejecutados por un número específico de

Tabla 4.3(a) Requisitos de hardware/software Ver ejemplos

Requisitos de hardware	
Actuación	El sistema SATURN completará todas las recuperaciones y mostrará la información solicitada, dentro de un minuto de que el usuario ingrese la consulta. Hasta 20 usuarios simultáneos pueden utilizar el sistema SATURN sin ninguna degradación del tiempo de respuesta.
Requisito de interfaz	El sistema SATURN funcionará a través de un navegador disponible comercialmente como Internet Explorer o Netscape.
Requisito de ingeniería especializada	El sistema SATURN se ejecutará en hardware comercial disponible (COTS) utilizando el sistema operativo Microsoft Windows.
Requisito ambiental	El sistema SATURN funcionará con energía monofásica disponible comercialmente con un voltaje de línea en el rango de 110 voltios, más o menos 20 voltios CA.
Requisitos de Software	
Requerimiento funcional	El sistema SATURN recuperará información de identificación básica de todos los empleados que cumplan con los criterios especificados.
Requisito no funcional	El sistema SATURN generará mensajes de error cuando una consulta no se ejecute hasta su finalización o un sistema heredado no responda dentro del tiempo asignado.

Fuente: Terry Bartolomé. Usado con permiso.

Tabla 4.3(b) Ejemplos de requisitos de contexto más detallados

Requisito del proceso de producción	El sistema SATURN estará disponible para su uso por todos los representantes de Recursos Humanos en cada instalación de la empresa.
Requisito del producto	El sistema SATURN recuperará información de identificación básica de todos los empleados que cumplan con los criterios de capacitación y habilidades predeterminados.
Requisito del proceso de prueba	Los registros de recursos humanos de prueba para verificar el sistema SATURN consistirán en un conjunto especial de registros de personal en cada ubicación de la empresa creados específicamente con datos artificiales.
Requisito del proceso operativo	El sistema SATURN tendrá la misma apariencia en cada ubicación de la empresa con la que los usuarios del sistema en esa ubicación están familiarizados y, por lo tanto, no requerirá capacitación.

Fuente: Terry Bartolomé. Usado con permiso.

Tabla 4.3(c) Ejemplos de requisitos de vista de taxonomía total

Requisito funcional del proceso	El sistema SATURN se desarrollará para proporcionar acceso en toda la empresa a las habilidades de los empleados y a la información sobre capacitación a todos los representantes de Recursos Humanos.
Requisito de interfaz de proceso	La información sobre habilidades y capacitación de todas las ubicaciones de la empresa estará disponible para todas las demás ubicaciones de la empresa.
Requisito de especialidad de proceso	Para garantizar que se capture información completa sobre habilidades y capacitación entre los sistemas heredados, se debe crear un modelo de datos.
Requisito ambiental del proceso	SATURN se desarrollará utilizando equipos de desarrollo conjunto de aplicaciones (JAD) compuestos por usuarios (representantes de recursos humanos), desarrolladores y probadores de sistemas.
Requisito de desempeño del proceso	El sistema SATURN estará listo para las pruebas de aceptación del sistema dentro de los 180 días posteriores al inicio del proyecto.

Tabla 4.3(c) Ejemplos de requisitos de vista de taxonomía total (continuación)

Requisito funcional del producto	El usuario de RR.HH. podrá recuperar las habilidades de los empleados y los datos de capacitación según categorías predefinidas.
Requisito de interfaz del producto	La apariencia del sistema SATURN será idéntica a la de cada sistema heredado local.
Requisito de especialidad del producto	El sistema SATURN utilizará tecnología de base de datos relacional.
Requisito ambiental del producto	El sistema SATURN no requerirá que se instale capacidad adicional de calefacción, ventilación y aire acondicionado (HVAC) en ningún lugar.
Requisito de rendimiento del producto	El sistema SATURN funcionará con un 97% de confiabilidad, 24 horas al día, los 7 días de la semana.

Fuente: Terry Bartolomé. Usado con permiso.

Tabla 4.3(d) Ejemplos de requisitos de vista RA

Necesidades y expectativas del cliente	Ejemplos
(Entrada de análisis de requisitos)	
Requisitos comerciales	Los gerentes necesitan acceso a datos oportunos y precisos sobre el personal para poder satisfacer las necesidades operativas.
Requisitos de usuario	El usuario necesita la capacidad de buscar personal en toda la empresa según conjuntos de habilidades predefinidos.
Requisitos del producto	Los formatos de datos se traducirán a través de los límites del sistema heredado al formato admitido por el sistema del usuario local.
Requisitos medioambientales	No habrá ningún impacto operativo en ningún usuario aparte del impacto en la recuperación de información causado por tener una mayor población de empleados entre los cuales seleccionar.
Especificaciones de requisitos del sistema (resultado del análisis de requisitos)	
Requisitos de alto nivel (o nivel de sistema)	El sistema SATURN mantendrá referencias cruzadas para los tipos de información contenidos en los sistemas heredados. Por ejemplo, el campo llamado "nivel_educación" en un sistema es el mismo que "educación" en otro.
	El sistema SATURN convertirá los datos de cada sistema heredado a los datos esperados por el usuario local. Por ejemplo, un título de maestría en un sistema podría reflejarse en otro sistema como "grado 17".
Requerimientos funcionales	El usuario local podrá buscar en todos los sistemas heredados en un área geográfica local, regional o nacional predefinida personal que cumpla con un conjunto de habilidades específico.
Requerimientos no funcionales	El sistema SATURN hará uso de la red pública comutada (PSN) y no requerirá líneas de comunicación dedicadas.
Requisitos derivados (o de diseño) y restricciones de diseño.	El sistema SATURN utilizará seguridad de comunicaciones de infraestructura de clave pública (PKI).
Requisitos de desempeño	El sistema SATURN admitirá hasta 20 usuarios simultáneos sin ninguna degradación notable del servicio.
	El sistema SATURN devolverá todos los conjuntos de habilidades disponibles al usuario dentro de 1 minuto de iniciar una búsqueda.
Requisitos de interfaz	El sistema SATURN deberá presentar una apariencia consistente con el sistema heredado de cada oficina local.

Fuente: Terry Bartolomé. Usado con permiso.

usuarios simultáneos. Nuestra máquina de prueba era mucho más pequeña de lo que iba a ser nuestra máquina de producción y, de hecho, uno de nuestros requisitos era determinar el tamaño y la configuración de la(s) máquina(s) que necesitábamos para cumplir con el requisito de rendimiento. Resultó que no había una forma razonable de extrapolar el rendimiento medido en nuestra caja de prueba al rendimiento esperado en una variedad de posibles cajas de producción. Además, ni siquiera podíamos comenzar a ejecutar los scripts de prueba en nuestra caja de prueba hasta que prácticamente hubiéramos completado el desarrollo de todas las funciones. Ninguno de estos hechos se conocía cuando nos comprometimos a fijar una fecha de entrega. De hecho, teníamos un requisito derivado del que no nos dimos cuenta: en este escenario, necesitábamos varios meses de estabilidad después del desarrollo de la versión 1 y antes del lanzamiento de la versión 1 en los que podíamos probar, medir, adquirir una nueva Lección aprendida: los compromisos de cronograma no deben hacerse hasta que se comprendan los requisitos.

Referencias

- [1] Colaboración para la mejora de procesos de ingeniería (EPIC), un modelo de madurez de capacidad de ingeniería de sistemas, versión 1.1. Pittsburgh, PA, Instituto de Ingeniería de Software, Universidad Carnegie-Mellon, 1995, en www.sei.cmu.edu/pub/documents/95.reports/pdf/mm003.95.pdf.
- [2] Norma EIA 632, "Procesos para diseñar un sistema", Arlington, VA, 1998.
- [3] Estándar IEEE 12207, "Procesos del ciclo de vida del software", Nueva York: IEEE, 1998.
- [4] Sitios web de Zachman Framework (por ejemplo, consulte www.zifa.com).
- [5] Inmon, WH, JA Zachman y JC Geiger, Almacenes de datos, almacenamiento de datos y el marco Zachman: gestión del conocimiento empresarial, Nueva York: McGraw Hill, 1997.
- [6] Hay, J., Análisis de requisitos: desde la visión empresarial hasta la arquitectura, Englewood Cliffs, Nueva Jersey: Prentice Hall, 2002.
- [7] Sitio web de CMMI, en www.sei.cmu.edu/cmmi.
- [8] Grady, JO, Análisis de requisitos de sistemas, Nueva York: McGraw-Hill, 1993.
- [9] Grady, JO, Validación y verificación del sistema, Boca Raton, FL: CRC Press, 1997.
- [10] Davis, AM, Requisitos de software: objetos, funciones y estados, Upper Saddle River, Nueva Jersey: Prentice Hall, 1993.
- [11] Whitten, N., "Cumplir con los requisitos mínimos: cualquier cosa más es demasiado", PM Network (septiembre de 1998), p. 19.
- [12] The Standish Group International, Inc., CHAOS Chronicles 2003 Report, West Yarmouth, MA: The Standish Group International, Inc., 2002, en www.standishgroup.com.
- [13] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.
- [14] Buede, DM, El diseño de ingeniería de sistemas: modelos y métodos, Nueva York: John Wiley e hijos, 2000.
- [15] Alexander, IF y R. Stevens, Writing Better Requisitos, Londres, Reino Unido: Addison-Wesley, 2002.

CAPÍTULO

5

Contenido

[Planifique el enfoque](#)[Resumen](#)[Caso de estudio](#)[Referencias](#)

Requisitos de reunión

La necesidad de recopilar requisitos se inicia con una solicitud de un cliente interno o externo. Las solicitudes pueden presentarse de muchas formas, incluida una solicitud de propuestas (RFP) [1], una SOW o una consulta formal o informal que describa una capacidad que se necesita. La solicitud inicia un conjunto de actividades de recopilación de requisitos. Es vital que la RA tenga un conocimiento profundo de estas actividades y adquiera experiencia en la realización de tareas relacionadas.

Mi experiencia me ha enseñado lo siguiente:

Se pierde mucho tiempo y esfuerzo en la fase de inicio del proyecto y en la realización de actividades de recopilación de requisitos. Hay un número de razones para esto:

1. El proyecto recién se está organizando y las cosas están confusas.
2. No existe una hoja de ruta ni una lista de verificación de actividades de inicio.
3. No todo el personal está presente; algunos todavía están siendo reclutados.
4. Todavía no hay mucha presión para cumplir el cronograma.
5. El cliente y los usuarios también intentan organizarse y empezar.

6. Es posible que el personal que trabajará en el desarrollo del producto final no comprenda completamente los objetivos del cliente y, en consecuencia, no pueda apreciar sus expectativas.

7. Un procedimiento eficaz y probado para la recopilación de requisitos.

Los siguientes pasos no están disponibles ni se utilizan.

Si el esfuerzo de recopilación de requisitos no es efectivo, se crea el escenario para que se desperdicie mucho tiempo y esfuerzo adicional cuando se inicia y realiza el trabajo técnico como resultado de requisitos deficientes. Por ejemplo, en una gran empresa de telecomunicaciones, se presupuestaron el 100% de los costos de desarrollo planificados.

para la reelaboración del software desarrollado basándose en experiencias previas, los requisitos establecidos no serían los que realmente se necesitaban.

Por lo tanto, la RA puede desempeñar un papel vital al garantizar que las actividades de recopilación de requisitos se planifiquen y realicen bien (eficazmente). Él o ella puede hacer mucho para garantizar que el tren (el proyecto) permanezca en las vías sugiriendo y recomendando prácticas, métodos, técnicas y herramientas efectivas y ayudando al PM, los clientes, los usuarios y los miembros del proyecto. equipo. Su función no se limita estrictamente a actividades relacionadas con los requisitos; más bien, el RA es un asesor valioso para el PM y todos los demás miembros del equipo del proyecto. Tómese unos momentos para revisar las nueve funciones de un RA (descritas en el Capítulo 2):

1. Trabajar en colaboración con clientes, usuarios y arquitectos y diseñadores de sistemas para identificar los requisitos reales de un sistema planificado o esfuerzo de desarrollo de software para definir el problema que debe resolverse;
2. Trabajar eficazmente con clientes y usuarios para gestionar requisitos nuevos y modificados para que el proyecto permanezca bajo control; instalar un mecanismo para controlar los cambios;
3. Esté alerta a las nuevas tecnologías que puedan ayudar;
4. Facilitar el proyecto en la reutilización de artefactos y el logro de repetibilidad;
5. Ayudar al proyecto y a sus clientes a visualizar una ruta de crecimiento desde el primer lanzamiento o versión de un producto a través de un conjunto de lanzamientos por etapas hasta el sistema o producto final;
6. Asesorar al proyecto (y al cliente) sobre los métodos, técnicas y herramientas automatizadas que están disponibles para respaldar mejor el trabajo y las actividades del proyecto relacionados con los requisitos;
7. Utilice métricas para medir, rastrear y controlar los requisitos relacionados. actividades y resultados del trabajo del proyecto;
8. Facilitar discusiones y mediar en conflictos;
9. Estudiar el dominio del área en la que se está utilizando el sistema o software. ing utilizado.

Estos roles proporcionan el contexto para este capítulo. El primer paso es planificar el enfoque.

Planifique el enfoque He sugerido

anteriormente el tremendo valor de dedicar algo de tiempo (en cualquier esfuerzo) a planificar el enfoque. Escribir (documentar) el enfoque planificado para abordar el trabajo relacionado con los requisitos en un plan de requisitos del proyecto. Como es el caso con otros planes, el plan de requisitos puede (y debe) revisarse y actualizarse con frecuencia durante el proyecto. Alguno

de los aspectos críticos de este plan se resumen en la lista de verificación proporcionada en la Tabla 5.1 y se analiza a continuación. Tenga en cuenta que no todos los aspectos son necesariamente siempre realizado de forma secuencial; algunos pueden realizarse simultáneamente. Nota que algunos aspectos son iterativos (se hacen repetidamente en función de la disponibilidad de información nueva o diferente). Mantenga la flexibilidad en su enfoque (siempre) para permitir que nueva información dé forma a mejoras en el enfoque.

Tabla 5.1 Lista de verificación para las actividades de recopilación de requisitos del proyecto

¿Hecho? Paso	Acción o Actividad
1	Revisar información histórica relacionada.
2	Revisar las políticas organizacionales relacionadas.
3	Identificar las partes interesadas del proyecto.
4	Desarrollar una estrategia para involucrar a los clientes y usuarios durante todo el desarrollo. esfuerzo
5	Escribir (e iterar) un documento de alcance y visión del proyecto
6	Desarrollar un plan de requisitos.
7	Proporcionar revisiones e inspecciones por pares de todos los productos de trabajo relacionados con los requisitos.
8	Iniciar un glosario del proyecto y una lista de acrónimos del proyecto.
9	Decidir el enfoque de ciclo de vida que se utilizará en el proyecto.
10	Comenzar a adaptar el proceso de requisitos corporativos (u otros)
11	Establecer un mecanismo para evolucionar los requisitos reales a partir de los requisitos establecidos.
12	Proporcionar capacitación relacionada con los requisitos para los participantes del proyecto, incluidos los clientes. y usuarios, y para RA
13	Vuelva a escribir los requisitos de alto nivel del sistema o software a medida que avanza en el pasos iniciales
14	Iniciar el desarrollo de los requisitos reales basándose en los requisitos establecidos.
15	Iniciar la documentación de la justificación de cada requisito.
detalles	Establecer un mecanismo para controlar los cambios en los requisitos y los nuevos requisitos.
	17 Realizar el enfoque de verificación y la planificación de validación.
18	Seleccionar las prácticas, métodos y técnicas que se utilizarán para recopilar la requisitos
19	Comenzar la consideración y selección de una herramienta automatizada de requisitos, identificación de los atributos que serán necesarios para cada requisito, y la composición de el repositorio de requisitos
20	Seleccionar y adquirir la herramienta de requisitos automatizados
21	Cargue los requisitos reales iniciales en la herramienta de requisitos seleccionada, etique cada uno requisito de forma única e iniciar la asignación de información de atributos apropiada a cada requerimiento
22	Realizar levantamiento de requisitos
23	Involucrar a arquitectos y diseñadores de sistemas en las revisiones de los requisitos.
24	Desarrollar la estrategia de trazabilidad a utilizar.
25	Identificar los requisitos que se cumplirán en la primera versión o productos iniciales. (priorizar necesidades reales)
26	Establecer un enfoque para una prueba de concepto, prototipo u otra aproximación de el producto del trabajo
27	Incorporar las mejores prácticas de requisitos y obtener soporte de gestión para Ingeniería de requisitos eficaz (incluido un enfoque de calidad integrado).
28	Recopilación completa de requisitos para la primera versión.

Cada uno de los pasos del enfoque de recopilación de requisitos se analiza a continuación y se proporcionan varias referencias sugeridas para indicarle información adicional. Piense en estos pasos como un procedimiento para implementar dos de los tres subprocessos del proceso de requisitos: Evaluar requisitos nuevos/cambiados y controlar cambios; y Comprender las necesidades y expectativas del cliente, RE100 y RE200 en la terminología de Prácticas de requisitos eficaces (consulte [2, págs. 114 y 115] para ver los diagramas de flujo del proceso real). Como siempre, adapte (modifique) el enfoque según sea necesario para su situación particular y su proyecto y entorno organizacional; es posible que pueda eliminar algunos de los pasos o tal vez desee agregar pasos.

Es probable que le resulte beneficioso tener copias de los diagramas de flujo frente a usted para poder considerar los cambios que le gustaría realizar. Quizás quieras visitar mi sitio web (www.ralphyoung.net); vaya al botón "Artefactos reutilizables" y busque "Proceso de requisitos de muestra". Esto le permitirá imprimir cuatro diagramas de flujo, un proceso macro (de alto nivel) y tres microprocesos (nivel inferior) o subprocessos para el flujo macro.

Tenga en cuenta el enlace en esa página web para las "Descripciones de procesos" que explican los cuatro diagramas de flujo (propósito del proceso, estándares y referencias, procesos relacionados, clientes del proceso, requisitos del cliente, criterios de entrada, entradas, salidas, criterios de salida, responsabilidades, herramientas), recursos y métricas sugeridas). Digerir estos artefactos te proporcionará muchas ideas.

1. Revisar la información histórica relacionada.

En cualquier esfuerzo, hay materiales disponibles que necesitan ser leídos, digeridos y analizados. Ejemplos de dicha información incluyen descripciones de los sistemas heredados, declaraciones sobre las necesidades de nuevas capacidades, documentos técnicos, descripciones de sistemas relacionados desarrollados por otras organizaciones, estudios de investigación, personas con las que podría reunirse para obtener información (como proponentes o defensores de una capacidad necesaria o de un nuevo sistema), etc. Sea abierto y minucioso al buscar y revisar materiales. Organice los materiales que encuentre de manera que le ayude a evaluar su importancia y valor relativos y también le permita recuperar información cuando sea necesaria. Piense en los materiales en el contexto de los otros pasos del enfoque de recopilación de requisitos enumerados en la Tabla 5.1.

2. Revisar las políticas organizacionales relacionadas.

En cualquier organización, debe existir un conjunto de políticas organizacionales sobre cómo se debe realizar el desarrollo de sistemas y software. Encuentre estas políticas; leerlos y digerirlos. Asegúrese de comprenderlos y poder aplicarlos en su trabajo. Pídale aclaraciones a sus compañeros de trabajo y a su gerente, si es necesario. Si existen políticas y nadie les presta mucha atención, esto en sí mismo es una oportunidad de mejora continua para la organización: involucre al personal de control de calidad y de ingeniería de procesos. Descubra si existe una biblioteca de procesos relacionados, planes de muestra como planes de proyecto y otros.

métricas que deben usarse, métodos, técnicas, herramientas automatizadas y manuales disponibles, lecciones aprendidas (o al menos observadas) de otros proyectos (anteriores), etc. Asegúrese de tener un conocimiento integral de los recursos disponibles para realizar su trabajo. Busque plantillas, listas de verificación, presentaciones y archivos de sesiones de familiarización y capacitación que puedan resultar útiles. Antes de embarcarse en una tarea para crear un artefacto, verifique si ya existe uno que pueda reutilizar o al menos usar como guía. Trate de evitar reinventar la rueda, es decir, recrear un artefacto cuando ya existe una plantilla o ejemplo. Si alguien ya ha hecho lo que usted está a punto de hacer, es probable que pueda ahorrar tiempo y esfuerzo al saberlo.

3. Identificar las partes interesadas del proyecto.

Una parte interesada es cualquier persona que tenga interés en el proyecto y cualquiera que se verá afectado por el sistema. Piense en clientes (aquellos que pagan por el trabajo), usuarios (personas que realmente utilizarán el sistema), asesores (como expertos legales o reguladores que tienen información relevante sobre los requisitos), grupos de proyectos que participan en el desarrollo del sistema. (como ingeniería de sistemas, ingeniería de software, control de calidad, CM, control de proyectos, documentación, capacitación, pruebas, etc.). Al igual que al diseñar un proceso, siempre hay más partes interesadas de las que pensamos inicialmente. Ian Alexander sugiere utilizar una "Plantilla de análisis de partes interesadas" [3] para identificar a las partes interesadas. John Boardman Associates (JBA) desarrolló la plantilla (correo electrónico: ian@jba.net). Las figuras 5.1 y 5.2 representan los roles de las partes interesadas y los puntos de vista de los roles.¹

El enfoque de ingeniería de sistemas sugiere desarrollar un concepto de operaciones (CONOPS) que se centre en las metas, los objetivos y las capacidades generales deseadas de un sistema o producto nuevo o mejorado.² Los escenarios operativos (secuencias de eventos esperados durante la operación de los productos del sistema) son desarrollado (ver la discusión de escenarios como una técnica de recopilación de requisitos más adelante en este capítulo). Estos incluyen las condiciones ambientales, tasas de uso, entradas al sistema (a veces denominadas estímulos esperados) y salidas (respuestas). Los escenarios operativos son el marco ideal para los requisitos de misión/negocio/usuario y también son útiles para identificar y aclarar aspectos del sistema.

1. Véase también el Capítulo 13 de Ingeniería de requisitos de Sommerville y Sawyer: una guía de buenas prácticas (Nueva York: John Wiley & Sons, 1997), que proporciona un enfoque sistemático para recopilar requisitos desde múltiples puntos de vista, llamado PREview. Un sitio web relacionado, www.info.comp.lancs.ac.uk/publications/index.phtml, bien merece una visita.

2. EIA 632, "Procesos para diseñar un sistema", proporciona un enfoque integral, estructurado y disciplinado para todas las fases del ciclo de vida. El proceso de ingeniería de sistemas se aplica de forma iterativa durante todo el ciclo de vida del sistema. El concepto operativo facilita separar los requisitos de la misión de otros requisitos del sistema, identificando escenarios que dictan la interacción entre el sistema y otros sistemas (incluidas las personas) y se centra en gran medida en las entradas y salidas del sistema.

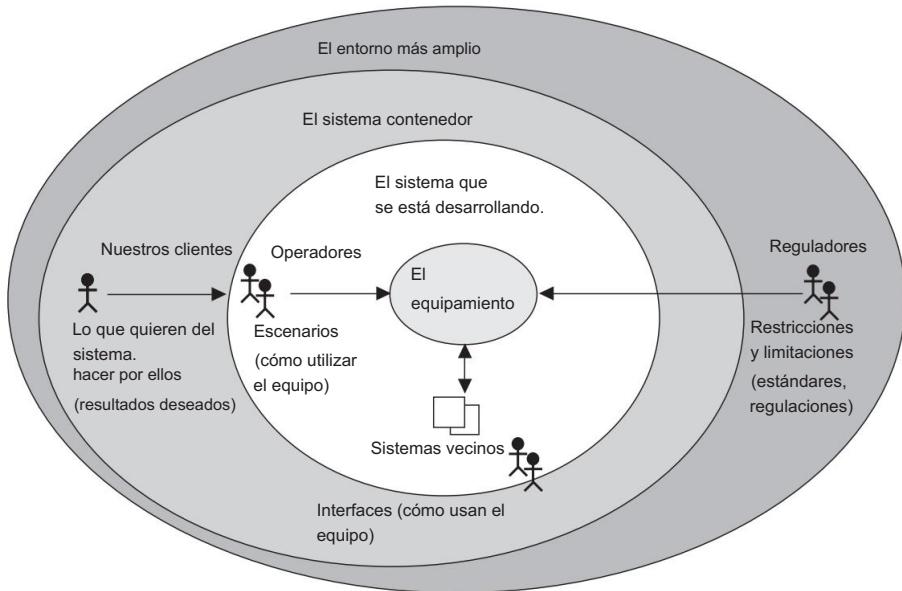


Figura 5.1 Roles de las partes interesadas. (Fuente: John Boardman Associates (JBA). Usado con autorización).

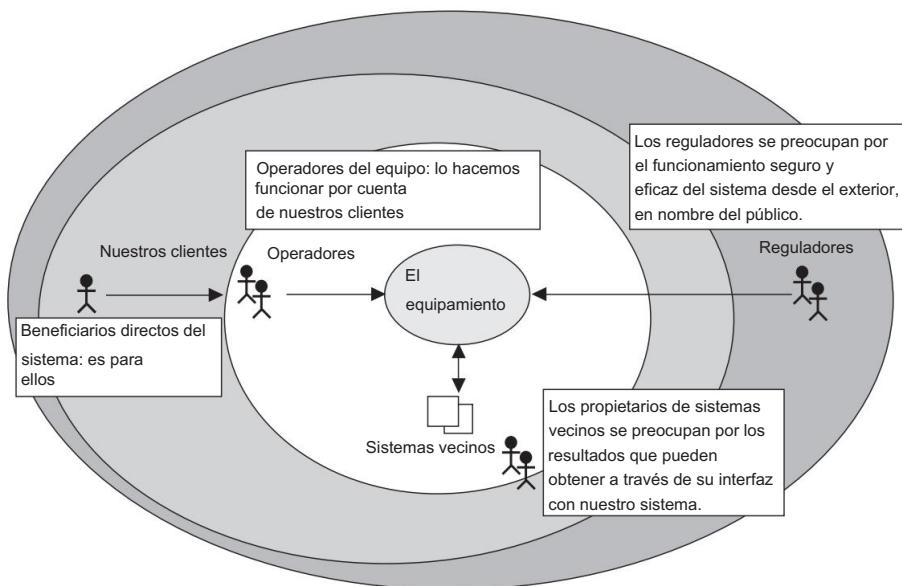


Figura 5.2 Puntos de vista de los roles. (Fuente: John Boardman Associates (JBA). Usado con autorización).

Helen Sharp [4] sugiere utilizar un enfoque recursivo para identificar a las partes interesadas: pida a su punto de contacto inicial una lista de partes interesadas, luego pregunte a cada persona de esa lista quién más tiene interés, y así sucesivamente, hasta que usted no lo sepa.

encontrar más partes interesadas (a veces este enfoque se conoce como “pelar la cebolla”). Sugiere nombrar cuatro grupos de partes interesadas en la línea de base (usuarios, desarrolladores, legisladores y tomadores de decisiones) y luego explorar la red de partes interesadas alrededor de la línea de base. Ellen Gottesdiener [5] sugiere incluir “usuarios indirectos” (o “usuarios secundarios”): personas que entrarán en contacto con los resultados del sistema (como archivos e informes) o con subproductos del sistema (como decisiones).

Es importante comprender la composición del cliente y la política interna. Por ejemplo, si el sistema propuesto reemplazará los sistemas existentes, cada uno de ellos propiedad de un grupo establecido (léase “atrincherado”), y el cambio se está forzando desde arriba, la estrategia utilizada para involucrar a los clientes podría ser muy diferente de la utilizada si el sistema propuesto El sistema consiste en integrarse con varios sistemas existentes y dejarlos esencialmente intactos. Una situación ideal podría ser construir un nuevo sistema para un solo cliente, pero frecuentemente los proyectos tienen varios clientes. Según la experiencia de un RA, una situación desafiante era un sistema que tenía tres agencias separadas como clientes: dos de carácter fuerte y uno dócil. Tratar de resolver la situación política era demasiado complicado para la mayoría de las personas razonables. Un segundo contratista que tuvo un impacto negativo significativo en los esfuerzos del proyecto complicó aún más la situación. En el análisis final, comprender el proyecto desde la perspectiva de la participación de varias organizaciones puede ser tan importante como comprender las necesidades de las partes interesadas.

4. Desarrollar una estrategia para involucrar a los clientes y usuarios durante todo el desarrollo. esfuerzo.

La experiencia nos muestra que los proyectos que involucran a clientes y usuarios durante todo el esfuerzo de desarrollo tienen más éxito. La razón es que existe una comunicación efectiva. Sin una comunicación efectiva, el cliente/usuario y el desarrollador se pierden de vista y pierden sus perspectivas. Si bien esta tarea no es principalmente función de la AR, es fundamental para el desempeño exitoso de sus tareas que esta estrategia se desarrolle e implemente. Abogar como parte del equipo del proyecto para que se lleve a cabo la estrategia.

Una estrategia para involucrar a clientes y usuarios es la “asociación”. Consulte el Inserto 5.1 y Prácticas efectivas de requisitos [2, págs. 30–41], para una discusión de este enfoque. Esta bien podría ser la pepita que potencie su estrategia general. He utilizado la colaboración y puedo decirles que funciona muy bien.

Otra estrategia es utilizar talleres de requisitos y otras técnicas de recopilación de requisitos que involucren a los clientes y usuarios de manera interactiva en las decisiones que se toman. Es mejor si el cliente y los usuarios participan íntimamente durante todo el ciclo de vida. Otra técnica para ayudar con esto es incluir a los clientes en un tablero de control de cambios del proyecto (CCB) para facilitar su participación activa en las actividades de desarrollo.

Tenga en cuenta que muchos productos comerciales están diseñados para las necesidades del público objetivo y no para las de un grupo finito de partes interesadas. Se utilizan grupos focales y otras técnicas para generar un modelo de comportamiento del cliente que

Inserto 5.1—Descripción de la asociación

La asociación es un proceso estructurado diseñado para crear una atmósfera de compromiso, cooperación y resolución colegiada de problemas entre organizaciones e individuos que trabajan juntos en un proyecto. Un entorno de seguridad y confianza mutuas es esencial para una asociación eficaz.

La asociación utiliza un facilitador externo capacitado para desarrollar declaraciones de visión, objetivos comunes, principios rectores, procedimientos de resolución de problemas y métodos de evaluación para ayudar a garantizar el éxito del proyecto. El proceso normalmente se inicia en un taller al inicio del proyecto. En el taller todos son considerados iguales. No se debe permitir que ningún participante u organización domine el proceso del taller. Todas las partes deben reconocer que la asociación es la construcción del equipo con el objetivo de lograr el compromiso para el éxito del proyecto.

La asociación no cambia las obligaciones contractuales de las partes. Facilita la manera en que las partes contratantes se tratan entre sí durante el curso de la ejecución del contrato. Crea un clima en el que los intereses y expectativas de las partes contratadas son más fácilmente alcanzables. Con este fin, durante el taller se crea una carta escrita que establece los intereses comunes de las partes para reducir las disputas costosas y que consumen mucho tiempo, así como para mejorar las comunicaciones en beneficio de todas las partes.

Todas las partes contratantes tienen un interés económico en el éxito del proyecto. Así como los clientes se preocupan por obtener un buen valor por su dinero, los proveedores están en el negocio para obtener una ganancia justa por los servicios que brindan. Cuando un proveedor se ve exprimido para obtener ganancias, la calidad del trabajo y las relaciones comerciales pueden verse afectadas o destruidas, generando hostilidad y litigios costosos y prolongados. Sacar a los buenos proveedores del negocio no es lo mejor para nadie. El objetivo a largo plazo de todo propietario debe ser mantener a los buenos contratistas en el negocio para que las licitaciones competitivas sean lo más sólidas posible en el futuro.

La asociación debe incluir a los usuarios finales del sistema. Los clientes deben participar en el proceso de asociación de principio a fin. Proporcionan información valiosa sobre las necesidades de su proyecto y pueden participar en sesiones de resolución de problemas en el taller y reuniones de seguimiento, y pueden comprender mejor a dónde se destina su dinero cuando se requieren modificaciones del contrato. Un objetivo típico del equipo asociado es entregar un proyecto de calidad al cliente que satisfaga sus necesidades funcionales y financieras.

restricciones. La satisfacción del cliente es un ingrediente esencial en prácticamente todos los esfuerzos de asociación.

Los costos de asociación generalmente incluyen de uno a tres días del tiempo de los participantes al inicio de un proyecto para realizar el taller y cualquier sesión de seguimiento posterior. También existen los honorarios del facilitador para estas reuniones. Sin embargo, estos costos son pequeños cuando un proyecto se entrega dentro del presupuesto del cliente, generando ganancias para el proveedor y

Inserto 5.1—Descripción de la asociación (continuación)

antes de lo previsto. El tiempo requerido es pequeño en comparación con el tiempo ahorrado durante el transcurso del proyecto. En el taller inicial, muchos equipos asociados identificarán y resolverán problemas potenciales.

Esto puede evitar semanas o meses de retrasos más adelante en el proyecto. Durante el proyecto, el equipo utiliza procedimientos de asociación para trabajar juntos para evitar otros retrasos en el cronograma y lograr los objetivos del proyecto.

Cualquier persona que tenga un impacto directo en el éxito del proyecto debe participar y asistir al taller inicial de asociación para convertirse en miembro del equipo. La participación en el taller asegura una comprensión de los objetivos comunes y la visión mutua del equipo. Cuando una persona no está presente en el taller, la asociación puede no tener significado para esa persona.

Las sesiones de seguimiento a veces se retrasan o cancelan debido a las presiones del desempeño del proyecto y los cronogramas de finalización. Esto impide que el proceso de asociación funcione eficazmente cuando más se necesita. Se necesita un fuerte compromiso de asociación para garantizar que las sesiones de seguimiento se lleven a cabo cuando todos estén ocupados.

Para evitar este problema, se debe establecer un cronograma para las sesiones de seguimiento durante el taller inicial. Se pueden establecer fechas para cumplir con los requisitos de programación de todos los miembros del equipo.

Los subcontratistas deben participar para informar a los demás participantes sobre sus intereses y valor para el proyecto. Los subcontratistas, a su vez, aprenderán y apreciarán lo que es importante para los demás participantes del proyecto. La participación de los subcontratistas en el taller puede ayudar a prevenir disputas durante la ejecución.

Dos elementos esenciales de una relación de asociación exitosa son la confianza y la comunicación. Si algún miembro del equipo siente que otros miembros del equipo se están aprovechando, la confianza se verá afectada negativamente. Se debe alentar a los miembros del equipo a que comuniquen este sentimiento a los demás de manera clara y oportuna cuando surja el problema.

Fuente: Chales Markert, facilitador de asociaciones.

Iuego sirve como marcador de posición. Este modelo debe validarse y verificarse tal como se haría con cualquier otro paso del proceso de requisitos, pero con frecuencia está muy mal elaborado y sólo más tarde se descubren las preferencias reales del mercado.

5. Escribir (e iterar) un documento de visión y alcance del proyecto.

Este documento solo debe tener unas pocas páginas, pero es muy importante porque ayuda a todas las partes interesadas a comprender mejor el sistema o software planificado. El documento debe presentar lo siguiente:

Los requisitos comerciales y los objetivos comerciales relacionados para la sistema;

La visión de la solución, consolidada desde los distintos stakeholders, y descrito en términos de lo que se necesita para cumplir con el negocio objetivos;

El alcance del sistema previsto en términos de lo que debería y no debe incluir.

Karl Wiegers proporciona una discusión instructiva y reveladora sobre por qué este documento es importante y un resumen detallado de lo que debe incluir en los Requisitos de software [6, págs. 81–93]. Le sugiero que repita el documento de visión y alcance del proyecto porque será más útil a medida que usted y otros obtengan una comprensión más completa del sistema planificado. Descubrirás que la visión puede cambiar un poco y el alcance mucho.

Tenga cuidado con el crecimiento de los requisitos (también conocido como “aumento progresivo” de los requisitos, donde los requisitos se agregan sin tener controles de RM implementados). Weinberg [7] utiliza el término fuga de requisitos para referirse a requisitos no oficiales que se agregan cuando en realidad no son necesarios; consulte la discusión de este importante tema en Prácticas efectivas de requisitos, [2, págs. 221-229]. Una vez más, esto demuestra el valor de la planificación y refuerza el valor de invertir en actividades de requisitos iniciales, antes de que se inicie el resto del trabajo técnico.

6. Desarrollar un plan de requisitos.

He subrayado la importancia de la planificación con respecto a cualquier actividad, y las actividades de requisitos no son una excepción. Está familiarizado con varios tipos de planes para un proyecto, como el plan de gestión de proyectos (PMP), SEMP, SDP, plan de control de calidad, plan CM, plan de capacitación y otros. Utilice la tabla de contenidos sugerida para un plan de requisitos del proyecto que se proporciona en la Tabla 5.2 para ayudarle a documentar las actividades necesarias relacionadas con los requisitos y lo que se hará, así como para discernir lagunas y necesidades [8].

La experiencia de Ellen Gottesdiener [5] es que el RA experimentado planificará de tres a cuatro iteraciones a través del proceso de desarrollo de requisitos. Tanto los clientes internos como los externos deben seguir cada iteración con una revisión formal o informal. Este enfoque puede parecer excesivo, especialmente si es nuevo en la recopilación de requisitos. La experiencia de Ellen Gottesdiener da crédito al valor de invertir en el proceso de requisitos para identificar los requisitos reales antes de comenzar otro trabajo. Este puede ser un factor importante para reducir el retrabajo (el retrabajo representa del 40% al 50% de los costos totales de la mayoría de los proyectos). Asegúrese de que todas las partes interesadas acepten el enfoque que ha definido. Si bien existen excelentes razones para planificar y realizar múltiples iteraciones de los requisitos, es probable que tenga que defender el tiempo y los recursos necesarios para lograrlo.

7. Prever revisiones e inspecciones por pares de todo el trabajo relacionado con los requisitos productos.

Tabla 5.2 Ejemplo de tabla de contenidos para un plan de requisitos del proyecto

Objetivo
Resumen de contrato, proyecto o tarea
Uso del proceso de ingeniería de sistemas
Estrategia sugerida para abordar los problemas de requisitos de la industria
El proceso de requisitos del proyecto
Importancia del proceso de requisitos para superar los requisitos
Problemas
Requisitos Proceso, Mecanismos, Prácticas, Métodos, Técnicas y Herramientas a Utilizar en el Proyecto
Enfoque sugerido para involucrar a clientes y usuarios
Mejores prácticas de requisitos de la industria
Referencias consultadas
Apéndices
Un proceso de requisitos a medida (diagramas de flujo y procesos descripciones)
B Resumen del proceso de asociación
C Criterios de un buen requisito
D Directrices para el desarrollo de sistemas basados en requisitos Consideraciones

En primer lugar, asegúrese de que exista un proceso de revisión por pares eficaz que realmente se utilice en todo el proyecto. Si no existe uno, trabaje con su gerente y el PM para adoptar y utilizar estas mejores prácticas de la industria. Bríndele la discusión en Prácticas efectivas de requisitos [2, págs. 248–250]. Trabajar duro para promover el uso de revisiones e inspecciones por pares. Ahorrártan tiempo, dinero y esfuerzo y también mejorará la calidad y la satisfacción del cliente. Utilice las revisiones por pares de software de Wiegers: una guía práctica [9] para implementar el enfoque de inspección y revisión por pares que mejor se adapte a su proyecto. (Aunque el título del libro de Wiegers dice "software", la información sobre revisiones por pares en el libro se aplica a cualquier producto de trabajo). A pedido, Northrop Grumman IT DES [10] puede proporcionar dos cursos de capacitación de 2 horas y soporte asociado para lanzar El proceso de revisión por pares de su proyecto u organización:

1. "Capacitación para participantes en la revisión por pares";
2. "Capacitación para moderadores de revisión por pares".

Comuníquese con la propietaria del proceso de revisión por pares, Penny Waugh, en PWaugh@ngc.com para obtener información.

Algunos proyectos reciben asesoramiento de no realizar revisiones por pares e inspecciones de documentos relacionados con requisitos en sus proyectos. En mi opinión, no contar con revisiones por pares es un riesgo imperdonable para el proyecto. La experiencia ha demostrado que ningún producto de trabajo debe desarrollarse, por simple que sea, sin haber sido revisado por al menos un par que tenga conocimientos sobre el tema del producto de trabajo. El punto es que (1) las palabras que uso pueden

o no comunicar mi intención de manera efectiva; (2) otros pueden tener una perspectiva diferente; (3) cualquier producto de trabajo se beneficiará de las ideas, sugerencias y correcciones proporcionadas por otros; y (4) tiempo y dinero se salvan cuando los defectos se identifican antes de lo que lo harían de otra manera ser. El desafío es encontrar personas lo suficientemente independientes como para brindar una un par de ojos nuevos, pero lo suficientemente familiarizados con el tema general y el proceso para Evite sobrecargas excesivas de revisión o hacer que los artículos sean engañosos o sin sentido. comentarios.

8. Iniciar un glosario del proyecto y una lista de acrónimos del proyecto.

Uno de los problemas que, según mi propia experiencia, ha puesto en peligro el trabajo en equipo, ha provocado demasiadas discusiones (lectura, confusión, frustración y retrasos), e incluso destruyó muchas buenas relaciones interpersonales es que Nosotros, los técnicos, tendemos a tener opiniones muy firmes sobre las definiciones de palabras particulares, y tendemos a resistirnos a seguir adelante cuando No sé ni entiendo un acrónimo que se está utilizando. Recomiendo y defiendo encarecidamente que cada proyecto desarrolle un glosario de proyecto y una lista de acrónimos del proyecto. Estos artefactos deben crearse lo antes posible, por ejemplo, como parte de la documentación de la visión del proyecto, y ampliarse a medida que el proyecto madure. Además de las siglas, lo básico La nomenclatura asociada con un concepto de sistema emergente puede tener un gran impacto en la libertad de diseño. (Lenguaje funcional versus físico Las opciones en particular tienen un gran beneficio al principio del ciclo de vida del proyecto.) Incluya palabras y siglas que sean aceptables y utilizadas por los clientes y usuarios; por ejemplo, nos referimos al conocimiento de la información del cliente. área como dominio de conocimiento o experiencia; personas que son extremadamente con conocimientos en un área particular se les conoce como expertos en la materia (PYMES). En aras de que el trabajo del proyecto se lleve a cabo rápidamente, hagamos lo siguiente:

Acordar definiciones de las palabras que usamos con las que todos podamos vivir. Esto no requiere que la definición consensuada sea la de cada persona. definición favorita de la palabra, solo que todos nosotros en nuestro proyecto El equipo puede vivir con la redacción de la definición. Sugerencia: usar un pulgar hacia arriba ("lo apoyo"), un pulgar hacia abajo ("no puedo vivir con esto"), o la técnica de los pulgares hacia los lados ("puedo vivir con ello") en las reuniones del proyecto para lograr un consenso rápidamente. Si la gente tiene problemas ("pulgares abajo"), pregunte: "¿Qué se necesita para convencerte de que puedas vivir?" ¿Con el enfoque o definición que la mayoría de la gente considera aceptable?" Si tienes un ambiente de TEAMWORKS, deberías estar capaz de lograr consenso sobre la mayoría de los temas fácilmente. (UN TRABAJO EN EQUIPO El entorno es un entorno de trabajo en el que trabajar juntos como Un equipo eficaz es valorado y apreciado y donde los compañeros de trabajo se apoyan unos a otros de forma proactiva. Consulte el Capítulo 8 para obtener más información. relacionado con el trabajo en equipo.) Si no puede, tal vez tenga una "causa especial de variación", como una persona escandalosa o diferente.

perspectivas.³ No tengo una cura de oro para las causas especiales de variación, sólo que debemos buscar constantemente las causas fundamentales de los problemas y cuestiones, utilizar nuestros múltiples espacios para generar ideas sobre contramedidas, priorizar las contramedidas de acuerdo con su efectividad percibida, implementarlas y luego evalúe si las contramedidas han tenido el impacto deseado en las causas fundamentales. Si no lo han hecho, es hora de identificar y seleccionar otras contramedidas.

Desarrollar una lista de acrónimos del proyecto que incluya todos los acrónimos que encuentran todos aquellos que trabajan en el proyecto. Esto se puede lograr poniendo la lista de siglas en un servidor compartido para que todos puedan tener acceso a ella y agregando siglas a medida que se encuentren. Además, cualquiera que encuentre un acrónimo puede utilizar este recurso para intentar averiguar qué significa. ¡No tiene ningún sentido que las reuniones de proyectos, sesiones de capacitación, sesiones informativas, etc., se estancen porque alguien no conoce un acrónimo!

9. Decidir el enfoque de ciclo de vida que se utilizará en el proyecto.⁴

Esta decisión puede parecerle fuera de su ámbito. Sin embargo, en mi experiencia, la mayoría de los proyectos no dan suficiente consideración a esta decisión.

Muy a menudo, los proyectos tienden a obstaculizar las actividades de diseño y desarrollo sin hacer explícito el enfoque del ciclo de vida. El proyecto debe seleccionar un enfoque de ciclo de vida específico. Esta decisión es importante porque los diferentes modelos de ciclo de vida tienen fortalezas y debilidades, y algunos son más apropiados para dominios particulares.⁵ Una pregunta clave es: ¿Qué fracción del diseño final diría que se conoce en este momento? Los proyectos con sólo cambios menores en los diseños existentes son muy diferentes de aquellos que requieren un trabajo fundamental. Tabla de estudio 5.3 y Figuras 5.3 a 5.6. Revise la información relacionada proporcionada en las referencias de Reed Sorenson [11] y Barry Boehm [12]. Si está considerando el enfoque del modelo espiral, lea "El modelo espiral como herramienta para la evolución" de Boehm y Hanse.

3. Ian Alexander informa de sus experiencias de consultoría que sus compañeros de trabajo a menudo insisten en usar palabras específicas de maneras que impiden el entendimiento común y que retrasan e incluso impiden el progreso para llegar a un consenso. Él cree que la causa fundamental de este grave problema no es el alboroto, sino más bien que las personas tienen diferentes perspectivas sobre las cosas. Esto enfatiza por qué es tan importante tener un ambiente de TRABAJO EN EQUIPO, así como utilizar mecanismos (como los pulgares hacia arriba, hacia abajo o hacia los lados y un glosario de proyectos) para llegar a un consenso y seguir adelante.
4. Nuestro agradecimiento a Rich Raphael de Northrop Grumman IT DES por desarrollar estos materiales y proporcionar análisis de modelos de ciclo de vida e ilustraciones de los mismos.
5. El subsecretario de Defensa estadounidense para adquisiciones, tecnología y logística, EC Aldridge Jr., publicó un memorando fechado el 12 de abril de 2002, expresando su preferencia por estrategias de adquisición evolutivas basadas en un proceso de desarrollo en espiral. Un punto de contacto para obtener más información es Skip Hawthorne (skip.hawthorne @osd.mil). El memorando reconoce que existe "confusión sobre lo que significan estos términos y cómo el desarrollo en espiral afecta diversos procesos, como la contratación y la generación de requisitos, que interactúan con una estrategia de adquisición evolutiva".

Tabla 5.3 Comparación de modelos de ciclo de vida

	Cascada	incremental	Evolutivo	Espiral
Descripción	Enfatiza la finalización de una fase de desarrollo antes de pasar a la siguiente fase; congelar los productos de una fase antes de pasar a la siguiente. Debe utilizar un mecanismo de cambio formal para realizar cambios en los requisitos.	Realiza la cascada en tramos superpuestos.	La etapa de desarrollo se realiza como una serie de incrementos. Cada incremento construye un subconjunto del sistema completo. Un incremento es un ciclo de vida completo de análisis, diseño, codificación, pruebas e integración. Puede ser entregado al usuario, pero no necesariamente. Si se entrega, es posible que haya algunos acabados: optimización, empaquetado y similares, después de la construcción.	Divide el desarrollo del sistema en cuatro actividades básicas: planificación, análisis de riesgos, ingeniería y evaluación. Dentro de cada bucle en espiral, se identifican los riesgos y se intentan mitigarlos antes de proceder a la actividad de ingeniería de la espiral.
Fortalezas	<p>modelado después de los procesos en otras disciplinas, lo que facilita a los gerentes comprender y aceptar.</p> <p>En este modelo cada fase se define por un conjunto de funciones, objetivos, hitos y entregables, lo que hace que el proceso sea muy visible y el proyecto sea más fácil de seguir.</p> <p>Dado que los requisitos y especificaciones se determinan en el principio, el primer ministro está en mejores condiciones de determinar sus necesidades de recursos y establecer horarios.</p>	<p>No es necesario especificar/aclarar completamente los requisitos desde el principio. Cada incremento es completado, se aclaran los requisitos.</p>	<p>Se centra en la atención temprana y continua. entrega de partes interesadas definidas por requisitos valor.</p> <p>Permite que los requisitos detallados surjan gradualmente.</p>	<p>Evita algunas de las dificultades de modelos de software existentes mediante el uso de un enfoque basado en riesgos.</p> <p>Intenta eliminar errores en las primeras fases.</p> <p>Proporciona mecanismos para el control de calidad.</p> <p>Aplicable a otro tipo de proyectos.</p> <p>Funciona bien para un proyecto complejo, dinámico e innovador.</p> <p>La reevaluación después de cada fase permite cambios en las perspectivas de los usuarios, los avances tecnológicos o las finanzas.</p>
Debilidades	<p>No funciona muy bien en situaciones donde los requisitos no están bien definidos en el comienzo de proceso.</p>	<p>Existe una tendencia a posponer los problemas difíciles para el futuro con el fin de demostrar temprano éxito para gestión.</p>	<p>Requiere un cambio de mentalidad cultural respecto de lo convencional métodos.</p>	<p>Carece de una guía de proceso explícita para determinar objetivos, limitaciones y alternativas.</p>

Tabla 5.3 Comparación de modelos de ciclo de vida (continuación)

	Cascada	incremental	Evolutivo	Espiral
Debilidades	<p>Las principales del modelo Su debilidad es el costo de cambiar los requisitos. Cuanto más avanza un proyecto, más costoso resulta un cambio en los requisitos.</p> <p>El cliente no ve un producto funcional hasta una etapa avanzada del ciclo de vida. Cuando el cliente tiene la oportunidad de opinar sobre el producto, cualquier error u omisión es muy costoso de corregir.</p> <p>Los proyectos reales rara vez fluyen de forma secuencial.</p> <p>Aunque es posible reiterar, la reiteración tiende a causar confusión a medida que avanza el proyecto.</p>	<p>Es difícil gestionar y medir el proyecto porque no se puede determinar cuándo se completarán todos los requisitos.</p>	<p>Requiere cierta formación y experiencia para aplicar el método de forma eficaz.</p>	<p>Proporciona más flexibilidad de lo conveniente para muchas aplicaciones.</p> <p>Experiencia en evaluación de riesgos: La evaluación de los riesgos del proyecto y su resolución no es una tarea fácil. Se necesita mucha experiencia en proyectos para realizar las tareas con éxito.</p>
Dominio de aplicaciones	Sistemas que tienen requisitos bien definidos desde el principio y sistemas donde los costos y cronogramas deben ser determinarse de antemano.	Muy adecuado para sistemas donde los requisitos no se pueden especificar.	Muy adecuado para sistemas donde los requisitos no pueden especificarse antes del inicio de las actividades de desarrollo del ciclo de vida.	Proyectos complejos, dinámicos, innovadores y ambiciosos realizados en equipos internos (no necesariamente limitados al software).

Fuente: Richard Rafael. Usado con permiso.

Adquisición” [13], que explica las mejoras al modelo espiral original que ahora se consideran esenciales para su uso. Tenga en cuenta que el enfoque de ganar-ganar es objeto de trabajo continuo por parte de Boehm y otros. Ponga la información a disposición de otros miembros de su equipo de proyecto (incluido el cliente), mantenga debates y llegue a un consenso sobre el modelo de ciclo de vida que mejor satisfaga las necesidades de su proyecto. Tenga en cuenta que algunos requisitos no se pueden conocer hasta que los clientes y usuarios comiencen a utilizar el sistema. Esta preocupación se aborda mejor mediante el uso de un enfoque de desarrollo incremental.

Algunos expertos de la industria se preguntan si los modelos evolutivo y espiral son realmente diferentes. El propio Boehm habla de “evolución” al describir el modelo espiral. Algunos expertos de la industria creen que los modelos incremental y evolutivo son muy diferentes e incompatibles; es muy

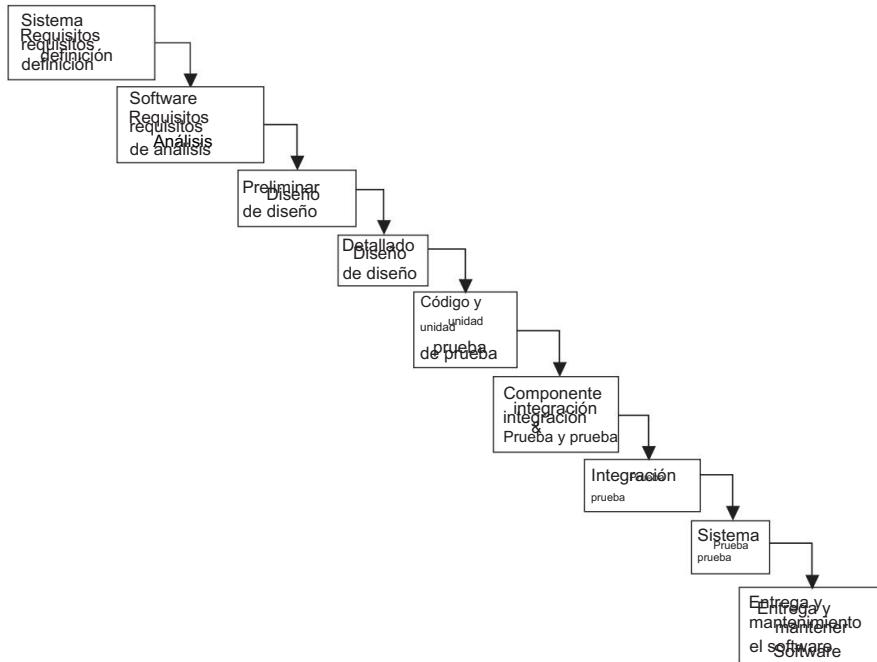


Figura 5.3 Ilustración del modelo de cascada.

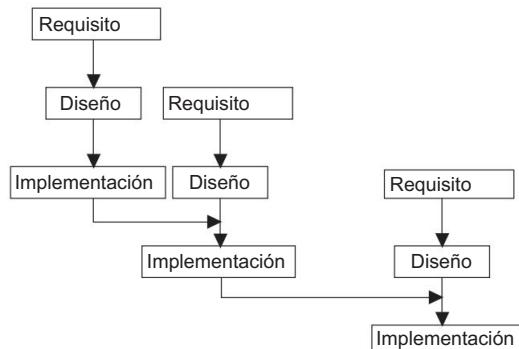


Figura 5.4 Ilustración del modelo de desarrollo incremental.

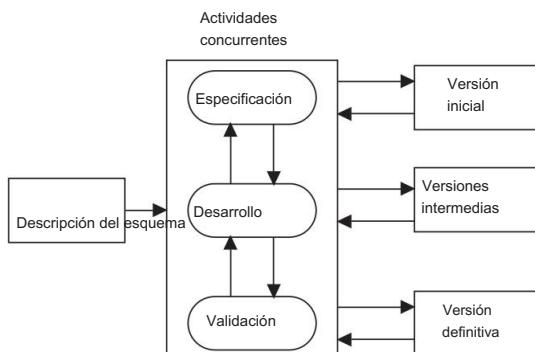


Figura 5.5 Ilustración del modelo evolutivo.

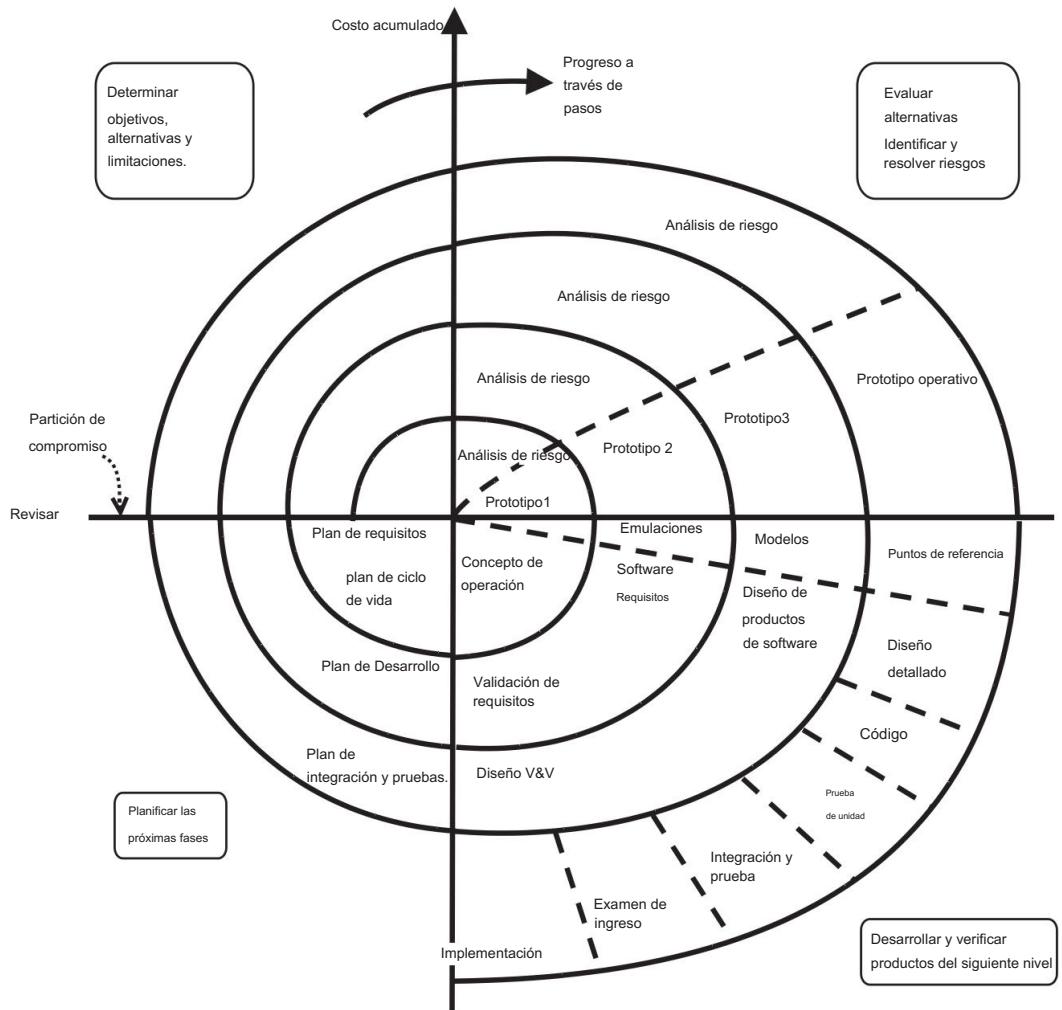


Figura 5.6 Ilustración del modelo en espiral. (Fuentes: [12, 13].)

Es complicado permitir la evolución cuando se intenta mantener una arquitectura sin cambios, mientras se le agregan componentes de forma incremental. Otro punto de vista es que todos los modelos citados son aplicables a algún aspecto de muchos proyectos; considérelos como vistas alternativas de la misma cosa, y cuantos más ángulos de visión se consideren, mejor será la comprensión. Hasta cierto punto, es como discutir si es mejor una vista superior o lateral.

10. Comience a adaptar el proceso de requisitos corporativos (u otros).

Mencioné anteriormente la importancia de utilizar un proceso de requisitos documentado. El punto esencial es que seguirás un proceso en cualquier caso, ya sea un proceso documentado o no. La experiencia nos ha demostrado que utilizar procesos documentados es un enfoque muy superior por las siguientes razones:

Se pueden incorporar al proceso las lecciones aprendidas de la experiencia industrial y corporativa, así como de proyectos anteriores en su organización.

Hay menos riesgo de tener que realizar retrabajos (el promedio de la industria para retrabajos en sistemas y proyectos de desarrollo de software es del 40% al 50% del esfuerzo y costo total del proyecto; reducir proactivamente el retrabajo es una oportunidad para ahorrar dinero, tiempo y esfuerzo). y aumentar la satisfacción del cliente y la calidad al mismo tiempo). Karl Wiegers, consultor en ingeniería de requisitos y experto en la industria, estima que el 80% del esfuerzo de retrabajo en un proyecto de desarrollo se debe a defectos de requisitos [14]. Esto sugiere fuertemente que invertir en el proceso de requisitos puede financiar niveles sustanciales de actividades de mejora de procesos.

Por ejemplo, Wiegers recomienda inspecciones formales de todos los documentos relacionados con los requisitos. Véanse los dos artículos de Wiegers [15, 16] sobre inspecciones de requisitos para obtener asesoramiento sobre cómo realizar inspecciones de requisitos.

Existe una mayor probabilidad de que pueda mejorar el proceso a medida que avanza en las actividades del proyecto si el proceso está documentado. "¡Vaya!" usted dice: "¡Ni siquiera sé a dónde ir para 'obtener' un proceso de requisitos!" ¡Si tu puedes! Vea mi sitio web (www.ralphyoung.net).

Por cierto, la adaptación es una habilidad crítica que se está perdiendo a medida que la base de experiencia se retira y es reemplazada por personas que son autodidactas o cuyos conocimientos se adquieren sólo a partir de folletos de marketing de software o estándares de la industria. La adaptación es la aplicación de conocimientos basados en la experiencia para llegar a una combinación inteligente de elementos de proceso estándar y desafíos situacionales. Recomiendo encarecidamente la revisión por pares del enfoque de adaptación siempre que sea posible.

11. Establecer un mecanismo para evolucionar los requisitos reales a partir de lo declarado. requisitos.

A medida que comience el proceso de identificación de requisitos más detallados, necesitará establecer un mecanismo para evolucionar los requisitos reales a partir de los requisitos establecidos (consulte el Capítulo 4 para aclarar las diferencias). En Prácticas efectivas de requisitos, me refiero a este mecanismo como el equipo conjunto (ver [2, pp. 46-53] para una discusión sobre el equipo conjunto). Realmente no importa cómo llames a este mecanismo. Lo que se necesita es uno o varios representantes del cliente que estén facultados para tomar decisiones relativas a los requisitos para reunirse con un número similar de personas facultadas del proyecto para revisar todos los requisitos y hacer lo siguiente:

Asegurar que cada requisito refleje las necesidades reales de los clientes y usuarios. Descubrirá que muchos de los requisitos establecidos no son requisitos reales.

Asegurar que cada requisito cumpla con los criterios de un buen requisito. Descubrirá que este paso requerirá mucho trabajo. Sabiendo que

Hay razones importantes para cada uno de los criterios, comprenderá que este trabajo es muy valioso y tiene un gran valor. Si usted no comprende las razones de uno o más de los criterios, tome algunas tiempo para estudiar los criterios y asegurarse de que cada criterio es básico.

Proporcione una justificación para cada requisito (por qué es necesario). Industria La experiencia es que hasta la mitad de los requisitos establecidos se pueden eliminar realizando este único paso. Párate a pensar por un minuto en el trabajo que este paso potencialmente puede salvar el proyecto ("proactivamente reducir el retrabajo"). Tenga en cuenta también que este esfuerzo bien podría marcar la diferencia entre el éxito y el fracaso del proyecto. Discuta esto con su gerente y el primer ministro. Ayuda al proyecto a beneficiarse de un TRABAJO EN EQUIPO enfoque aplicando enfoques de equipo en sus propias actividades laborales (ver Capítulo 9 para una discusión sobre el trabajo en equipo).

Mientras realiza este trabajo, concéntrese en los beneficios del producto (requisitos reales necesarios), no en las características de los productos del trabajo. Podemos proporcionar un montón de funciones, todo lo cual requiere tiempo y dinero para desarrollarse e incluirse en el sistema, pero debemos centrarnos en los requisitos mínimos (recuerde: "¡Cualquier cosa más es demasiado!" [Neal Whitten]).

Hacer un esfuerzo intencional para descubrir requisitos reales que sean no declarado. Al trabajar con sus clientes y usuarios, trabaje deliberadamente para identificar requisitos reales que no han incluido. Esta voluntad requieren una comprensión profunda de las necesidades del cliente, atención a los requisitos a nivel del sistema y un análisis cuidadoso. Nosotros nunca Prometió que el trabajo de la RA sería simple o fácil. De una sola mano abordar esto es la evolución: ver una versión temprana o un prototipo o La demostración es una poderosa ayuda para descubrir requisitos reales no declarados. Los modelos y maquetas son formas económicas de obtener requisitos no declarados. D. Leffingwell y D. Widrig [17] y Ellen Gottesdiener [5] ofrece otras técnicas sencillas y económicas.

Este es un buen momento para enfatizar la importancia de la verificación (¿la ¿La solución de diseño cumple con los requisitos identificados?) y la validación (¿son los requisitos implementados en el sistema entregado?) (V&V). Ver paso 17 y el Capítulo 7 para mayor discusión y aclaración.

Aquí hay algunos otros objetivos de los buenos profesionales de requisitos, como se identifican: cumplida por Ivy Hooks en su capacitación de requisitos:

Identificar suposiciones incorrectas;

Garantizar la coherencia;

Incrementar el cumplimiento;

Reducir los malentendidos entre organizaciones e individuos;

Mejorar la capacidad de respuesta de los proveedores;

Mejorar la satisfacción de todos los clientes;

Escriba buenos requisitos.

Vale la pena señalar que muchas partes interesadas sólo podrán comunicar los requisitos en términos generales. Luego, el equipo deberá "traducirlos" en requisitos de diseño utilizables. Mi sugerencia es ser tolerante al principio del proceso para estimular la comunicación abierta y luego reforzar las expectativas a medida que el tiempo y el conocimiento del proceso en desarrollo lo permitan.

12. Proporcionar sesiones de capacitación relacionadas con los requisitos para los participantes del proyecto, incluidos clientes y usuarios, y para los RA.

Aquí tienes un desafío. Es muy probable que sus compañeros del proyecto se resistan a esta formación. Afirman que no lo necesitan y que están muy ocupados con su propio trabajo. En realidad, ninguna de estas afirmaciones es válida, dadas las prioridades y necesidades generales del proyecto, porque es importante que todas las partes interesadas comprendan el valor de la inversión en el proceso de requisitos y la información relacionada. Una lección importante que todos hemos experimentado, pero que desafortunadamente aún no hemos aprendido, es que el trabajo técnico se inicia antes de que se identifiquen los requisitos reales, lo que resulta en una gran cantidad de retrabajo con sus costos asociados, cronograma, calidad y problemas de satisfacción del cliente. Todas las partes interesadas del proyecto necesitan el beneficio de comprender la experiencia de la industria en relación con los requisitos. Consulte la Tabla 5.4 para ver los temas sugeridos. Visite mi sitio web para ver un ejemplo de "Resumen de requisitos iniciales del proyecto" [18].

Tabla 5.4 Temas sugeridos para la sesión informativa inicial sobre los requisitos del proyecto

Cuestiones de la industria en ingeniería de requisitos;
El valor de invertir más en el proceso de requisitos;
El proceso de requisitos del proyecto y/o de la organización;
Descripción general de los mecanismos, métodos, técnicas y herramientas que se utilizarán;
Tipos de requisitos;
Requisitos de reunión;
Funciones de la AR;
Criterios de un buen requisito;
Tipos de errores de requisitos y cómo se pueden reducir;
Por qué y cómo todos los miembros del equipo del proyecto deben reconocer los requisitos nuevos y modificados durante el desarrollo y comunicarlo al RA y al PM;
Apoyo a la gestión de una ingeniería de requisitos eficaz;
Medios para reducir el retrabajo en el proyecto:
Identificar las necesidades reales;
Controlar los cambios en los requisitos y la adición de nuevos requisitos;
Utilizar revisiones por pares;
Proporcionar inspecciones de todos los documentos relacionados con los requisitos.

Otro aspecto importante es que los AR que apoyan el proyecto deben recibir la capacitación adecuada. Consulte la Tabla 5.5 para obtener una lista de temas recomendados. Por ejemplo, es importante que todos los RA estén de acuerdo sobre cómo redactar buenos requisitos. Deben estar familiarizados con los tipos de requisitos (ver Capítulo 4) y los esfuerzos a realizar para reducir los errores de requisitos. Particularmente importante es el enfoque que se utilizará para reducir la repetición del trabajo en el proyecto.

Cuando la cultura se resiste vigorosamente a la capacitación, a menudo es posible lograr el mismo objetivo a través de sesiones de trabajo que aplican los métodos que se desean comunicar. Un RA experimentado compartió su éxito al informar a la gerencia sobre la capacitación que se brindaría a los empleados: ¡creía que era el equipo gerencial el que más necesitaba esa información!

13. Vuelva a escribir los requisitos de software o sistema de alto nivel a medida que avanza en el pasos iniciales.

Es posible que ya exista una declaración de las necesidades, expectativas y requisitos de alto nivel del cliente, al menos en forma preliminar, en el contexto histórico.

Cuadro 5.5 Temas sugeridos para la capacitación de los AR

La importancia de los requisitos para el éxito del proyecto, basada en la experiencia de la industria;

El valor de los buenos requisitos;

Roles, habilidades y características de un AR eficaz;

Tener y utilizar un proceso de requisitos;

El valor de invertir más en el proceso de requisitos (8% a 14% de los costos totales del proyecto);

El proceso de requisitos del proyecto;

Descripción general de los mecanismos, métodos, técnicas y herramientas que se utilizarán:

Tipos de requisitos;

El repositorio de requisitos (y sus numerosos componentes);

Recopilación de requisitos: las técnicas que se utilizarán;

Escribir buenos requisitos:

Asegurar que cada requisito cumpla con los criterios de un buen requisito; Documentar la

justificación de cada requisito; Priorizar los requisitos: no todos

los requisitos son iguales; No inventar requisitos de forma independiente;

No tomar decisiones sobre requisitos; No chapado en

oro.

Tipos de errores de requisitos y cómo se pueden reducir; Utilizar la

herramienta de requisitos automatizada del proyecto;

Garantizar que los requisitos identificados se utilicen y no se conviertan en "artículos de estantería"; V&V

en el proyecto; Apoyo a

la gestión de una ingeniería de requisitos eficaz; Reducir el retrabajo en el

proyecto:

Identificar las necesidades reales; Controlar

los cambios en los requisitos y la adición de nuevos requisitos; Utilizar revisiones por pares.

información que ya has digerido. Por alto nivel me refiero a declaraciones amplias que describen las capacidades necesarias que buscan los clientes. (En un proyecto que apoyé, el cliente proporcionó una especificación de requisitos demasiado detallada al comienzo de nuestro contrato; el cliente había invertido varios años-persona de esfuerzo durante un período de dos años para desarrollar la especificación. Cuando sugerimos volver a visitar los requisitos de alto nivel, el cliente indicó que no hicíramos esto, sino que procediéramos con el esfuerzo de desarrollo utilizando la especificación detallada. Un año después, después de una inversión de 40 años-persona de esfuerzo, el cliente decidió rehacer el requisitos de nivel.) Comience a reescribir los requisitos de alto nivel. Este conjunto debe constar de 50 a 200 requisitos, dependiendo del alcance del sistema. Centrarse en los requisitos de alto nivel tiene importantes ventajas:

Le ayudará a comprender el alcance del sistema.

Evitará que se sienta abrumado por información más detallada.
requisitos.

Le ayudará a obtener información sobre los requisitos reales.

Administre las expectativas culturales para evitar que su primer borrador se convierta en el documento “construido”.

T. Korson [19] enfatiza la importancia de los niveles de abstracción en la recopilación de requisitos y cree que existen principios críticos de recopilación de requisitos que deben observarse. (También cree que un enfoque de requisitos impulsado por casos de uso a menudo resulta en una falla en la identificación de los requisitos reales). La Tabla 5.6 muestra cómo existen niveles de abstracción asociados tanto con los tipos de requisitos como con las actividades de desarrollo.

Los siguientes principios de recopilación de requisitos amplían esto:

Comience con los requisitos del sistema de alto nivel y trabaje en niveles más detallados (tenga en cuenta que esto es diferente de la descomposición funcional, que implica descomponer una función particular en el sistema).

Mantenga los requisitos comerciales separados de las especificaciones de la interfaz. Korson cree que esta es la forma en que la mayoría de los equipos OO entran

Tabla 5.6 El uso de niveles de abstracción ayuda a gestionar el proceso de requisitos

Tipos de requisitos (Ver Capítulo 4)	Actividad de desarrollo
Negocio	Análisis y modelado de dominio.
Interfaces	Análisis y modelado de aplicaciones.
Diseño	Desarrollo de arquitectura
Especificaciones detalladas	Desarrollo de código fuente

dificultad: los requisitos de alto nivel se incluyen en especificaciones detalladas que impiden a los clientes considerar alternativas y que engañan a los diseñadores.

Facilitar a los usuarios un conocimiento más profundo de sus necesidades reales y requisitos.

No derivar el diseño de casos de uso. Korson cree que los casos de uso deberían detenerse en el límite de la interfaz del sistema. La experiencia de otros es que los límites del sistema se establecen según el punto de vista y los casos de uso a menudo también pueden ser excelentes ayudas analíticas a nivel de subsistema. Deberá determinar sus propios puntos de vista según su experiencia.

14. Iniciar el desarrollo de los requisitos reales en base a los requisitos planteados.

Evolucionar y priorizar [20] los requisitos reales. Recordará esta sugerencia del Capítulo 1. No deje de hacerlo. La “lista de requisitos reales” es un producto de trabajo que evolucionará durante un período de varias semanas o meses. Asegúrese de utilizar el control de versiones y el control de cambios, de modo que siempre sepa con precisión cuál es realmente el producto de trabajo actual y exactamente cómo debe leerse. Si no está familiarizado con estas técnicas de CM, consulte la discusión sobre CM en el Capítulo 7. Además, busque a alguien que tenga experiencia con CM y aprenda de él o ella. No hay presión, pero tenga en cuenta que todo el proyecto depende totalmente de una declaración exhaustiva, precisa y actual de los requisitos reales. Sin él, el proyecto está fuera de control y en peligro. Es posible que se encuentre tratando de ponerse al día con diseñadores que avanzaron mucho en el ciclo de vida mientras se generaban los requisitos. Este es un problema muy común y una excelente motivación para que la RA haga llegar los documentos guía a la comunidad lo antes posible.

15. Iniciar la documentación de la justificación de cada requisito.

Este es otro producto del trabajo de requisitos que evolucionará durante un período de semanas o meses. Asegúrese de rastrear la fuente de cualquier justificación aceptada. Este es un componente importante del repositorio de requisitos, que se analiza a continuación. Los estudios comerciales bien realizados no sólo proporcionan una justificación sólida sobre el camino elegido, sino que también contienen la lógica mediante la cual se hizo la elección y los supuestos que se utilizaron. Esto hace que sea mucho más fácil defender la decisión o revisarla si las suposiciones cambian o se demuestra que son falsas.

16. Establecer un mecanismo para controlar los cambios de requisitos y nuevos requisitos.

La segunda contribución más importante que puede hacer al proyecto (después de establecer el equipo conjunto o mecanismo similar para desarrollar los requisitos reales) es establecer un mecanismo para controlar los cambios en los requisitos y

nuevos requisitos. La experiencia de la industria verifica que sin tal mecanismo, la mayoría de los proyectos pronto se saldrán de control y correrán el riesgo de una alta probabilidad de fallar. Nos referimos a los cambios en los requisitos como "volatilidad de los requisitos".

La experiencia de la industria es que los proyectos que exceden el 2% de volatilidad de los requisitos incurren en costos y riesgos de cronograma. Se recomienda un objetivo de volatilidad de requisitos del 0,5% por mes. "¡Vaya! Dices: "¡Eso no es mucho!" Precisamente.

Ésa es exactamente la razón por la que se necesita un mecanismo para controlar los cambios en los requisitos y la adición de nuevos requisitos.

Un mecanismo excelente que podría considerar para esto es el equipo conjunto. Tiene algunos miembros capacitados que pueden hablar en nombre del cliente y del proyecto. Con suerte, los miembros se han conocido durante el proceso de evolución de los requisitos reales y están trabajando juntos como un equipo de alto rendimiento. De lo contrario, sugiera un taller de equipo conjunto para considerar las características de un equipo de alto desempeño (discutido en la sección titulada "Trabajo en equipo" en el Capítulo 8). Seleccione las características que los miembros de su equipo conjunto quieren utilizar. Esto debería ayudar a crear un ambiente de TRABAJO EN EQUIPO.

A menudo vemos proyectos con una volatilidad de requisitos superior al 24% anual. Esto sugiere que no debemos esperar éxito.

Deberíamos identificar proactivamente formas de mitigar los riesgos asociados con requisitos nuevos y modificados; Por ejemplo:

Utilice un enfoque de desarrollo incremental.

Proporcionar lanzamientos o versiones posteriores de productos de trabajo.

Incrementar el presupuesto y cronograma, reconociendo que existe una relación geométrica entre cambios y costo/cronograma. Reconozca también que existen riesgos para la satisfacción del cliente y la calidad.

La compañía de Capers Jones, Software Productivity Research (SPR) (www.spr.com), ha documentado que las tasas de defectos asociadas con las nuevas funciones agregadas durante el desarrollo son aproximadamente un 50% mayores que las de los artefactos asociados con los requisitos originales [21]. Los niveles de eficiencia de eliminación de defectos también se reducen, a veces en más del 15%. Esta combinación significa que un porcentaje muy significativo de los defectos entregados pueden ser

6. Las AR deberían involucrarse en las actividades de gestión de riesgos del proyecto. Visite el sitio web de la revista Software Quality Management (www.sqmmagazine.com). Suscribir. Revisar artículos de interés y adquirir competencia en la identificación, análisis, evaluación, planificación, gestión, mitigación y seguimiento y control de riesgos. Convírtase en miembro del equipo de gestión de riesgos de su proyecto. Como señaló el experto en riesgos de la industria David C. Hall, "A pesar del creciente consenso sobre el valor de la gestión de riesgos, la implementación efectiva de procesos de gestión de riesgos en organizaciones y proyectos está lejos de ser común". (Consulte www.sqmmagazine.com/issues/2002-04/mature.html.) En mi experiencia, incluso cuando existe un proceso de riesgo en un proyecto y se realiza una gestión de riesgos, las actividades de gestión de riesgos generalmente no son exhaustivas. Las actividades del proyecto relacionadas con la gestión de riesgos suelen ser una oportunidad para la mejora continua y estas actividades pueden tener un impacto potencialmente enorme en el éxito o el fracaso del proyecto. Rich Raphael es el propietario del proceso de riesgo de TI DES de Northrop Grumman y puede brindar asesoramiento experto. Contáctelo en RRaphael@ngc.com.

se remonta a las crecientes necesidades de los usuarios. En otras palabras, al controlar los cambios en los requisitos, la calidad de los productos aumenta significativamente y los costos se reducen. Para minimizar el daño causado por requisitos tardíos, se recomiendan encarecidamente procedimientos formales de gestión de cambios y herramientas de control de configuración de última generación. Las inspecciones formales de diseño y códigos también son útiles.⁷

Otra idea valiosa de la experiencia de Ivy Hooks en la NASA es que un cambio de un tercio en los requisitos da como resultado una duplicación del costo del sistema desarrollado. Esto ayuda a explicar la ira que sienten nuestros clientes cuando los costos de los sistemas prometidos aumentan. Los clientes y usuarios necesitan de nuestra ayuda para entender que los constantes cambios que solicitan tienen un gran impacto en los costos. y el horario. Si podemos ayudarles a entender esto y si podemos ganar su apoyo en el control de cambios y nuevos requerimientos, tendremos se asociaron para lograr un resultado más exitoso del proyecto. Una comprensión acordada del sistema es fundamental para el éxito de un proyecto.

Los cambios de requisitos debidos a interpretaciones, cambios de alcance u otros factores aleatorios deben suprimirse en la mayor medida posible. Los cambios en los requisitos que resultan del conocimiento emergente deben ser adoptados y administrado. Los altos niveles de volatilidad de los requisitos sugieren que el subyacente El problema que está resolviendo el sistema o el software aún no se comprende completamente. En ese caso, avanzar más en el ciclo de diseño bien podría ser la raíz causa del problema, no la inestabilidad de los requisitos en sí. Uno RA experimentado opinó que muchas empresas se mienten a sí mismas sobre el diseño madurez únicamente para lograr métricas internas!

17. Realizar la planificación V&V.

Uno de los criterios de un buen requisito es que sea verificable, que es, que el código que proporciona esta capacidad se puede probar para determinar que el requisito se cumple efectivamente en el sistema entregado. El enfoque que se utilizará para lograr esto debe proporcionarse tempranamente como un aspecto el esfuerzo de desarrollo de requisitos y debe identificarse como un atributo de cada requisito en la herramienta de requisitos automatizada. Ver el Comentarios y referencias sobre V&V en el Capítulo 7 y el libro de Grady. sobre este tema [22]. Hooks y Farry [23] alientan a que se aborde V&V durante el desarrollo de requisitos, porque este enfoque mejora la calidad de los requisitos, garantiza que los requisitos respalden la verificación, proporciona una base para estimar el costo y el cronograma de verificación, y brinda oportunidades para controlar el costo y el riesgo. Identifican palabras que señalan requisitos no verificables y sugieren posibles sustitutos que usted pueda utilizar.

7. Consulte la discusión sobre los crecientes requisitos de los usuarios y la calidad del software en Jones's Software Quality: Analysis and Directrices para el éxito [21, págs. 134-137]. Tenga en cuenta que realmente se han aprendido muchas lecciones, pero no se han aplicado. Los principales problemas son que (1) nosotros, los profesionales, no leemos lo suficiente, no estudiamos lecciones, no tomamos medidas para aplicar las lecciones para nuestros procesos y procedimientos, e implementar prácticas mejoradas; y (2) la administración se contenta con permitirnos salir del paso y no mejorar continuamente. En el contexto de Watts Humphrey, "no practicamos lo que predicamos". Consulte www.sei.cmu.edu/publications/articles/practice-preach/practice-preach.html.

usar. Proporcionan ejemplos de requisitos verificables y no verificables y proporcionan una lista de verificación que resume las preguntas relacionadas con la verificación que deben formularse. Grady está de acuerdo: "Exigir que la misma persona que escribe un requisito de producto también escriba el requisito del proceso de verificación correspondiente aproximadamente en el mismo período de tiempo reduce el riesgo del programa más que cualquier otra actividad" (mensaje de correo electrónico personal, 2 de abril de 2000). . Este es un buen momento para reflexionar sobre los comentarios proporcionados en el paso 11 sobre la evolución de los requisitos reales. Existe mucha confusión en nuestra industria acerca de la diferencia entre validación y verificación (consulte la aclaración de estos dos términos en el Capítulo 7, tema 13).

18. Seleccionar las prácticas, métodos y técnicas que se utilizarán para recopilar los requisitos.

Hay muchas prácticas, métodos y técnicas disponibles. Algunas son más útiles y efectivas que otras. Su proyecto disfrutará de un alto retorno de la inversión si se toma algún tiempo para seleccionar los que se van a utilizar. Primero, digiera las discusiones relacionadas sobre ellos en Prácticas efectivas de requisitos [2]. Al hacer esto, obtendrá muchos conocimientos de la experiencia en la industria. Asegúrese de que los miembros de su equipo de proyecto hayan tenido experiencias exitosas con todos los métodos y técnicas seleccionados. Un proyecto, a menos que sea de investigación y desarrollo (I+D), no es momento para probar un nuevo método o técnica. Su proyecto debe seleccionar sólo métodos y técnicas que sean conocidos, familiares para los desarrolladores y probados. (Una excepción es que usted quiera poner a prueba una práctica nueva o prometedora para determinar su aplicabilidad a su proyecto. Al utilizar un mentor experimentado, la prueba piloto podría convertirse en una demostración de prueba de aplicabilidad).

Amplíe su análisis para considerar las mejores prácticas del proyecto, en función de los riesgos identificados para ese proyecto. Mantenga a su gerente y al primer ministro involucrados e informados sobre estas deliberaciones. Todos los involucrados en un proyecto particular deben utilizar un proceso común, un conjunto de prácticas y mecanismos, técnicas y métodos, y herramientas automatizadas. Mantenga discusiones con el equipo del proyecto, seleccione y acuerde un conjunto común y brinde capacitación formal según sea necesario para garantizar que las personas que se espera que los utilicen estén empoderadas. Es contraproducente que los individuos salgan corriendo solos, haciendo lo suyo, no en coordinación con el resto del proyecto. equipo.

Es posible que su proyecto requiera métodos que no le resulten familiares. Esto sugiere que puede ser necesaria una planificación para la capacitación o que el proyecto puede necesitar contratar a un consultor que haya tenido éxito en el uso de un método, técnica o herramienta desconocidos. Proceder con cautela y mitigar los riesgos.

19. Comenzar la consideración y selección de una herramienta de requisitos automatizada, la identificación de los atributos que serán necesarios para cada requisito y la composición del repositorio de requisitos.

Estas tres tareas son esenciales y se espera que la AR proporcione liderazgo y asuma la responsabilidad de su desempeño exitoso.

Hay menos de una docena de herramientas de requisitos automatizadas sólidas en la industria que brindan la funcionalidad que necesitan proyectos de diversos tamaños (consulte la Tabla 5.7).

Todos los que han trabajado con sistemas y software durante algún período de tiempo han tenido experiencias con herramientas de requisitos automatizados (probablemente buenas y malas); todos tendrán sus propios prejuicios y opiniones sobre herramientas específicas.

Aléjese de todos estos consejos y comentarios, junte sus habilidades analíticas y hágase las siguientes preguntas:

¿Qué es lo que estamos tratando de hacer?

¿Qué nivel de sofisticación se requiere?

¿Cuánta capacitación formal en herramientas se necesita y cuánto podemos costear?

¿Qué productos de trabajo necesitamos de la herramienta?

¿Cómo ingresaremos datos (por ejemplo, los requisitos reales) en la herramienta? ¿La herramienta proporciona un enfoque de aportes que respaldará las necesidades del proyecto?

¿Quién en el proyecto tiene experiencia práctica con qué herramientas? ¿Puede esa persona estar disponible para ayudar con las actividades iniciales relacionadas con las herramientas?

¿Quiénes serán los usuarios principales, secundarios y de respaldo de las herramientas?

¿Cuántas licencias ("puestos") necesitamos?

Tabla 5.7 Requisitos comerciales, herramientas, proveedores y sitios web

Herramienta	Proveedor	Sitio web
Sistema de equipo estrella Requisitos (anteriormente Calibre RM)	Starbase, Inc., Santa Ana, California	www.borland.com/caliber
CUIDADO 3.0	Grupo SOPHIST, Núremberg, Alemania	www.sophist.de/sophist.nsf/zStartEngl!OpenFrameSet
NÚCLEO 4.0	Corporación VITECH, Viena, Virginia	www.vtcorp.com
PUERTAS 6.0	Telelogic, Malmo, Suecia y Irvine, California	www.Telelogic.com/products
Requisito racional Pro (ReqPro)	Corporación IBM, llanuras blancas, Nueva York	www.rational.com/products
Requisitos RTM y Trazabilidad Gestión	Chipware integrado, Inc., Reston Virginia	www.chipware.com
PIZARRA	Northrop Grumman Electronic Systems, Baltimore, Maryland CMD Corporation,	http://slate.md.essd.northgrum.com
SinergiaRM Enlace vital	Addison, Texas www.cmdcorp.com Compliance Automation, Inc., Boerne, Texas	www.complianceautomation.com
Requisitos de Xtie-RT Trazador	Ingeniería Teledyne Brown, Los Angeles, California	www.tbe.com/products/xtie/xtie.asp

¿ Cómo se comunicarán los clientes y usuarios con el equipo del proyecto en relación con la información relacionada con los requisitos, incluidas las solicitudes de cambios en los requisitos, nuevos requisitos y planes, ideas y sugerencias sobre versiones y lanzamientos posteriores de productos de trabajo?

Piense en otras preguntas que deben hacerse, según su entorno. La decisión de seleccionar una herramienta de requisitos automatizada particular debe basarse en un análisis exhaustivo y objetivo. Esta decisión es importante y no debe tomarse rápidamente ni basarse en opiniones sesgadas o desinformadas. Se debe redactar un estudio comercial (en la terminología CMMI⁸, el proceso de estudio comercial se conoce como resolución de análisis de decisiones [DAR], un nombre actualizado del análisis de soluciones candidatas [ACS] que se utilizó en SE-CMM⁹ [24]). Revise un ejemplo de estudio comercial de herramientas de requisitos automatizados en mi sitio web [25] y reutilícelo como plantilla para su análisis. Consulte el Inserto 5.2 para obtener información importante basada en la experiencia.⁸ Consulte también el artículo de Wiegers, "Automatización de la gestión de requisitos" [26], que compara cuatro herramientas de requisitos automatizadas: DOORS, RTM Workshop, Calibre RM (ahora "Requisitos del sistema Star Team") y Requisito profesional.

Como se enfatizó anteriormente, la selección de la herramienta de requisitos automatizados específica para respaldar su proyecto es una decisión importante. Involucre a su gerente y al primer ministro lo antes posible. Conozco muchos proyectos que intentaron funcionar sin una herramienta de requisitos automatizada. Esta es una receta para el desastre para un proyecto de cualquier tamaño. Los proyectos pequeños (de dos a tres personas durante tres a seis meses) pueden salirse con la suya utilizando MS Word o Excel como herramienta automatizada de requisitos. Un proyecto de mayor tamaño requiere una de las herramientas enumeradas en la Figura 5.3. Soy consciente de muchos proyectos en los que la complejidad de la herramienta en comparación con las necesidades de la misma se convirtió en una de las razones por las que el proyecto perdió el control. Como RA, asegúrese de cumplir adecuadamente su función a este respecto. No permita que puntos de vista sesgados y opiniones desinformadas impulsen esta decisión.

Iniciar la identificación de los atributos que se necesitarán para cada requisito inmediatamente después de la asignación al proyecto como (o el) RA. Un atributo es una característica de un requisito que es útil para ordenar, clasificar y gestionar requisitos. Esta es una de esas listas que debe tener a mano y en la que debe trabajar constantemente, porque a medida que trabaja, identificará los atributos que son necesarios. Es probable que termine con una lista de 20 a 40 atributos cuando esté listo para cargar su herramienta de requisitos automatizada. Da esta consideración.

Además, comience a pensar inmediatamente después de su asignación a un proyecto como RA en la composición del repositorio de requisitos del proyecto. Mucha gente piensa en esto como una base de datos automatizada (en la herramienta de requisitos).

En mi opinión, este es un enfoque mucho menos que adecuado. Piense en el repositorio de requisitos como una combinación de lo siguiente:

8. Nuestro agradecimiento al ingeniero de procesos y calidad Earl Hoovler por compartir sus experiencias y conocimientos.

Inserto 5.2—Lecciones aprendidas al realizar estudios comerciales

Seguir un proceso definido en un proyecto de desarrollo de software o sistema suele ser algo muy bueno. Sin embargo, siempre es necesario tener una mente abierta al seguir un proceso: estar alerta a los cambios en lo "real". Requisitos del proceso. Sea flexible y esté preparado para adaptar el proceso según sus necesidades. Usted sigue adelante para que pueda cumplir con esos requisitos. Este estudio de caso La recopilación de algunas lecciones aprendidas al realizar evaluaciones de alternativas proporciona ideas importantes.

Mientras trabajaba en una propuesta para un importante programa de implementación de software de recursos humanos COTS, un analista senior fue responsable de revisar las herramientas CM automatizadas y hacer una recomendación. Aunque él siguió En el proceso de estudio comercial estándar de la organización, después de presentar su recomendación inicial para una herramienta, encontró información de que la herramienta no funcionaría en el entorno al que serviría. Se dio cuenta de que él Necesitaba volver a la mesa de dibujo y revisar su recomendación inicial para reflejar los requisitos del mundo real. Las siguientes lecciones aprendidas desde su experiencia le puede ayudar en la realización de evaluaciones de COTS productos, independientemente del tipo de herramienta automatizada necesaria.

1. Tenga siempre en cuenta las preferencias de sus clientes, pero no haga recomendaciones que simplemente atiendan a esas preferencias, cuando el comercio El análisis del estudio no lo respalda. Cualquier cliente merece una revisión equilibrada de las opciones según los criterios seleccionados. El El propósito de un estudio comercial es explorar los hechos y hacer recomendaciones basadas en esos hechos. Es bueno ser consciente de una preferencias del cliente, pero no debe ser la principal ni la única tenga en cuenta su recomendación final, a menos que los hechos respalden la preferencia del cliente. Esto es muy difícil de lograr, pero es un objetivo que vale la pena. En mi experiencia de estudio comercial, el cliente ya había invertido en un conjunto de herramientas y quería una recomendación que respaldó esa compra. Necesitábamos Evite la tentación de aprobar la elección del cliente y intentó demostrar exactamente por qué su herramienta preferida no funcionaba trabajar en su entorno de desarrollo. Fuimos capaces de mostrar que si seleccionáramos su herramienta preferida, el proyecto sería experimentar más esfuerzo y costo y crear la necesidad de en su mayoría soluciones manuales (no un enfoque automatizado). Finalmente, pudimos convencer al cliente de que su herramienta preferida no era aceptable porque no podía lograr lo necesario Funciones de identificación y control.

2. Si utiliza criterios de selección previamente definidos para su estudio comercial, tenga en cuenta Asegúrese de adaptar los criterios para incluir factores y circunstancias apropiados. En nuestra organización, nuestras Pymes CM previamente reunidas las funciones y otros criterios que les gustaría ver respaldados a través de herramientas automatizadas. La mayoría de los equipos probablemente harán una lluvia de ideas

Inserto 5.2—Lecciones aprendidas al realizar estudios comerciales (continuación)

y enumerar sus propios criterios. Si están utilizando un conjunto de criterios que no crearon ellos mismos, el equipo de estudio comercial debe revisar los criterios y adaptarlos para eliminar funciones que sean

no son necesarios, incluyen requisitos o funciones adicionales y reflejan el entorno de trabajo esperado del equipo. Criterios más específicos proporcionarán los criterios de búsqueda básicos para las herramientas y ahorrarán tiempo al realizar selecciones iniciales para su revisión.

3. Hable continuamente con los clientes y equipos de ingeniería que utilizarán la herramienta e involúcrelos en el proceso de selección. Mantener a los clientes y a los equipos involucrados en todas las etapas del proceso de selección es la clave del éxito. Asegúrese de que se incluyan en demostraciones, revisiones de criterios y revisiones por pares de los resultados preliminares de los estudios comerciales. En mi ejemplo, no incluí especialistas en aplicaciones en la primera iteración del estudio comercial y, por lo tanto, recomendé una herramienta CM más convencional, no una que aborde las necesidades especializadas del entorno de desarrollo. Tampoco conocía las preferencias de herramientas del cliente. En mi segundo esfuerzo, utilicé la experiencia de ingenieros de desarrollo de aplicaciones que estaban familiarizados con los problemas de este entorno único. Trabajamos juntos para encontrar una solución que sirviera mejor al cliente y a los desarrolladores.
4. Siempre que sea posible, obtenga una copia de demostración de las herramientas que está revisando y pruébelas en su entorno. Con demasiada frecuencia, los proveedores no comprenden bien sus criterios ni los requisitos de sus clientes. A menudo traen una demostración predefinida de su herramienta con capacidades limitadas que no está configurada para funcionar en su entorno. También trabajan para convencerle de que las capacidades que demuestran funcionarán en su entorno, pero no pueden mostrarle exactamente cómo funcionarán. Esto puede parecer el mejor uso de su tiempo y supone una carga para el proveedor para demostrar que puede desempeñarse, pero a menudo obliga a tomar decisiones basadas en un sentimiento, en lugar de en un hecho. El mejor remedio para esta situación es solicitar al proveedor que le proporcione una versión por tiempo limitado de su herramienta actual, pedirle ayuda para configurar la herramienta y revisar el rendimiento de la herramienta según sus criterios. Por supuesto, deberá incluir suficiente tiempo en su plan de estudio comercial para realizar estas demostraciones en la planta. Esto proporcionará a los gerentes y clientes los datos que necesitan para tomar sus decisiones.
5. Asegúrese de que este sea un proceso iterativo. Generalmente ocurre que cuando se realiza un estudio comercial, no toda la información que necesita para tomar decisiones está disponible. Cuando hechos adicionales (p. ej., requisitos, capacidades de herramientas) superficie, debe estar

Inserto 5.2—Lecciones aprendidas al realizar estudios comerciales (continuación)

preparado para volver a la mesa de dibujo, revisar sus criterios, actualizar su análisis y posiblemente cambiar su recomendación. Por supuesto, el tiempo es un factor importante para hacer esto, pero al menos debe mencionar estos cambios a su cliente cuando haga su recomendación y dejar que el cliente decida si estos factores requieren otra revisión.

Earl Hoover

Ingeniero de Procesos

El documento de visión y alcance del proyecto;

El glosario del proyecto y la lista de acrónimos del proyecto;

La lista de requisitos a nivel del sistema;

Documentos fuente relacionados con los requisitos (cuando los requisitos vienen de);

Notas de reuniones y entrevistas con clientes y usuarios;

Notas de talleres de requisitos y documentación relacionada;

Cualquier documentación de los requisitos establecidos;

Actas de reuniones del equipo conjunto;

Listas de necesidades reales;

La base de datos en su herramienta de requisitos automatizada, incluido el RTM;

Cualquier producto de trabajo relacionado con requisitos, como requisitos especificaciones;

Descripciones de sistemas heredados (históricos) relacionados, complementadas con resúmenes de las capacidades proporcionadas que se necesitan en la nueva capacidad o sistema;

Limitaciones conocidas de la capacidad planificada, incluida la funcionalidad que no se puede o no se proporcionará;

Una descripción/visión de la trayectoria de crecimiento desde el lanzamiento inicial hasta el final sistema mate a través de un conjunto de lanzamientos o versiones escalonadas;

Definición de excepciones a la situación normal y error apropiado condiciones;

Otros.

20. Seleccionar y adquirir la herramienta de requisitos automatizados.

Suena fácil, ¿no? En ocasiones, la RA puede obtener acceso a la herramienta de requisitos automatizados seleccionada desde otro proyecto o una empresa.

entorno de software de ingeniería (ESE). 9 Sin embargo, la mayoría de las veces, el proyecto necesitará utilizar el proceso de adquisiciones de la organización. Dependiendo de situación, esto puede ser rápido y fácil, pero a menudo este paso resulta ser muy laborioso y complejo. No permita que su proyecto se vea comprometido debido a la disponibilidad tardía de los requisitos automatizados apropiados. herramienta. Comenzar temprano. Explique la importancia a su gente de adquisiciones. Facilite que formen parte de su equipo de requisitos. Haga un seguimiento agresivo. Involucre a su gerente, al PM y a la alta dirección de la organización. para garantizar que obtenga el apoyo que necesita para que las cosas sucedan dentro limitaciones de tiempo razonables.

21. Cargue los requisitos reales iniciales en la herramienta de requisitos seleccionada, etiquete cada uno requisito de forma única e iniciar la asignación de información de atributos apropiada para cada requisito.

Mencionamos esta tarea anteriormente en este capítulo. Quizás descubras que esta tarea resulta ser mucho más trabajo de lo que anticipa. Algunas de las razones para estos son los siguientes:

Los documentos que proporcionan requisitos de origen no siempre se cargan automáticamente o, si lo hacen, no toda la información se captura correctamente o fácilmente.

Habrá una curva de aprendizaje asociada con cualquier herramienta automatizada. Descubrirá que en realidad tiene capacidades de las que no se da cuenta inicialmente, y es posible que tengas que volver a trabajar.

Deberá decidir el sistema de numeración único o el enfoque para ser usado.

Deberá pensar en la trazabilidad bidireccional para cada requisito. ment y cómo esto debería lograrse.

Se debe identificar e incluir en la base de datos la fuente, historia, prioridad, estado, autor, asignación y trazabilidad de cada requisito. Muchos otros atributos de cada requisito deberán ser

Seguimiento también. Por ejemplo, hay dos tipos de atributos en PUERTAS, atributos definidos por el usuario y atributos definidos por el sistema. Los atributos definidos por el usuario se pueden crear a partir de tipos de atributos específicos, como texto, número entero, fecha, etc., y los usuarios crean instancias para su propias necesidades. Los atributos definidos por el sistema, sin embargo, están predefinidos por DOORS y registra automáticamente información esencial y muy útil en segundo plano. Los atributos le permiten asociar información con grupos de requisitos individuales o relacionados y a menudo facilitan

9. Algunas organizaciones patrocinan la disponibilidad de un conjunto de herramientas automatizadas que pueden ser utilizadas por los administradores de la organización. proyectos. En Northrop Grumman IT DES, llamamos a esta biblioteca de herramientas disponibles para prestar el software de ingeniería. medio ambiente o ESE.

Análisis de datos de requisitos mediante filtrado y clasificación basados en valores de atributos. Los atributos definidos por el sistema también se pueden utilizar para filtrar y clasificación. Aunque, en su mayor parte, son de sólo lectura y no modificables por el usuario, realizan la recopilación de información esencial y automática.

Consulte la Tabla 5.8 para ver ejemplos de atributos que quizás desee incluir en un matriz de atributos.

Evite la tentación de utilizar más atributos de los realmente necesarios para la tarea a mano. Algunos atributos bien elegidos que realmente se ingresan y gestionados son mucho más útiles que docenas que están mal ejecutadas o fácilmente confundido.

El inserto 5.3 proporciona algunas ideas sobre el uso de los atributos del sistema basadas en información proporcionada por Pete Carroll, anteriormente de Telelogic.

Tabla 5.8 Ejemplo de matriz de atributos de requisitos

Atributo	Requisito A	Requisito B, etc.
Identificación única	NFAK028	
Requisito texto	El tiempo necesario para que el equipo calentar antes de la operación no deberá exceder un (1) minuto desde un arranque en frío a -20 grados C.	
Fuente	Especificación de requisitos del sistema	
Dueño	Carlos Smith	
Razón fundamental	Verificado	
Prioridad	Medio	
Estado	Aprobado	
Costo	Bajo	
Dificultad	Medio	
Estabilidad	Medio	
Asignado a Bob Jones		
Ubicación	NFA Anexo K sección C	
Autor	Rick Chardón	
Revisión	1.7	
Fecha	10 de enero de 2003	
Razón	Traza marcada hasta NFASSS289.	
Rastreado desde NFARD125		
Remontado a	NFASSS289	
Etiqueta raíz#	208	
Historia	Requisito original	
Verificación	En Diseño xyz	
Validación	Sistema entregado	
Liberar	1.0	
Módulo	LDAP_Autenticar	
Etc.		

Inserto 5.3—Algunas ideas sobre el uso de los atributos del sistema en DOORS

Los atributos en DOORS permiten a los usuarios asociar datos con objetos, tablas marcadores, celdas de tablas, módulos y proyectos. Hay dos tipos de atributos, definidos por el usuario y definidos por el sistema. Usuario definido Los atributos pueden crearse a partir de tipos de atributos específicos, como texto, entero, fecha y similares, y los usuarios pueden crear instancias para sus propias necesidades. Los atributos definidos por el sistema, sin embargo, están predefinidos por DOORS y registra automáticamente una gran cantidad de información esencial y muy útil en segundo plano.

¿Alguna vez te has preguntado cómo hacer uso de estos sistemas definidos? atributos proporcionados automáticamente por DOORS? Aprovechar estos “para Los atributos gratuitos y listos para usar pueden facilitar su trabajo en DOORS. y más productivo.

Los atributos le permiten asociar información con individuos o grupos de requisitos relacionados y, a menudo, facilitan el análisis de los datos de requisitos mediante el filtrado y la clasificación en función de los valores de los atributos.

Los atributos definidos por el sistema también se pueden utilizar para filtrar y ordenar, y mientras son en su mayor parte de sólo lectura y no modificables por el usuario, realizan una recopilación de información esencial y automática para nosotros. Los atributos del sistema de sólo lectura, que existen automáticamente en todos los objetos y módulos, incluyen lo siguiente:

Nombre	Nombre del módulo formal, descriptivo o de enlace
Creado por	Nombre del usuario que creó el objeto
Creado en	Fecha de creación del objeto
Creado a través de	La forma en que se creó un objeto. (copia, entrada manual, extracción)

Última modificación por Nombre del usuario que modificó el objeto por última vez

Última modificación el objeto Fecha en que se modificó por última vez

Número absoluto Número único asignado al crear el objeto

Mapeo de enlaces Reglas para la vinculación de objetos (uno a uno, muchos a uno, etc.)

Algunos atributos definidos por el sistema no son de sólo lectura y permiten alguna modificación por parte del usuario. Objeto y módulo modificables por el usuario. Los atributos definidos por el sistema de nivel incluyen lo siguiente:

Título de objeto	El título del objeto.
Texto de objeto	El texto real del objeto.
Texto breve del objeto	Una breve indicación textual de la objeto utilizado para la visualización gráfica
Descripción	Nombre descriptivo completo del módulo.

Inserto 5.3—Algunas ideas sobre el uso de los atributos del sistema en DOORS
 (continuado)

Prefijo	Prefijo opcional para lo absoluto número utilizado para garantizar la singularidad identificación de objetos
---------	--

Todos los atributos definidos por el sistema anteriores, modificables o no, registran información automáticamente y pueden usarse tal como los definidos por el usuario. atributos para mostrar datos esenciales para la gestión de sus requisitos. Utilice atributos definidos por el sistema siempre que necesite mostrar información sobre quién, qué, dónde y cuándo, así como información crucial información sobre modificaciones de requisitos.

En cualquier módulo DOORS, inserte una columna y seleccione de la lista Atributo de visualización el atributo Creado por definido por el sistema para mostrar quien creó un requisito. Inserte una segunda columna y use el atributo Última modificación definido por el sistema para mostrar quién hizo la última cambiar a un requisito. Inserte una tercera columna utilizando el atributo Última modificación definido por el sistema para mostrar cuándo se realizó el último cambio. se hizo. Ahora tienes una vista que refleja la creación original.

¡e información de cambio esencial sobre sus requisitos! Mejor todavía, seleccione el Asistente de impacto/rastreo en el menú Herramientas-Impacto/rastreo, y en la segunda ventana seleccione mostrar cualquier número de las opciones disponibles de atributos de enlace y objetos definidos por el sistema. Este asistente, completo con atributos definidos por el sistema disponibles para su selección, Crea rápida y fácilmente vistas de análisis de impacto y seguimiento completas con información esencial, como la creación de objetos (Creado por, Creado el), fechas de cambios (Última modificación el) y otra información útil, como el nombre del módulo de enlace en el que se registran los enlaces (Módulo de enlace Nombre).

En resumen, los atributos definidos por el sistema proporcionan soluciones a algunos Necesidades de seguimiento de información. Utilice atributos definidos por el sistema para su ventaja para simplificar los requisitos y la gestión de cambios.

Pete Carroll, ex miembro de Telelogic.

Ian Alexander advierte que, si bien pueden ser necesarios entre 20 y 40 atributos, a menudo es posible arreglárselas con menos. Ha visto a gerentes técnicos de alto rendimiento identificar 20 o más atributos, no verificar la necesidad de y crear mucho trabajo innecesario. Así como somos responsables de Para identificar los requisitos reales, debemos asegurarnos de que las actividades laborales que recomendamos y utilizamos sean realmente necesarias. Una buena planificación y un análisis cuidadoso de “Por qué” las actividades laborales son “requeridas” son componentes importantes de la RA. role.

22. Realizar levantamiento de requisitos.

"¡Vaya!" Usted dice: "ya estamos en el paso 22 de este capítulo y recién ahora vamos a hablar sobre cómo reunir los requisitos". Buen punto. En realidad, hemos estado hablando de pasos relacionados a lo largo de este capítulo. Cada uno de los pasos anteriores es parte del proceso de recopilación de requisitos.

Le sugiero que descargue el artículo "Prácticas recomendadas de recopilación de requisitos" [27], del sitio web de CrossTalk (haga clic en la edición de abril de 2002 sobre requisitos riesgosos). Allí, brindo consejos detallados y discusiones relacionadas con este paso, recomiendo técnicas específicas y sugiero varias referencias apropiadas que lo ayudarán. (No sienta que tiene que leer y digerir cada referencia; simplemente obtenga suficiente información para comenzar a trabajar de manera efectiva en su tarea actual. Siempre puede regresar y leer más o más. En otras palabras, intente utilizar referencias para informar., motivarte e inspirarte, en lugar de permitir que te creen un sentimiento de frustración por tu falta de conocimiento o que sean una barrera que te frene).

Entre las más de 40 técnicas de obtención de requisitos disponibles,
Las técnicas más efectivas son las siguientes:

Entrevistas;

Análisis de documentos;

Lluvia de ideas;

Talleres de requisitos (una versión moderna de JAD);

Creación de prototipos;

Casos de uso (cuando se usan correctamente);

Guiones gráficos;

Análisis de interfaces;

Modelado;

Análisis de desempeño y capacidad;

Escenarios.

Entre los muchos libros que incluyen tratamientos extensos de los requisitos, obtención de comentarios, asegúrese de tener lo siguiente en su biblioteca personal:

Alexander y Stevens, Redactando mejores requisitos

Buede, El diseño de ingeniería de sistemas: modelos y métodos.

Carr, et al., Asociación en la construcción: una guía práctica para el éxito del proyecto

Cockburn [pronunciado co-burn], Redacción de casos de uso eficaces

Gottesdiener, Requisitos por colaboración: talleres para definir necesidades

- Grady, Análisis de requisitos del sistema
- Grady, Validación y Verificación de Sistemas
- Harmon, Watson, Comprensión de UML: la guía para desarrolladores
- Hooks, Farry, productos centrados en el cliente: creación de productos exitosos a través de la gestión inteligente de requisitos
- Humphrey, Introducción al proceso de software personal
- Kotonya y Sommerville, Ingeniería de requisitos: procesos y Técnicas
- Leffingwell y Widrig, Gestión de requisitos de software
- McConnell, Guía de supervivencia de proyectos de software
- Sommerville y Sawyer, Ingeniería de requisitos: una buena práctica Guía
- Wiegers, revisiones por pares en software
- Wiegers, Requisitos de software, 2.^a ed.
- Wiley, Requisitos esenciales del sistema: una guía práctica para la gestión basada en eventos. Métodos
- Madera y Plata, Desarrollo Conjunto de Aplicaciones
- Prácticas de requisitos jóvenes y eficaces

Le recomiendo encarecidamente que desarrolle un conocimiento práctico de todos estos libros. No es suficiente tener un conocimiento superficial de algunos procesos, prácticas, métodos, técnicas y herramientas relacionados con los requisitos.

Para ser eficaz en su función, la AR debe poder recomendar un enfoque bueno y probado y facilitar su despliegue, implementación e institucionalización.

Consulte la página web de Vitech (www.vtcorp.com) para obtener información sobre la herramienta de requisitos CORE y una versión que se puede descargar para su evaluación. Esta herramienta proporciona capacidades de modelado de comportamiento. The Engineering Design of Systems: Models and Methods [28] de Buede describe el uso de esta herramienta para modelar y proporciona problemas de ejemplo para ayudarlo a familiarizarse con el uso de la herramienta. Rational Rose y BPwin son otras herramientas que proporcionan modelos de comportamiento. A veces, los diagramas son extremadamente útiles para transmitir los requisitos esenciales del sistema, para ver cómo un sistema encaja en sistemas más grandes en su entorno y para comprender cómo encajan entre sí los componentes de un sistema.

Una de las técnicas mencionadas anteriormente es un escenario. Los escenarios operativos se mencionaron al principio de este capítulo como parte integral del enfoque de ingeniería de sistemas para desarrollar conceptos operativos. Se ha desarrollado un método llamado Análisis de Necesidades del Usuario Basado en Escenarios (SUNA) para facilitar la clarificación y el refinamiento de las necesidades de los usuarios. El inserto 5.4 proporciona una descripción de este método.

Finalmente, lea "Una forma rápida y precisa de determinar las necesidades del cliente" de Cristina Afors y Marilyn Zuckerman [29]. Los autores de este artículo creen que los clientes tienden a decir una cosa durante la obtención de requisitos y luego hacen algo completamente diferente. Recomendan una tecnología llamada análisis de huellas que tiene en cuenta las emociones humanas. Creen que el análisis de huellas puede en realidad predecir el comportamiento humano.

23. Involucrar a arquitectos y diseñadores de sistemas en las revisiones de los requisitos.

Una de las 10 prácticas de requisitos efectivas que recomiendo en mi libro anterior es "Iterar los requisitos y la arquitectura repetidamente". La experiencia de la industria ha demostrado que este es un consejo valioso. Anteriormente recomendé realizar tres o cuatro iteraciones del desarrollo de requisitos (consulte la experiencia de Ellen Gottesdiener analizada en el paso 6). En varias situaciones en mi carrera, he tenido la oportunidad de trabajar con un arquitecto o diseñador de sistemas para iterar los requisitos reales y la arquitectura o diseño previsto para el sistema o software planificado. Mi experiencia es que cuando dedicamos tiempo y esfuerzo a hacer esto, desarrollamos mejores requisitos y una arquitectura más sólida. La razón es que los requisitos y la arquitectura dependen unos de otros. Cuando iteramos uno con el otro, ambos se vuelven mejores, más fuertes, más robustos y flexibles y son más capaces de adaptarse a cambios futuros y nuevas tecnologías (porque tenemos una mejor comprensión). Piensa sobre esto. Tiene sentido y es una de las mejores prácticas comprobadas de la industria.

Invite a los arquitectos y diseñadores de sistemas involucrados en su proyecto a revisar los requisitos reales. Comparta con ellos la información proporcionada en Prácticas efectivas de requisitos [2, págs. 134-135]. Considérelos parte del equipo de requisitos del proyecto y facilite que obtengan esta misma perspectiva.

Recuerda, dos cabezas piensan mejor que una (siempre).

24. Desarrollar la estrategia de trazabilidad a utilizar.

Mencionamos este paso anteriormente en este capítulo. Resulta que la trazabilidad bidireccional (desde las necesidades y requisitos reales del cliente y usuario hasta los productos, y viceversa) de los requisitos es (1) crítica para el éxito del proyecto, y (2) una tarea compleja y difícil que requiere una experiencia considerable para desempeñarse bien. La trazabilidad de los requisitos es la capacidad (1) de correlacionar la necesidad del cliente con el requisito; (2) rastrear (identificar y rastrear) la instanciación de cada requisito en todos los productos de trabajo, desde la especificación de requisitos hasta el diseño y el desarrollo de componentes del sistema, a través de pruebas y documentación del sistema; y (3) asignar un requisito principal a un requisito secundario, y viceversa. Esta capacidad es absolutamente crítica para todos los sistemas. Una pauta clave aquí es ser coherente: utilizar el mismo tipo de rastreos para todos sus documentos originales. Un RTM automatizado en la herramienta de requisitos automatizados es el mecanismo que debe utilizarse para proporcionar esta trazabilidad. Deberías digerir a James D.

Consulte el artículo de Palmer "Trazabilidad" [30] para fortalecer su comprensión y conocimiento sobre esta importante habilidad de AR. Ver también la discusión

Inserto 5.4—Uso de escenarios en la recopilación de requisitos

Técnicas bien conocidas como análisis de documentos, entrevistas, observaciones y talleres son buenas formas de recopilar requisitos, particularmente cuando el trabajo es ampliar o evolucionar un sistema existente.

Sin embargo, en ciertos tipos de proyectos, puede ser valioso tomar una enfoque completamente diferente. Cuando existe una amplia interacción humana, un número limitado de limitaciones heredadas, el alcance y/o el negocio

Los requisitos no están claros o existe un mandato para la innovación, la redacción colaborativa de escenarios puede ser una herramienta útil. Utilizado al inicio de la fase de recopilación de requisitos y, junto con otras técnicas de recopilación, puede proporcionar una buena base para un éxito.

desarrollo.

El término escenario se usa ampliamente, particularmente en el mundo del diseño y desarrollo de sistemas, donde puede significar cualquier cosa, desde un conjunto de guiones gráficos a descripciones textuales estructuradas de las interacciones del usuario con un sistema informático. En este contexto, los escenarios a los que nos referimos son historias: narrativas amplias, descriptivas, del día a día que son pretende involucrar al lector en la comprensión de cómo el nuevo desarrollo, producto o servicio se integrará y afectará las vidas de diferentes partes interesadas. La narrativa rica es una herramienta poderosa porque la El acto de crearlo nos obliga a visualizar situaciones y la lógica de las interacciones con todo detalle. En el contexto de la visualización de sistemas de software, Organiza nuestros pensamientos y habilita las necesidades humanas y empresariales y posibles soluciones técnicas que se considerarán simultáneamente. en un entorno grupal, ayuda a las personas a pensar “fuera de la caja” (es decir, más allá cualquier limitación preconcebida) y sintetizar su inspiración, conocimiento y experiencia en un todo lógico. Las historias resultantes transmitir la visión de una manera que pueda ser fácilmente entendida por todos los partes involucradas en el proyecto.

Sin embargo, el pensamiento creativo desenfrenado corre el riesgo de generar confusión y ampliar alcance y costos en espiral. Por eso es importante garantizar que los beneficios del proceso de visualización se puede aprovechar para generar resultados rentables.

Soluciones de software innovadoras. La Universidad de Essex en Estados Unidos Kingdom ha seguido desarrollando SUNA, un proceso concebido originalmente por British Telecommunications (BT) en 1998 que proporciona un conjunto de pasos lógicos para este propósito. Es una guía no prescriptiva que respalda a un equipo combinado de clientes, usuarios, RA, diseñadores y desarrolladores.

pasando por la generación de escenarios hasta un proceso de destilación y, finalmente, integración con la recopilación de requisitos estándar de la industria y métodos de desarrollo de software. En relación con estos métodos, el propósito de SUNA es refinar la forma del desarrollo propuesto generado por los requisitos del negocio y para proporcionar un punto de partida para la Actividad más detallada de recopilación de requisitos reales. Por lo tanto, se superpone y alimenta la actividad de captura de requisitos o, alternativamente, alimenta directamente un ciclo de desarrollo iterativo.

Inserto 5.4—Uso de escenarios en la recopilación de requisitos (continuación)

A continuación se presenta un resumen muy breve del proceso:

1. Establecer un equipo de personas que representen los diferentes aspectos de el proyecto.
2. Recopilar, analizar y difundir al equipo la documentación que se relaciona con el alcance y definición del proyecto y las investigaciones relevantes (técnicas, sociales y de mercado). Esto proporciona el punto de partida para los escenarios.
3. Convocar al equipo a un taller con el objetivo de hacer lo siguiente: Identificar a todas las partes interesadas. Seleccione tres o cuatro partes interesadas grupos en los que basar los escenarios.
Escriba varios escenarios desde diferentes perspectivas de las partes interesadas. Generalmente se trata de viñetas de 1.000 palabras sobre cómo el desarrollo propuesto encaja en las vidas de las partes interesadas y de los usuarios y en las interacciones que tienen lugar. Intenta pensar más allá de las limitaciones.
Analizar los escenarios y extraer las necesidades comunes de los usuarios (es decir, los requisitos). Compílelos en una lista numerada de forma única que formará la base de una especificación de requisitos.
4. Después del taller, organice las necesidades del usuario en una jerarquía utilizando descripciones breves y resumidas (lo que se conoce como jerarquía de necesidades). Esto proporciona una estructura lógica inicial para el desarrollo propuesto y crea una valiosa herramienta para la toma de decisiones al permitir que todos los elementos del desarrollo se vean en una sola página (o al menos en una pequeña cantidad de páginas).
5. Comprobar la cordura de los escenarios con personas que tengan un nivel adecuado de conocimientos, pero que estén fuera del equipo del proyecto.
6. Evaluar en qué medida las partes interesadas valoran las diferentes necesidades enviando un cuestionario a un grupo selecto. Coteje los resultados y reflejelos en la jerarquía de necesidades, por ejemplo, resaltando las necesidades en diferentes colores que reflejen las categorías de valor alto, medio y bajo.
7. Llevar toda esta información a un segundo taller de “scooping” con los siguientes objetivos:
Presentar la jerarquía de necesidades y utilizarla para facilitar la decisión sobre el alcance del desarrollo en función de las restricciones presupuestarias, la viabilidad técnica y el valor para las partes interesadas. Marque las decisiones en la jerarquía.
Documentar la justificación de cada requisito e indicar qué necesidades están dentro del alcance y cuáles fuera del alcance en comparación con la lista numerada de necesidades del usuario.

Inserto 5.4—Uso de escenarios en la recopilación de requisitos (continuación)

Recopilar los resultados y pasar a la siguiente etapa de requisitos.

recopilación de elementos.

Los productos finales clave son (1) una serie de escenarios que pueden ser utilizado para ilustrar la visión de desarrollo, (2) una lista numerada y anotada de necesidades y requisitos de los usuarios, (3) una jerarquía de necesidades que muestra tanto la visión amplia como el alcance del desarrollo propuesto, y (4) un documento que contenga los detalles de la toma de decisiones.

razón fundamental.

SUNA fue desarrollado por primera vez en 1998 por BT como subproducto de un proyecto paneuropeo que examinó cómo aprendemos y a quién aprendemos. de. El resultado del proyecto fue un innovador servicio basado en web. prototipo que influyó en el diseño de productos de aprendizaje establecidos. Positivo Las relaciones, creadas en ese momento, impulsaron posteriores colaboraciones.

SUNA ahora se ha utilizado ampliamente en pequeñas colaboraciones. proyectos de investigación en entornos principalmente comerciales y algunos académicos. Se ha aplicado en campos que van desde el aprendizaje y la educación hasta los aspectos técnicos de la gestión de múltiples dispositivos y, en general, Se ha descubierto que ayuda a promover la claridad, la innovación y las buenas relaciones. No es probable que SUNA sea relevante para grandes sistemas complejos y los beneficios del enfoque pueden verse restringidos cuando Hay una interacción humana limitada y numerosas limitaciones fijas, pero si hay flexibilidad y deseo de innovación, vale la pena considerarlo. Para obtener más información, consulte www.essex.ac.uk/chimera/consultancy.html.

Nota: El análisis de necesidades del usuario basado en escenarios (SUNA) fue desarrollado por primera vez en 1998 por van Helvert y Fowler mientras trabajaban para Las telecomunicaciones británicas como subproducto de un proyecto paneuropeo observando cómo aprende la gente. Usado con permiso.

Joy van Helvert y Chris Fowler (2003)
Quimera—Instituto de Investigación e Innovación Sociotécnica

La Universidad de Essex
Ipswich, Suffolk

en Prácticas efectivas de requisitos [2, págs. 208–210], que incluye “Definiciones y Directrices para la trazabilidad de requisitos”. Digest también Leffingwell y Widrig, Gestión de requisitos de software [17, págs. 333–346 y capítulos 32 y 33 sobre V&V].

25. Identificar los requisitos que se cumplirán en la primera versión o productos iniciales. (priorizar necesidades reales).

Implícito en esta declaración está el concepto de tener más de una versión, reconociendo que rara vez, o nunca, podemos cumplir con todos los requisitos reales en una sola versión. Si ha trabajado bien y en colaboración con sus clientes y usuarios, ellos habrán llegado a comprender que el sistema planificado no puede serlo todo para todas las partes interesadas, probablemente nunca, pero ciertamente no en su entrega inicial, instalación y conversión de servicios relacionados, y bases de datos necesarias, implementación, operación, documentación y esfuerzos de capacitación (por favor, comprenda de esta lista que la entrega o rotación del sistema es un conjunto de actividades muy complejo; el proyecto puede descarrilarse fácilmente porque existen muchos riesgos, muchos de los cuales no están bajo el control del proyecto). Trabaje duro con sus clientes y usuarios para evolucionar e identificar los requisitos reales que se cumplirán en la primera versión o en los productos iniciales. Una línea base de requisitos es el conjunto de requisitos asociados con una versión particular de un producto o sistema. Involucre a su gerente, el PM, los clientes y los usuarios en el desarrollo de un enfoque de priorización de requisitos que tenga una alta probabilidad de éxito. Mire detenidamente para identificar, evaluar y mitigar los riesgos (con suerte, todavía es miembro del equipo de gestión de riesgos del proyecto y el proceso de gestión de riesgos está bien y activo).

Un RA experimentado señaló que, además de ser la forma correcta de abordar los requisitos y el proceso de diseño, priorizar los requisitos también tiene la gran ventaja de permitir que el RA acepte aportes cuestionables de personas políticamente poderosas y los desactive asignando el requisito a una publicación posterior. . Muy a menudo, el individuo o el requisito cuestionable desaparecen antes de que se produzca un daño real, y la RA vive para luchar un día más.

26. Establecer un enfoque para una prueba de concepto, prototipo u otra aproximación de producto de trabajo.

Este paso (y muchos otros) entra en la categoría de sentido común. A las personas a menudo les resulta imposible decir lo que quieren hasta que ven algo tangible a lo que puedan reaccionar. Para cambios menores de diseño, la versión anterior del producto puede realizar esta función bastante bien. Cuando el diseño es fundamentalmente nuevo, los prototipos son esenciales para comprender realmente los requisitos. Si se omite la creación de prototipos en el proceso de diseño, la primera versión se convierte en realidad en el prototipo y todas sus deficiencias se hacen visibles para el mercado. Todos somos conscientes de que lograr que los clientes y usuarios revisen prototipos y pruebas de concepto les permite identificar necesidades y problemas de manera temprana, antes de que los desarrolladores hayan desarrollado los productos de trabajo finales. Esto ahorra esfuerzo, tiempo y dinero en nuestro objetivo de que los clientes y usuarios acepten los productos de trabajo. La experiencia de la industria muestra que los prototipos son eficaces para reducir el aumento de los requisitos¹⁰ y pueden combinarse con otros efica-

10. La AR debe tener un conocimiento profundo de la proliferación de requisitos, la fuga de requisitos, las fuentes de requisitos no oficiales y las formas de controlar estos problemas. Estudie el Capítulo 10 en Prácticas efectivas de requisitos [2].

métodos, como talleres de requisitos y JAD [31]. Los prototipos por sí solos pueden reducir el aumento de los requisitos entre un 10% y un 25% (lea costos y cronogramas reducidos). La creación de prototipos debe considerarse tanto una técnica de obtención como parte del ciclo de vida; es una forma especialmente buena de abordar y validar las necesidades tácitas y aclarar los requisitos reales.

27. Incorporar las mejores prácticas de requisitos y obtener apoyo de gestión para una ingeniería de requisitos eficaz (incluido un enfoque de calidad integrado).

¡Ahora hay una tarea difícil! Analizaremos las mejores prácticas de requisitos en el siguiente capítulo. Sin embargo, creo que este tema es tan importante que decidí escribir un libro completo sobre él (de hecho, dos libros). Estudie las prácticas efectivas de requisitos [2], prestando especial atención al Capítulo 11. Discuta el uso de las mejores prácticas que considere apropiadas para su situación en su entorno con su gerente y el PM. Tomar acción. No te reprimas. Descubrirá que ha prestado un valioso servicio a su proyecto. Tenga en cuenta que es mucho más fácil iniciar una acción que seguirla y garantizar que, de hecho, la mejor práctica se haya implementado e implementado efectivamente en un proyecto y se haya institucionalizado en toda la organización. El despliegue, la implementación efectiva y la institucionalización de cualquier práctica son un desafío. Es necesario convencer a los demás de que vale la pena dedicar tiempo y esfuerzo a realizar las prácticas. Siempre que sea posible, “gestionar por hechos”; es decir, recopilar datos para poder determinar si las cosas han mejorado y, de ser así, en qué medida. Gestionar por hechos (en lugar de hacerlo por simpleza o por intuición) es un hábito valioso y útil que se debe desarrollar.

Estudie el Capítulo 8 de este volumen. Un revisor del borrador del índice de este libro me advirtió que quizás el tema de la calidad esté fuera del alcance de este libro. Luché con esta retroalimentación y finalmente llegué a la conclusión de que la calidad es inseparable de un trabajo de requisitos efectivo. Lea el Capítulo 8 y espero que esté de acuerdo.

28. Completar la recopilación de requisitos para la primera versión.

Antes de completar los requisitos para la versión 1, asegúrese de que exista una base o fundamento válido para iniciar el trabajo técnico posterior, es decir, la programación, el desarrollo o la codificación. Tenga en cuenta que la experiencia de la industria es que una vez que se completan los requisitos, las actividades posteriores, como las inspecciones de diseño, inspecciones de códigos y pruebas, no son muy efectivas para eliminar los defectos de los requisitos. De hecho, una vez que los defectos importantes están incluidos en los requisitos, tienden a ser inmunes a la mayoría de las formas estándar de eliminación de defectos y son especialmente resistentes a la detección mediante pruebas. Estas lecciones abogan firmemente por una tesis importante de este libro: es necesario dedicar más tiempo y esfuerzo al proceso de requisitos y a identificar los requisitos reales. Le sugiero que se familiarice con los materiales escritos por Capers Jones para ser más consciente de los temas relacionados [32–38].

Por supuesto, el proceso de recopilación de requisitos continúa después de cada lanzamiento. Es valioso incorporar las lecciones que hemos aprendido hasta la fecha en

nuestras actividades posteriores. Con demasiada frecuencia, no dedicamos el tiempo y el esfuerzo a hacer esto. Esta es otra promulgación del PDCA: tómate el tiempo en varios kilómetros. piedras para armar el equipo y evaluar lo que funcionó y lo que podría haber sido se ha hecho mejor y cómo. Un equipo, un proyecto y una organización pueden aprender una enorme cantidad de información valiosa simplemente haciendo esas preguntas a sus propios miembros. La parte difícil es seguir adelante y tomar medidas sobre las sugerencias resultantes. Necesitará un mecanismo para realizar un seguimiento de esto: un plan de acción, una lista de elementos de acción que se actualice periódicamente, un calendario administrado por personas responsables, lo que sea que funcione para usted.

Resumen

En este capítulo, he sugerido utilizar una lista de verificación de 28 pasos que comprenden un procedimiento para recopilar requisitos. Esto puede parecer una gran cantidad de pasos, quizás adecuados sólo para un proyecto grande y maduro. En realidad, los proyectos de todos los tamaños y todos los niveles de madurez deberán abordar estos pasos. Todos los proyectos abordarán estos pasos ya sea de manera ordenada o al azar. He enfatizado que cuando el enfoque de recopilación de requisitos no es efectivo, se crea el escenario para un esfuerzo técnico desperdiciado durante las actividades de seguimiento del proyecto (léase: el resto del proyecto), creando la necesidad de retrabajo y poniendo en peligro el éxito del proyecto. Tengo otra sugerencia para usted: lea "Los hábitos de los analistas eficaces" de Wiegers [39]. En este artículo, otro experto de la industria¹¹ comparte su opinión. Creo que encontrará mucha correspondencia entre este capítulo y el consejo de Wiegers. Visite su sitio web y aproveche sus numerosas sugerencias, escritos, ideas, recetas y "obsequios". Wiegers, Ian Alexander, Ivy Hooks, Jeff Grady (y yo) brindamos capacitación y consultoría sobre requisitos eficaces.

Piense en términos generales. Se Flexible. Fomenta siempre el trabajo en equipo y la mejora continua en todo lo que haces. Revise el Capítulo 3 periódicamente (digamos durante un fin de semana) y piense en las cosas que debería hacer o hacer de manera diferente. Ser una influencia positiva para los demás y el proyecto. Pídale a su gerente comentarios sobre su desempeño. Hacer algo al respecto. Divertirse.

Caso de estudio

Una vez que se establecieron y acordaron los requisitos, el equipo del proyecto se comprometió con un cronograma de entrega y un método para controlar los requisitos. Si de repente surgía un nuevo requisito, por cualquier motivo, el cliente tenía que priorizarlo. Para hacer esto, los clientes exigieron saber el impacto que tendría (por ejemplo, semanas-persona de esfuerzo). Para proporcionar esta estimación, el líder del equipo clave y con más conocimientos tuvo que gastar

11. Fortalezca su hábito de utilizar materiales e ideas desarrollados por otros, incluidos expertos de la industria. Por ejemplo, Karl Wiegers, Ian Alexander, Ivy Hooks, Charles Markert, Tom Gilb, Jeff Grady y otros me han ayudado a crecer y aprender. Se han hecho amigos y sus enseñanzas ahora son parte de cómo hago mi trabajo diario.

tiempo con el cliente profundizando en los detalles de ese requisito en particular, desarrollando una serie de requisitos derivados. El ritmo al que llegaban nuevos requisitos comenzó a abrumar a las personas clave y se creó una acumulación de análisis de requisitos que siguió aumentando. A nuestro cliente le pareció que no podíamos evaluar un requisito simple; Según los RA claves, la presión aumentó hasta el punto de ebullición a medida que se acercaba la fecha de entrega programada. Lección aprendida: perdimos el control del proyecto porque no pudimos gestionar eficazmente las solicitudes de requisitos nuevos y modificados.

Referencias

- [1] Porter-Roth, B., *Solicitud de propuesta: una guía para el desarrollo eficaz de RFP*, Boston, MA: Addison-Wesley, 2002.
- [2] Young, RR, *Prácticas de requisitos eficaces*, Boston, MA: Addison-Wesley, 2001.
- [3] Alexander, I. y A. Farncombe, JBA, *Plantilla de análisis de partes interesadas*, Sistemas Curso Fundamentos de Ingeniería, 2003.
- [4] Sharp, H., et al., "Identificación de partes interesadas en el proceso de ingeniería de requisitos", IEEE (1999): 387–391.
- [5] Gottesdiener, E., *Requisitos por colaboración: talleres para definir necesidades*, Boston, MA: Addison-Wesley, 2002.
- [6] Wiegers, KE, *Requisitos de software*, 2.^a ed., Redmond, WA: Microsoft Press, 2003.
- [7] Weinberg, gerente general, "¡Simplemente diga no! Mejorando el proceso de requisitos", Programador estadounidense (10) (1995): 19–23.
- [8] Young, RR, *Plantilla de plan de requisitos y plan de requisitos de muestra*, en www.ralphyoung.net.
- [9] Wiegers, KE, *Revisiones por pares en software: una guía práctica*, Boston, MA: Addison-Wesley, 2002.
- [10] Waugh, P., "Materiales de capacitación para participantes en la revisión por pares y moderadores de la revisión por pares". Northrop Grumman IT DES, 2003. Contáctela en penny.waugh@ngc.com.
- [11] Sorensen, R., *Comparación de metodologías de desarrollo de software*, Centro de soporte de tecnología de software, enero de 1995, en www.stsc.hill.af.mil/crosstalk.
- [12] Boehm, B., "Spiral Model of Software Development and Enhancement", IEEE Computer (mayo de 1988) (también publicado en Barry Boehm, *Software Risk Management*, IEEE Computer Society Press, 1989, 26).
- [13] Boehm, B. y WJ Hanse, "The Spiral Model As a Tool for Evolutionary Acquisition", un esfuerzo conjunto del Centro de Ingeniería de Software de la Universidad del Sur de California y el SEI, CrossTalk (mayo de 2001) : 4–11 .
- [14] Wiegers, KE, "10 Requisitos Trampas a Evitar", Revista de Ingeniería de Calidad y Pruebas de Software (enero/febrero de 2000), en www.stqemagazine.com/featured.asp?id=8.
- [15] Wiegers, KE, "¿Funcionan sus inspecciones?" StickyMinds.com (24 de junio de 2002), en www.stickyminds.com.

- [16] Wiegers, KE, "Inspección de requisitos", StickyMinds.com (30 de julio de 2001), en www.stickyminds.com.
- [17] Leffingwell, D. y D. Widrig, Gestión de requisitos de software, Reading, MA: Addison-Wesley, 2000.
- [18] Young RR, "Resumen de requisitos iniciales del proyecto", en www.ralphyoung.net.
- [19] Korson, T., "El mal uso de los casos de uso: gestión de requisitos", en www.korson-mcgregor.com/publications/korson/Korson9803om.htm.
- [20] Wiegers, KE, "Lo primero es lo primero: priorizar los requisitos", Software Revista Desarrollo 7(9) (septiembre de 1999): 24–30.
- [21] Jones, C., Calidad del software: análisis y directrices para el éxito, Londres: Prensa internacional de computadoras Thomson, 1997.
- [22] Grady, JO, Validación y verificación del sistema, Boca Ratón: CRC Press, 1997.
- [23] Hooks, IF y KA Farry, Productos centrados en el cliente: creación de productos exitosos a través de la gestión inteligente de requisitos, Nueva York: AMACOM, 2001.
- [24] EPIC, A Systems Engineering Capability Maturity Model, Versión 1.1, noviembre de 1995. SEI, Carnegie-Mellon University, Pittsburgh, PA, en www.sei.cmu.edu/pub/documents/95.reports/pdf/mm003.95.pdf.
- [25] Young, RR, "Estudio sobre el comercio de herramientas y requisitos", en www.ralphyoung.net.
- [26] Wiegers, KE, "Automatización de la gestión de requisitos", Desarrollo de software (julio de 1999). Disponible en www.processimpact.com.
- [27] Young, RR, "Prácticas recomendadas de recopilación de requisitos", CrossTalk 15(4) (abril de 2002): 9–12, en www.stsc.hill.af.mil/crosstalk/2002/index.html.
- [28] Buede, DM, El diseño de ingeniería de sistemas: modelos y métodos, Nueva York: John Wiley & Sons, Inc., 2000.
- [29] Afors, C. y MZ Michaels, "Una forma rápida y precisa de determinar las necesidades del cliente", Sociedad Estadounidense para la Calidad, Progreso de la Calidad (julio de 2001): 82–87.
- [30] Palmer, JD, "Trazabilidad", Ingeniería de requisitos de software, RH Thayer y M. Dorfman, eds., Los Alamitos, CA: IEEE Computer Society Press, 1997, págs.
- [31] Wood, J. y D. Silver, Desarrollo conjunto de aplicaciones, Nueva York: John Wiley & Hijos, 1995.
- [32] Jones, C., Evaluación y control de riesgos de software, Englewood Cliffs, Nueva Jersey: Prentice Salón, 1994.
- [33] Jones, C., Estimación de costos de software, Nueva York: McGraw Hill, 1998.
- [34] Jones, C., "Factores positivos y negativos que influyen en la productividad del software", Burlington: MA, Software Productivity Research, Inc., Versión 2.0, 15 de octubre de 1998.
- [35] Jones, C., Evaluaciones de software, puntos de referencia y mejores prácticas, Reading, MA: Addison-Wesley, 2000.
- [36] Jones, C., "Calidad del software en 2000: qué funciona y qué no", Burlington, MA: Software Productivity Research Inc., 18 de enero de 2000.
- [37] Jones, C., "Software Project Management in the 21st Century", American Programmer 11(2) (febrero de 1998), en <http://spr.com/news/articles.htm>.

[38] Jones C., "Qué significa ser 'el mejor en su clase' en software", Burlington, MA,
Software Productivity Research, Inc., versión 5, 10 de febrero de 1998.

[39] Wiegers, KE, "Habits of Effective Analysts", Desarrollo de software (octubre de 2000), en
www.processimpact.com.

Machine Translated by Google

CAPÍTULO

6

Contenido

[Resumen](#)[Caso de estudio](#)[Referencias](#)

Mejores prácticas para requisitos Desarrollo y Gestión

En capítulos anteriores he sugerido que hagas ciertas cosas y no hagas otras. En este capítulo, comparto un conjunto de mejores prácticas para el desarrollo y la gestión de requisitos. La frase "Mejores prácticas" se utiliza con frecuencia en la ingeniería de sistemas y software (entre muchas otras profesiones). Escuchamos y leemos mucho sobre las mejores prácticas. Sin embargo, en realidad no dedicamos tiempo ni esfuerzo a evaluarlos, analizarlos, ponerlos a prueba, desplegarlos, implementarlos e institucionalizarlos. La razón es bastante sencilla: supone mucho trabajo. Requiere lo siguiente:

Desarrollar una comprensión profunda de la práctica;

Comunicar su valor a compañeros de trabajo y gerentes;

Obtener el compromiso de probar la práctica (probarla);

Proporcionar capacitación sobre la práctica para que otros la entiendan y lo que intentamos lograr mediante su uso;

Implementar la nueva práctica, cambiar lo que estamos haciendo ahora a hacer algo diferente;

Implementar la nueva práctica, asegurando que la nueva forma sea usado;

Mantener la nueva práctica, incluida la obtención de apoyo para su usar;

Evaluar su impacto, tal vez incluso diseñar una forma de medir los resultados de su uso;

Declarar la victoria, reconocer que el nuevo método es mejor que el antiguo, tal vez incluso celebrar el éxito;

Institucionalizar el uso de la nueva práctica en todo el proyecto u organización.

Visto desde esta perspectiva, es más fácil entender por qué las mejores prácticas Las políticas no se implementan ni institucionalizan.

La Tabla 6.1 proporciona una lista de mejores prácticas para el desarrollo de requisitos. y gestión.

Algunas de estas mejores prácticas se han discutido con cierta profundidad en otras partes de este libro, por lo que limitaré mis comentarios sobre ellas en este capítulo.

Mi objetivo es convencerte de que vale la pena el esfuerzo de al menos poner a prueba cada uno de estas mejores prácticas en su proyecto y haga un esfuerzo serio para evaluar los resultados del uso de cada mejor práctica.

La tabla 6.1 ha sido elaborada cuidadosamente y me gustaría explicar su estructura.

Primero, las actividades de requisitos en una tarea o proyecto están indisolublemente entrelazadas con las actividades de gestión de proyectos, así como con otras disciplinas.

como CM, ingeniería de sistemas y control de calidad. El enfoque de requisitos que se utiliza en una tarea o proyecto no se desarrolla ni se implementa en el vacío por la RA. Se desarrolla a través de un conjunto de decisiones que necesariamente implican otras personas clave, incluido el cliente, el PM, el ingeniero de sistemas y otros.

Le recomiendo que comparta la Tabla 6.1 con el equipo de liderazgo de la tarea o proyecto. (incluido el cliente) y seleccionar conjuntamente las mejores prácticas que ustedes, como equipo, determinen que deben utilizarse en su tarea o proyecto. Este artefacto está disponible para descargar en mi sitio web (www.ralphyoung.net).

En segundo lugar, las mejores prácticas recomendadas en la Tabla 6.1 se agrupan en tres categorías:

1. Desarrollo de requisitos;
2. Gestión de requisitos;
3. Gestión de proyectos.

Dentro de cada categoría, están organizadas aproximadamente de forma secuencial: primero se haría 1, luego 2, y así sucesivamente. Sin embargo, tenga en cuenta que muchos

¡Algunas de las mejores prácticas relacionadas con la gestión de proyectos son generales! Para

Por ejemplo, la mejor práctica 25 recomienda que una meta, propósito,

o se establezca la misión para la tarea o proyecto. Falta un acuerdo

meta, propósito o misión de la tarea o proyecto hará que sea difícil

lograr algo de valor. Todas las demás mejores prácticas en la categoría de gestión de proyectos abordan áreas que pueden estar más allá del alcance del

REAL ACADEMIA DE BELLAS ARTES. Son las mejores prácticas comprobadas de la industria que pueden tener un impacto positivo significativo en los esfuerzos de gestión y desarrollo de requisitos. Tú

Necesitarás el compromiso del equipo de liderazgo de la tarea o del proyecto para implementar estas prácticas de manera efectiva. No basta con que la RA permita

prácticas para evolucionar queremos o no. Su responsabilidad es proporcionar esta lista a su

equipo de liderazgo de la tarea o proyecto con la solicitud de que se seleccionen las prácticas a utilizar colaborativamente por el equipo.

En tercer lugar, la columna situada más a la derecha del cuadro 6.1 proporciona una referencia a la

capítulo de este libro donde se analizan las mejores prácticas para que pueda obtener

más información, orientación y referencias adicionales al respecto. Cada una de las

Las mejores prácticas recomendadas se analizarán en el orden en que aparecen en la lista.

Tabla 6.1.

Tabla 6.1 Mejores prácticas para el desarrollo y la gestión de requisitos

Mejores prácticas numéricas		Requisitos Desarrollo	RM	Proyecto Gestión	Capítulo Referencia
1	Desarrollar un plan de requisitos.	X	XX		15
2	Escribir requisitos que cumplan con los criterios de un buen requisito.	X			diecisésis
3	Identificar e involucrar a todos los stakeholders en la tarea o proyecto.	X			1, 5, 6
4	Asegurar que los objetivos de la tarea o proyecto han sido identificados, documentados, y acordado por las partes interesadas.	X			5
5	Utilice talleres de requisitos para lograr una visión compartida, facilitar el compromiso y obtener la aceptación de todas las partes interesadas.	X			5, 6
6	Proporcionar capacitación sobre requisitos para RA, para miembros del personal del proyecto, y para partes interesadas.	X		XX 5	
7	Identificar los requisitos reales. Colaborar con clientes y usuarios en relación con la requisitos establecidos para identificar el verdadero requisitos. Mira los requisitos desde múltiples puntos de vista.	X			2, 5, 6
8	Documento la justificación de cada requisito, es decir, por qué es necesario).	X			6
9	Utilice una recopilación de requisitos eficaz técnicas.	X			5
10	Involucrar a los clientes y usuarios en todo el esfuerzo de desarrollo.	X		XX 6	
11	No tome decisiones sobre requisitos.	X		XX	6
12	No dorar, es decir, añadir características o capacidades).		XX 6		
13	Utilice un glosario de proyectos y un proyecto lista de acrónimos.	X		XX 5	
14	Iterar los requisitos y arquitectura repetidamente para evolucionar mejor requisitos y un sistema más robusto arquitectura.	X			6
15	Utilice expertos en el dominio/PYME que sean conocedor y experimentado en el áreas funcionales abordadas por el esfuerzo técnico.	X			5, 6
descripción	Cuantificar el ROI para seleccionar los requisitos mecanismos, prácticas, métodos, técnicas y herramientas a utilizar.		XX 6		
17	Identificar los requisitos mínimos que satisfacer necesidades reales.		XX 6		
18	Priorice los requisitos de forma temprana y frecuente.	X		XX	6
19	Proporcionar inspecciones de todos. documentos relacionados con los requisitos.	X		X	5, 7
20	Limitar los cambios a los requisitos y al adición de nuevos requisitos consistentemente, con presupuesto adicional y cronograma hecho disponible por el cliente para completar el tarea, proyecto o sistema.			X	5, 6

Tabla 6.1 Mejores prácticas para el desarrollo y la gestión de requisitos (continuación)

Mejores prácticas numéricas		Requisitos Desarrollo	RM	Proyecto Gestión	Capítulo Referencia
21	Utilice versiones y lanzamientos de productos de trabajo. para adaptarse a nuevos requisitos, modificados requisitos y de menor prioridad requisitos.	X		XX 5	
22	Utilice un sistema automatizado de potencia industrial herramienta de requisitos. Proporcionar y utilizar atributos de los requisitos.	X		XX	5, 6
23	Desarrollar o adaptar y utilizar herramientas organizativas y políticas de requisitos del proyecto y una proceso de requisitos que es continuamente mejorado en su tarea, proyecto o organización. Invertir del 8% al 14% del total costos del proyecto en el (ciclo de vida del sistema) proceso de requisitos.	X		XX	5, 6
24	Utilice requisitos probados y familiares mecanismos, enfoques, prácticas, métodos, técnicas y herramientas.	X		XX	5, 6
25	Establecer una meta, propósito o objetivo acordado. misión para la tarea o proyecto. escribir (y iterar) la visión y el alcance de una tarea o proyecto documento.			X	5
26	Desarrollar, implementar y hacer cumplir la reunión. reglas que describen cómo el personal del proyecto los miembros deben tratarse unos a otros.			X	6
27	Desarrollar y aplicar un conjunto de pautas para reuniones efectivas y lineamientos para envío de correo electrónico eficaz.			X	6
28	Realizar una evaluación de riesgos de productos nuevos y requisitos cambiantes.	X		XX	3, 7
29	Aprenda a gestionar equipos de forma eficaz.	X		XX	7, 9
30	Establecer un proceso y mejora de la calidad. clima de mejora.	X		XX 8	

1. Desarrollar un plan de requisitos.

Las razones para realizar la planificación con respecto a las actividades relacionadas con los requisitos y los contenidos sugeridos de un plan de requisitos se analizan en Capítulos 1 y 5.

2. Escribir requisitos que cumplan con los criterios de un buen requisito proporcionado en Tabla 1.1.

Si no realiza este paso, deténgase aquí. La Tabla 1.1 proporciona una lista de criterios sugeridos para un buen requisito. Hay mucha información sobre este tema. disponible: varios autores han proporcionado varias versiones con resultados muy similares. criterios. Sorprendentemente, estos criterios rara vez se aplican en la práctica. Esto es un

ejemplo flagrante de una situación en la que sabemos cómo hacerlo mejor, pero elegimos no utilizar nuestro conocimiento y experiencia. Esta es una oportunidad para que usted haga una valiosa contribución a los proyectos que apoya. Considere incluir estos criterios como una lista de verificación en su herramienta de requisitos automatizada.

Descubrirá que se ahorrará mucho tiempo y dinero como resultado de la aplicación de los criterios.

3. Identificar e involucrar a todos los actores de la tarea o proyecto.

Asegúrese de que todas las partes estén identificadas e involucradas en el proceso de desarrollo de requisitos. Con demasiada frecuencia, no identificamos a todas las partes interesadas que deberíamos. Omisiones a un grupo de partes interesadas puede provocar un estallido más adelante en el trabajo de desarrollo. Las partes interesadas incluyen el cliente, los usuarios, las organizaciones de control y gestión de programas, los equipos de desarrollo y arquitectura, el personal jurídico, los grupos de pruebas, los clientes de interfaz, etc.

En el Capítulo 5 se proporcionan sugerencias y enfoques sobre cómo lograr esto.

4. Asegurar que los objetivos de la tarea o proyecto hayan sido identificados, documentados y acordados por las partes interesadas.

Esto debe hacerse con antelación y puede lograrse escribiendo un “documento de visión y alcance” [1]. La disponibilidad de objetivos de proyecto definidos y acordados ayuda al equipo de desarrollo a mantener el enfoque y proporciona una base común para identificar los requisitos reales y evaluar sus prioridades. Ayuda a garantizar que todos consideren el sistema o las capacidades de software necesarias desde la misma perspectiva y también ayuda a quienes proporcionan la financiación a comprender qué se debe hacer y cómo respalda a la organización.

5. Utilice talleres de requisitos para lograr una visión compartida, facilitar el compromiso y lograr la aceptación de todas las partes interesadas.

De todos los métodos y técnicas de recopilación de requisitos, los talleres de requisitos parecen ser los más eficaces. La definición de Ellen Gottesdiener de un taller de requisitos proporciona información sobre por qué es tan eficaz [2, p. 9]:

Un taller de requisitos es una reunión estructurada en la que un grupo cuidadosamente seleccionado de partes interesadas y expertos en contenido trabajan juntos para definir, crear, perfeccionar y llegar a un acuerdo sobre los entregables (como modelos y documentos) que representan los requisitos del usuario. El beneficio del proceso del taller es que fomenta la comunicación en equipo, la toma de decisiones y el entendimiento mutuo. Los talleres también son una forma eficaz de reunir a clientes, usuarios y proveedores de software para mejorar la calidad de los productos sin sacrificar el tiempo de entrega. Estas sesiones tienden a comprometer a los usuarios con el proceso de definición de requisitos y promover su sentido de propiedad de los entregables y, en última instancia, del sistema.

6. Proporcionar capacitación en requisitos a los RA, a los miembros del personal del proyecto y a los partes interesadas.

Del material presentado hasta ahora debería resultar evidente que es Es ventajoso proporcionar capacitación sobre requisitos para tres grupos distintos: AR, miembros del personal del proyecto y partes interesadas. La razón es que La experiencia en la industria tiene mucho que ofrecer a cada grupo para mejorar la acercarse. El tema difiere para cada uno de los grupos, como se señala en Capítulo 5. De particular beneficio es la comprensión de que la comunicación entre todos los grupos mejora cuando todos comparten las mismas ideas y comprensión.

7. Identificar los requisitos reales. Colaborar con clientes y usuarios en relación con los requisitos establecidos para identificar los requisitos reales. Mire los requisitos desde múltiples puntos de vista [3].

Confío en que ya comprenda la diferencia entre los requisitos establecidos y los requisitos reales. Su principal responsabilidad es colaborar con clientes y usuarios sobre los requisitos establecidos para identificar los requisitos reales. Este es el Rol 1 en el contexto de los roles definidos en el Capítulo 2: servir como facilitador de requisitos para trabajar en colaboración con los clientes, usuarios, arquitectos y diseñadores de sistemas para identificar los requisitos reales. Su primer paso será convencer a su PM, cliente y usuarios de que es Es esencial y vale la pena invertir tiempo y esfuerzo adicionales en el proceso de requisitos, en este caso, para revisar los requisitos establecidos y evolucionar los requisitos reales. requisitos utilizando un concepto o mecanismo de equipo conjunto. No te saltes esta crítica paso: es el problema más importante de la industria en ingeniería de requisitos y al que casi siempre se le presta atención insuficiente. Aplicar efectivo técnicas de recopilación de requisitos como las descritas en el Capítulo 5. Al colaborar con sus clientes y usuarios, adapte la lista de verificación proporcionada en Tabla 5.1 a las necesidades de su proyecto en su entorno. revisar lo real requisitos desde una variedad de perspectivas, es decir, las de todas las partes interesadas del proyecto.

8. Documente la justificación de cada requisito, es decir, por qué es necesario.

La justificación es un atributo que debe incluir en su herramienta automatizada de requisitos. La experiencia de la industria muestra que al dar el paso de documentar la justificación de cada requisito, hasta la mitad de los requisitos establecidos Los requisitos pueden eliminarse. El ahorro en términos de no tener que hacer El trabajo técnico posterior para cumplir con los requisitos eliminados es obviamente enorme. Además, este esfuerzo aclarará y endurecerá los requisitos eliges conservar. La experiencia de Ivy Hooks es que registrar la justificación para cada requisito reduce el número total de requisitos, expone malas suposiciones, elimina la implementación no deseada, mejora la comunicación entre los miembros del equipo, acorta el ciclo de revisión, mantiene el conocimiento corporativo, reduce el riesgo al definir un producto derivado y soporta los costos de mantenimiento y operación [4].

9. Utilice técnicas eficaces de recopilación de requisitos.

Este fue el tema del Capítulo 5. Algunas técnicas de recopilación de requisitos son más efectivas que otras. Asegúrese de que alguien en su tarea o proyecto haya utilizado previamente las técnicas seleccionadas con éxito.

10. Involucrar a los clientes y usuarios durante todo el esfuerzo de desarrollo.

Reconocer que la experiencia de la industria muestra que los proyectos que involucran a clientes y usuarios a lo largo del proceso de desarrollo son exitosos: diseñar y utilizar mecanismos para mantener involucrados a los clientes y usuarios del proyecto, como el equipo conjunto, técnicas colaborativas de recopilación de requisitos y un control de configuración conjunto. (CCB) para gestionar el proyecto.

11. No tome decisiones sobre requisitos.

Por decisiones sobre requisitos me refiero a decisiones sobre lo que es o debería ser un requisito, incluida su redacción. Una de las formas en que los RA creamos problemas para nuestros proyectos es tomando decisiones sobre requisitos. Establezca una política personal para no tomar decisiones sobre requisitos. Las decisiones sobre los requisitos son responsabilidad del cliente y del usuario dentro del mecanismo del equipo conjunto. Si bien puede ser más rápido y más fácil decidir algo en lugar de obtener una aclaración, resista esta tentación porque es peligrosa.

Reflexione sobre lo difícil que es comunicarse eficazmente y cómo las personas interpretan de manera diferente las cosas que escuchan, leen y ven. Tienes un alto riesgo de tomar una decisión incorrecta. Además, su decisión podría tener un impacto negativo importante en el proyecto, aunque no sea intencionado.

El enfoque de no tomar decisiones sobre requisitos debe comunicarse a todo el equipo de desarrollo, para garantizar que los desarrolladores tampoco tomen decisiones sobre requisitos. Esto debe aclararse en la capacitación relacionada con los requisitos proporcionada al equipo del proyecto.

12. No dorar, es decir, añadir características o capacidades.

¡No decida que tiene una idea que sabe que al cliente y a los usuarios les encantará! De hecho, puede que les encante, y puede aumentar el costo y el cronograma, así como el trabajo técnico que ya se ha completado (léase: provocar reelaboración). Cumplir con los requisitos mínimos reales. No dorar.

13. Utilice un glosario de proyectos y una lista de acrónimos de proyectos.

Anteriormente hice esta sugerencia y proporcioné la justificación para hacerlo. Consulte el Capítulo 5, paso 8.

14. Iterar los requisitos y la arquitectura repetidamente para evolucionar mejor los requisitos y una arquitectura más robusta.

El punto aquí es que los requisitos y la arquitectura se impactan entre sí. A medida que modificamos la arquitectura para abordar el cumplimiento de la realidad

Si mejoramos los requisitos, aprendemos más sobre los requisitos y descubrimos que los cambios en la arquitectura nos hacen querer cambiar los requisitos, y así seguimos. La iteración repetida de los requisitos y la arquitectura da como resultado mejores requisitos reales y una arquitectura más robusta. Este trabajo se puede lograr en conexión con las tres o cuatro iteraciones del proceso de desarrollo de requisitos previamente recomendado.

15. Utilice expertos en el campo/PYME que tengan conocimientos y experiencia en el áreas funcionales abordadas por el esfuerzo técnico.

Mencioné anteriormente algunas ventajas aportadas a un proyecto por un RA recién asignado, como tener una nueva perspectiva, libre de las limitaciones y la historia del sistema y los procedimientos heredados. La otra cara de esto es el valor de involucrar a personas que tengan amplia experiencia y conocimiento en las áreas funcionales que aborda el sistema. Tienen un profundo conocimiento de por qué se hacen las cosas de determinada manera y analizan las necesidades del cliente con una perspectiva experimentada que puede incluir aspectos inimaginables para aquellos con menos conocimientos o menos experiencia. Involucre a dichas personas en actividades de recopilación de requisitos, por ejemplo, talleres de requisitos, o utilícelas como asesores.

16. Cuantificar el ROI para seleccionar requisitos mecanismos, prácticas, métodos, tecnologías. Técnicas y herramientas a utilizar.

Proporciono información detallada sobre esta mejor práctica en Prácticas efectivas de requisitos [5, págs. 50–52]. Tomar decisiones basadas en datos en lugar de hacerlo espontáneamente o mediante la intuición es una buena práctica; en nuestra empresa, nos referimos a este hábito como "administrar por hechos". Su PM debe esperar que se proporcionen datos cuando se soliciten decisiones. Proporcionar datos sobre el retorno de la inversión en prácticas de requisitos mejoradas es una forma de obtener apoyo para sus sugerencias y recomendaciones. Realmente no es difícil desarrollar información sobre el retorno de la inversión (ROI). Utilice la plantilla proporcionada en mi libro anterior.

17. Identificar los requisitos mínimos que satisfagan las necesidades reales.

Algunas personas tienen dificultades con el concepto de identificar los requisitos mínimos. Interpretan esto como no hacer todo lo posible para satisfacer a los clientes. El punto es que necesitamos estar asociados con nuestro cliente, comprometidos con el éxito del proyecto, y el proceso de desarrollo de requisitos debe resultar en un conjunto de requisitos que sean el mínimo requerido para satisfacer las necesidades reales. Cualquier requisito y característica adicional que vaya más allá de las necesidades reales complica el proceso de desarrollo, lo encarece, toma tiempo adicional, pone en riesgo la calidad del producto del trabajo y potencialmente pone en peligro el éxito del proyecto (definido como un sistema eficaz, completado a tiempo y dentro del presupuesto), utilizando una relación de asociación beneficiosa para todos durante todo el ciclo de vida del sistema). El proceso de desarrollo de sistemas y software es

complicado y difícil. Ambos socios deben esforzarse constantemente por cumplir requisitos mínimos reales en aras del éxito del proyecto. Si tiene dificultades para comprender o aceptar este concepto, lea "Cumplir con los requisitos mínimos".

Requisitos: cualquier cosa más es demasiado", por Neal Whitten [6]. Determinar cuáles son los requisitos mínimos priorizados adecuadamente debe involucrar a miembros del equipo de desarrollo, la comunidad de usuarios y el CCB del proyecto: esta tríada debería trabajar en conjunto para recomendar la priorización y el financiamiento de los requisitos.

18. Priorice los requisitos desde el principio y con frecuencia.

Es igualmente importante priorizar los requisitos reales de manera temprana y frecuente. Tenga en cuenta (y ayude a sus clientes y usuarios a comprender) que nunca hay suficiente tiempo y dinero para hacer todo y que no se cumplan todos los requisitos de igual prioridad. Utilice su equipo conjunto o un mecanismo similar para llegar a un acuerdo conjuntamente sobre las prioridades de requisitos. Hay artículos y herramientas disponibles para ayuda.¹ Tómese el tiempo para leerlos y utilizarlos. No dejes de abogar y aplicar mecanismos, prácticas, métodos, técnicas y herramientas que mejorará las posibilidades de que su proyecto tenga éxito.

Una vez identificados y priorizados los requisitos reales, el equipo de desarrollo puede estimar el esfuerzo necesario para proporcionar funciones adicionales. y el cliente puede evaluar el costo de proporcionarlos y decidir si valen el dinero y el tiempo adicionales. La clave es garantizar que el Los productos de trabajo desarrollados serán aceptables para el cliente y los usuarios. lograr el acuerdo de todas las partes interesadas desde el principio.

19. Proporcionar inspecciones de todos los documentos relacionados con los requisitos.

La justificación para realizar inspecciones de todos los documentos relacionados con los requisitos se proporciona en el Capítulo 7, tema 11.

20. Limitar los cambios a los requisitos y la adición de nuevos requisitos de manera consistente. con presupuesto y cronograma adicionales puestos a disposición por el cliente para completar la tarea, proyecto o sistema.

Esta es la segunda cosa más importante que puede hacer una AR para apoyar un proyecto (después de establecer un mecanismo y un enfoque de colaboración conjunto para identificar los requisitos reales). Los cambios de requisitos y los nuevos requisitos son el segundo. razón principal por la que los proyectos se salen de control. Tu responsabilidad en este ámbito es familiarizar a su equipo de proyecto, a su cliente y a los usuarios con experiencia en la industria y ganar compromiso para controlar los cambios y nuevos requisitos. Por ejemplo, considere el enfoque de tener versiones y lanzamientos posteriores de productos de trabajo, en lugar de pretender que el proyecto pueda adaptarse a los cambios mientras está en desarrollo.

1. Véase, por ejemplo, "Lo primero es lo primero: priorizar los requisitos" de Karl E. Wieggers, Revista de desarrollo de software.

7(9) (septiembre de 1999): 24–30. Wieggers proporciona una herramienta de hoja de cálculo fácil de usar que se puede descargar desde su sitio web. sitio, en www.processimpact.com (consulte el botón "obsequios"). Véase también los requisitos de software de Wieggers, [1].

Por supuesto, deberá asegurarse de que cualquier solicitud de cambio representa una necesidad real y es, por tanto, una necesidad real. Como antes, determinar el fundamento de la solicitud: ¿por qué es necesaria? Un análisis cuidadoso permitirá eliminar hasta la mitad de las solicitudes. Lo más importante es que el equipo de desarrollo debe aprender a decir no. no esta en ninguno
El interés del cliente ni del desarrollador para permitir que el proyecto se lleve a cabo. fuera de control. Crear y mantener una asociación con el objetivo principal de completar exitosamente el proyecto. La asociación se trata de comprometerse con el éxito y ser flexible para lograr el resultado deseado.

Establecer objetivos para diferentes situaciones, por ejemplo, requisitos del 0,5%. cambio por mes para requisitos validados, y buscar verificación de la equipo técnico que cualquier cambio propuesto no pondrá en peligro el éxito del proyecto. Los cambios en los requisitos después de establecer la línea base de requisitos ponen en peligro resultados y éxito del proyecto, a menos que sirvan para aclarar la intención de un requisito en lugar de cambiar la funcionalidad. Algunos creen que un límite de La volatilidad de los requisitos del 0,5% es demasiado estricta, poco realista e inalcanzable. Sin embargo, proporciona un buen objetivo. Una interesante directriz de la industria La experiencia es que un cambio de un tercio en los requisitos (33% por año o 2,75% mensual) resultará en una duplicación de los costos del proyecto. Seguir tu métrica de volatilidad de requisitos dentro del mecanismo del equipo conjunto. Asegurar que el cliente está dispuesto a proporcionar cronograma y presupuesto adicionales en proporción al porcentaje de volatilidad de los requisitos; de lo contrario, no aceptar cambios en los requisitos o la adición de nuevos requisitos. Aprender decir no.

21. Utilizar versiones y lanzamientos de productos de trabajo para adaptarse a nuevos requisitos.
requisitos modificados y requisitos de menor prioridad.

Se analiza la importancia de utilizar versiones y lanzamientos de productos de trabajo. en el Capítulo 5.

22. Utilice una herramienta de requisitos automatizada de potencia industrial. Proporcionar y utilizar atributos de requisitos.

Seleccione con antelación y con cuidado su herramienta de requisitos automatizados de potencia industrial. Asegúrese de que la herramienta seleccionada respalte su proceso. Seleccionando la herramienta sin tener primero el proceso de requisitos implementado puede causar que el proyecto forzar el ajuste de su proceso a la herramienta, un riesgo importante. La herramienta equivocada o una herramienta que es demasiado complejo porque el trabajo puede retrasar los esfuerzos del proyecto. Asegúrate de que Utilice una herramienta de requisitos automatizada comprobada: no puede permitirse el lujo de invertir el dinero. tiempo y esfuerzo necesarios para escribir software que realice funciones tales como trazabilidad. Dadas las herramientas comerciales disponibles, no es rentable desarrollar sus propias capacidades de herramientas de requisitos automatizadas.

Proporcione capacitación formal a quienes utilizarán la herramienta con mayor frecuencia; esta es una inversión valiosa y no debe pasarse por alto. Determine los atributos de requisitos que se necesitan; consulte la discusión sobre atributos en el Capítulo 5. Con demasiada frecuencia, la elección de la herramienta de requisitos automatizados que se utilizará en un proyecto está dictada por factores fuera del control.

del proyecto. Por ejemplo, la experiencia de una AR respecto de la selección de La herramienta de requisitos automatizada para respaldar cinco proyectos diferentes que apoyó fue la siguiente:

En un proyecto, desarrollamos nuestra propia base de datos de requisitos utilizando Informix porque ya teníamos Informix y mucha experiencia en usándolo.

Por otro lado, propusimos un enfoque OO utilizando Rational Unified Process (RUP) y Rational Tool Suite: RequisitePro fue el herramienta predeterminada simplemente porque era parte de nuestro conjunto de herramientas general.

En un tercer proyecto, se seleccionó Rational debido a que el director técnico impartía una clase sobre casos de uso en una universidad local y estaba ya está familiarizado con Rational Suite.

En otro proyecto más, el cliente especificó Rational RequisitePro en la SOW.

Y por otro lado, el proyecto utilizó DOORS porque el cliente utilizó PUERTAS.

Entonces, de cinco proyectos, la experiencia de esta AR fue que la herramienta de requisitos automatizados se seleccionó basándose en criterios arbitrarios en las cinco situaciones: debido a restricciones presupuestarias, porque estaba predeterminado o porque alguien tenía una preferencia personal. No hubo ningún caso en el que un Se seleccionó la herramienta de requisitos automatizados porque era la mejor opción para ¡Ese proyecto específico! El enfoque en la selección de los requisitos del proyecto. La herramienta contrastaba con la aplicada para CM y herramientas de prueba, donde más Se utilizaron procesos de pensamiento lógico y criterios específicos. La AR debería recomendar que se redacte un estudio sobre el comercio de herramientas y requisitos [7] para garantizar que Los criterios para seleccionar la herramienta de requisitos automatizados son consistentes. con las necesidades del proyecto.

23. Desarrollar o adaptar y utilizar políticas de requisitos organizacionales y de proyecto y un proceso de requisitos que se mejora continuamente en su tarea, proyecto u organización. Invierta del 8% al 14% de los costos totales del proyecto en (ciclo de vida del sistema) proceso de requisitos.

Es útil tener (y utilizar) una política organizacional relativa a los requisitos. Todos conocemos proyectos y organizaciones que tienen políticas, pero no los uses en la práctica. Estoy hablando de una situación diferente: ¡recomiendo que las organizaciones y proyectos tengan políticas y las utilicen! El desarrollo de la política organizacional debe involucrar a la alta dirección. e incluir su dirección de que los requisitos se utilizarán como base para actividades de ingeniería y gestión. Las políticas organizativas relativas a los requisitos pueden ser tan simples como las sugeridas por los dos áreas de proceso relacionadas con requisitos del CMMI [8], desarrollo de requisitos y RM:

Respecto al desarrollo de requisitos: "Con el fin de identificar y satisfacer necesidades de los clientes, los proyectos: (a) Recopilarán las necesidades de las partes interesadas, (b) Formularán los requisitos del producto y de los componentes del producto, y (c) Analizarán y validarán estos requisitos".

Con respecto a RM: "Para garantizar que se satisfagan las necesidades de los clientes, los proyectos: (a) Gestionarán los requisitos y requisitos cambios, y (b) identificar inconsistencias entre el trabajo del proyecto y requisitos".

Porque las actividades relacionadas con los requisitos que se realizan en los proyectos son fundamentales para el éxito de los proyectos, propongo un proyecto más detallado política de requisitos, como la proporcionada en Prácticas efectivas de requisitos [5, págs. 119-122] y también disponible en mi sitio web (www.ralphyoung.net). Este artefacto sirve como plantilla que debes adaptar (modificar) a las necesidades de su proyecto en su entorno. Una alternativa a tener un proyecto La política de requisitos es incorporar los componentes necesarios en el plan del proyecto. proceso de requisitos.

Desarrollar o adaptar y utilizar un proceso de requisitos documentado. Ver Consulte el capítulo 8 de este libro para obtener orientación sobre cómo diseñar un proceso. No es difícil (o al menos no tiene por qué serlo) diseñar o adaptar un proceso. Si Si no está familiarizado con el diseño y el uso de procesos, es posible que desee Solicite la ayuda de alguien que esté muy familiarizado con esto para que le sirva como un facilitador para sus partes interesadas. Es importante para los miembros de la equipo del proyecto para tener una buena comprensión de los procesos que realiza el equipo. usando. Tómese el tiempo para informar a todos sobre los procesos y asegúrese de que haya consenso y que se acepte el enfoque de proceso: los miembros Los miembros del equipo pueden ofrecer mejoras en función de sus experiencias.

Los datos proporcionados en la Figura 4.1 de Prácticas efectivas de requisitos [5] proporcionan un caso convincente para invertir del 8% al 14% de los costos del proyecto en la proceso de requisitos. De la discusión hasta ahora debería resultar evidente que proporcionar prácticas de requisitos mejoradas es una buena inversión que produce influencia en el control de costos, por ejemplo, de retrabajo (40% a 50% de los costos totales del proyecto promedio). Fomentar la inversión en el proceso de requisitos del proyecto y utilizar prácticas de requisitos efectivas brinda oportunidades para que la RA tenga un impacto positivo importante en el proyecto. éxito.

24. Utilice mecanismos, enfoques, métodos, técnicas y herramientas de requisitos probados y familiares.

Comprometerse a utilizar mecanismos, enfoques, métodos, técnicas y herramientas. Debería estar familiarizado con ejemplos de estos de partes anteriores del libro, pero mencionaré algunos ejemplos en cada categoría para que los tengas siempre en mente:

Mecanismos: el equipo conjunto (o como usted elija llamar a este mecanismo de colaboración); un conjunto de reglas de conducta en su tarea, proyecto o

organización para describir cómo se tratarán los miembros entre sí; PDCA para determinar cómo fueron las reuniones o cómo nos está yendo en un momento dado, por ejemplo, al completar un hito; un Propósito, Agenda y Límite (PAL) proporcionado antes de las reuniones para que las personas puedan prepararse para la reunión y saber cuánto tiempo planificar para la reunión;

Enfoques: asociación; uso de revisiones por pares y técnicas de prevención de defectos (DP) a lo largo de la tarea o proyecto; planificación y seguimiento de proyectos; capacitación; CM; control de calidad; utilizar técnicas para facilitar la comunicación del proyecto; medición; Etcétera;

Métodos: métodos de recopilación de requisitos como entrevistas, análisis de documentos, talleres de requisitos, creación de prototipos, guiones gráficos, escenarios y modelado;

Técnicas: gestión de riesgos de proyectos, revisiones por pares, DP, "bolsas marrones";

Herramientas: ReqPro, DOORS y otras herramientas de requisitos automatizados que se indican en la Tabla 5.7; herramientas automatizadas de gestión de riesgos; gráficos de ladrillos; un cuaderno de proyecto.

Puede que le resulte útil aprender y utilizar un nuevo método o técnica; Sin embargo, reconozca que se necesitará tiempo y esfuerzo para que las personas aprendan nuevos métodos y técnicas para utilizarlos de manera efectiva y que existe un riesgo al utilizar un método o técnica que no está probado y no es familiar.

Toma decisiones sobre el uso de nuevos métodos y técnicas con los ojos bien abiertos.

25. Establecer una meta, propósito o misión acordados para la tarea o proyecto. Escriba (e itere) un documento de visión y alcance de una tarea o proyecto.

Como se señaló al principio de este capítulo, la falta de una meta, un propósito o una misión acordados para una tarea o proyecto hace que sea difícil lograr algo de valor. Es necesario poder articular el objetivo y obtener el apoyo de las partes interesadas para lograrlo. Escribir e iterar un documento de visión y alcance de una tarea o proyecto proporciona una base común para identificar y priorizar objetivos más específicos e identificar los requisitos reales.

26. Desarrollar, implementar y hacer cumplir reglas de reuniones que describan cómo deben tratarse entre sí los miembros del personal del proyecto.

Esto implica dos elementos clave: establecer y seguir una agenda (PAL) y seguir reglas de conducta. Cada uno de estos elementos es un componente clave de las reuniones diseñadas para obtener y adoptar nuevos requisitos o cambios a los requisitos existentes. La persona que solicita una reunión proporciona un PAL antes de la reunión para que todos sepan lo que se va a discutir, cada persona pueda prepararse adecuadamente y todos sepan cuándo terminará la reunión.

Según mi experiencia, disfruto el trabajo y me siento más eficaz cuando mis compañeros de trabajo aprecian y apoyan mi contribución al esfuerzo general. He descubierto que tener un conjunto de reglas de conducta para los esfuerzos laborales en los que estoy involucrado ha sido una forma eficaz de facilitar una actitud de apoyo mutuo en nuestro entorno laboral. Ejemplos de reglas de conducta que valoro incluyen las siguientes:

- respetar a cada persona;
- Compartir la responsabilidad;
- Criticar ideas, no personas;
- Mantenga la mente abierta;
- Cuestionar y participar;
- Llegar a tiempo;
- Mantener las interrupciones al mínimo;
- Gestionar por hechos.

Estas reglas están publicadas en las salas de conferencias de nuestra empresa. Las personas son llamadas a criticar cuando violan una de estas reglas. Me siento capacitado para contribuir con mis mejores esfuerzos. Sé que mis compañeros de trabajo me respetarán, incluso cuando mis ideas parezcan inusuales. Intentamos apoyarnos unos a otros en todas las formas posibles. Todos llegan a tiempo a las reuniones y comenzamos a tiempo.
No se permiten conversaciones secundarias. Se espera concentración. Ahorramos cantidades increíbles de tiempo. Pero lo más importante es que nos respetamos unos a otros y estamos ahí para ayudarnos.

27. Desarrollar y aplicar un conjunto de pautas para reuniones efectivas y pautas para reuniones efectivas.
correo electrónico activo.

Pasamos mucho tiempo en reuniones y leyendo y escribiendo correos electrónicos. Es lógico que una serie de directrices para estos devoradores de tiempo les ahorre tiempo y esfuerzo. Anteriormente recomendé un conjunto de pautas para cada uno: consulte Prácticas efectivas de requisitos [5, págs. 165–167 y págs. 167–172, respectivamente].

Las pautas específicas son menos importantes que (1) comprometerse a utilizar pautas para estos propósitos, y (2) que las personas en proyectos y organizaciones se tomen el tiempo para desarrollar pautas que respetarán y cumplirán.
usar.

28. Realizar una evaluación de riesgos de requisitos nuevos y cambiantes.

En el Capítulo 7, tema 18, se proporciona orientación sobre cómo realizar evaluaciones de riesgos relacionados con los requisitos.

29. Aprenda a gestionar equipos de forma eficaz.

Hemos visto a lo largo del libro lo importante que es trabajar en colaboración, lograr consenso, lograr la aceptación de las partes interesadas y lograr

resultados trabajando en equipo. La AR debería desarrollar la capacidad de gestionar equipos de forma eficaz. Hay sesiones de capacitación y talleres a los que se puede asistir para aprender y practicar las habilidades y técnicas necesarias. Uno de los mejores libros sobre gestión de equipos es *The Team Handbook* de Scholtes et al. [9].

30. Establecer un clima de mejora de la calidad y mejora de procesos.

En el Capítulo 8 se proporciona orientación sobre cómo lograr esto.

Resumen

Este capítulo presenta 30 mejores prácticas para el desarrollo y la gestión de requisitos para su consideración. No se puede hacer todo, al menos no al mismo tiempo. Desempolva tu plan de requisitos. Desarrollar un enfoque razonable para implementar, implementar e institucionalizar aquellas mejores prácticas que considere apropiadas para su proyecto en su entorno durante un período de tiempo razonable. Colabora con el equipo de tarea o proyecto para priorizar el valor de las mejores prácticas que decidas implementar. Escriba un plan de acción que permita al proyecto implementarlos. Para cada buena práctica, defina las acciones y el cronograma necesarios para implementarla. Haga esto en colaboración con su equipo de proyecto y su cliente. Obtenga su aceptación y apoyo para las mejores prácticas seleccionadas. Comunique lo que está haciendo mediante reuniones del personal del proyecto o reuniones de trabajo. Involucre a su equipo de proyecto y a su cliente en el proceso de toma de decisiones para obtener la aceptación y el apoyo de los demás. Lo principal es garantizar que el equipo del proyecto avance junto con su cliente, no tener la mayor cantidad de mejores prácticas. Recuerde, se necesita compromiso para lograr algo de valor.

Caso de estudio

Hace varios años, me pidieron que ayudara a abogados que trabajaban en un caso legal entre un contratista de integración de sistemas y el gobierno de Estados Unidos. El gobierno rescindió el contrato por incumplimiento y luego volvió a adquirir el sistema de otro contratista. El sistema construido por el nuevo contratista explotó cuando fue expuesto a volúmenes de datos reales. Las pocas horas de asistencia de consultoría solicitadas originalmente se convirtieron en años de apoyo en litigios a medida que avanzaba el caso, antes de llegar a un acuerdo cinco años después.

¿Qué causó esta desconexión masiva? La respuesta corta es la falta de Comprensión de las funciones y responsabilidades para la gestión de requisitos.

El gobierno reconoció que necesitaba requisitos y había contratado años antes un estudio de viabilidad que incluía requisitos de datos y requisitos funcionales. Luego, cuando quedó claro que necesitaba avanzar rápidamente y ya no tenía tiempo para completar el contrato para construir el sistema, encargó al contratista de desarrollo una serie de órdenes de trabajo. Las dos primeras tareas fueron (1) "Evaluar el hardware y

especificaciones de software y realizar un análisis de requisitos para la red de área local (LAN) de la sede central" y (2) "Revisar y validar la estudio (previo), frente a los requisitos actuales".

La primera tarea, "evaluar las especificaciones de hardware y software y realizar un análisis de requerimientos para la LAN Sede", se tomó como Tarea de bajo riesgo para recomendar ordenadores personales (PC) ofimáticos. Hardware y software para estaciones de trabajo y servidores. fue emprendido rápidamente, utilizando fondos de fin de año, y tenía como objetivo entregar equipos a los usuarios en el campo. El contratista consideró que esta tarea era "alcanzable".

La segunda tarea, "revisar y validar el trabajo previo", resultó en una documento que describía áreas donde los requisitos funcionales actuales habían Cambió e identificó áreas donde era necesario trabajar con requisitos adicionales. El contratista estaba más preocupado por poder lograr esto funciona de manera efectiva, debido a los problemas relacionados con los requisitos.

El gobierno encargó una tercera tarea: diseñar una capacidad de transmisión electrónica para el sistema. La SOW para esta tarea fue extremadamente detallado y requirió el diseño de numerosas interfaces. Esto sugirió un cambio importante en la arquitectura. El estudio original había asumido que el El sistema sería un sistema centralizado basado en mainframe. De repente, Quedó claro que el cliente tenía un enfoque diferente en mente: una arquitectura cliente-servidor. Al expresar cómo se implementarían los requisitos, el gobierno estaba imponiendo restricciones detalladas a la solución. El contratista empezó a tener ansiedad porque había muchas incógnitas, como el diseño del resto del sistema más allá de su capacidad de transmisión. El trabajo continuó sin una comunicación efectiva entre el gobierno y el contratista. Las cuestiones abiertas no se resolvieron.

En una revisión importante del diseño, el contratista y cierto personal del gobierno se reunieron por primera vez y la luz comenzó a amanecer. El diseñador principal de la capacidad de transmisión electrónica declaró: "Ahora sé lo que ¡desear!" Todos declararon que la reunión fue un éxito. Un elemento de acción fue preparar un plan para cuando se complete la capacidad de transmisión.

Cuando se entregó el plan, indicó que la fecha proyectada de finalización del sistema sería un año más allá de la fecha deseada. Esto impulsó el gobierno a rescindir el contrato. El gobierno esperó hasta Se completaron otros entregables del diseño, se los entregó a un nuevo contratista y se puso en marcha el trabajo para construir rápidamente el sistema.

En las pruebas de aceptación del sistema, el sistema explotó con una corrupción masiva de datos y estaba claro que simplemente no funcionaría. ¿Cuál fue el problema?

Algunas de las cuestiones relacionadas con los requisitos fueron las siguientes:

1. Las especificaciones de hardware y software evaluadas en la tarea 1.

no se hicieron correctamente porque los componentes seleccionados no pudieron acomodar el volumen de datos. ¿Eran esas especificaciones simplemente requisitos del sistema para la compra de PC comerciales, o eran esos PC compradas con la intención de que fueran parte de la solución. para el sistema en general? Este matiz resultaría más tarde significativo,

porque el gobierno afirmó que la evaluación de las PC fue por su idoneidad para apoyar el sistema de toda la agencia. Los requisitos de desempeño para tal sistema no habían sido definidos, porque Nadie del lado del contratista se dio cuenta de que la arquitectura no era ya no está centralizado.

2. No se desarrolló ningún documento o repositorio general unificado de requisitos. Los requisitos se encontraban en numerosas fuentes de antigüedad y validez variables (por ejemplo, notas escritas a mano por personal gubernamental sobre los entregables que revisaron). En algunos casos, los requisitos eran contradictorios o inexistentes. Áreas identificadas en la revisión y validación de documentos de requisitos previos que necesitaban una definición adicional de requisitos se dejaron así porque no Se recibió tarea para explorarlos. No hubo ningún CM del requisitos.
3. La especificación de requisitos detallados por parte del gobierno para la capacidad de transmisión, sin comunicar la arquitectura general del sistema ni permitir al contratista diseñar el arquitectura, fue una especificación excesiva de los requisitos de diseño y una restricción al sistema, una que (como resultó) no funcionaría.
4. Volúmenes de datos identificados en los requisitos de datos realizados por el organismo anterior. El contratista creció en algunos lugares clave sin ninguna reevaluación de el impacto general en la arquitectura, hasta que el sistema entregado no funcionaría.
5. El diseño del sistema por parte del contratista fue considerado por el gobierno era tan pobre que se rescindió el contrato, pero el gobierno hecho de que el gobierno ordenó al contratista siguiente que utilizara esos mismos entregables de diseño para construir el sistema implica la aceptación de ese diseño. De hecho, eran un conjunto de requisitos del sistema. para el nuevo contratista.

La principal lección que se puede aprender de este estudio de caso se refiere a los roles y liderazgo de las tareas de requisitos. Porque el gobierno definió a través de su asignación de tareas, qué partes de los requisitos requeriría la contratista asumiría la responsabilidad y de qué partes el gobierno especificaría en detalle, no existía una estrategia o proceso general de movilización de recursos. Alguno de los resultados finales de este escenario fueron (1) los requisitos del usuario no eran cumplió lo previsto tanto por el gobierno como por los contratistas, (2) el contrato original fue rescindido, (3) el sistema desarrollado por el segundo contratista no funcionó, (4) se desperdió mucho dinero y tiempo, (5) algunos personas perdieron sus trabajos debido a problemas que surgieron, (6) algunas familias se vieron afectados negativamente debido a diversos problemas interpersonales que se desarrollaron, (7) se gastaron años en litigios costosos, (8) importantes información sobre ingeniería de sistemas y software y los problemas de ambas partes fue expuesto en diversas comunicaciones, incluida la periódicos, y (9) se produjeron muchas acusaciones. En mi experiencia, esto

El escenario no es único: escenarios relacionados se han repetido de diversas maneras durante los últimos veinte años y continúan incluso hoy.

Nombre omitido a petición, consultor de ingeniería de requisitos

Referencias

- [1] Wiegers, KE, Requisitos de software, 2.^a ed., Redmond, WA: Microsoft Press, 2003, 77–93.
- [2] Gottesdiener, E., Requisitos por colaboración: talleres para definir necesidades. Boston, MA: Addison-Wesley, 2002.
- [3] Sommerville, I., P. Sawyer y S. Viller, "Puntos de vista para la obtención de requisitos: un enfoque práctico". Actas de la Conferencia Internacional sobre Ingeniería de Requisitos de 1998 (ICRE'98), 6 al 10 de abril de 1998, Colorado Springs, CO, Nueva York: IEEE Computer Society, 1998, 74–81. Véase <http://computer.org/procedimientos/icre/8356/8356toc.htm>. Véase también el Capítulo 13 de Ingeniería de requisitos de I. Sommerville y P. Sawyer : una guía de buenas prácticas, Nueva York: John Wiley & Sons, 1997, y Ingeniería de requisitos de G. Kotonya e I. Sommerville: procesos y técnicas, Chichester, Reino Unido: John Wiley e hijos, 1998.
- [4] Hooks, IF y KA Farry, Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos, Nueva York: AMACOM, 2001, 120–133.
- [5] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.
- [6] Whitten, N., "Cumplir con los requisitos mínimos: cualquier cosa más es demasiado", PM Network, septiembre de 1998, p. 19.
- [7] Young, RR, "Estudio sobre el comercio de herramientas y requisitos", en www.ralphyoung.net/publicaciones/Requirements_Tools_Trade_Study1.doc.
- [8] Sitio web de CMMI, en www.sei.cmu.edu/cmmi.
- [9] Scholtes, PR, et al., The Team Handbook, 2.^a ed., Madison, WI: Oriel, Inc., 2001.

CAPÍTULO

7

Contenido

[Resumen](#)[Caso de estudio](#)[Referencias](#)

Las habilidades especiales de la RA

Los temas de este capítulo cubren un conjunto de habilidades especializadas que le ayudarán a desempeñar sus responsabilidades. Se han mencionado algunas habilidades recomendadas en la Tabla 3.1, Matriz de Habilidades de RA, pero aún no se han discutido en detalle. Además, hay áreas relativas al trabajo de la AR que merecen mayor elaboración. La tabla 7.1 proporciona una lista de estos temas. Es posible que descubras que no los necesitas todos en un momento particular de tu carrera o en un proyecto específico. Sin embargo, es probable que necesites conocer la mayoría de estos temas en algún momento de tu trabajo. Utilice este libro como guía de escritorio cuando surja la necesidad. También sugeriré otras referencias útiles que proporcionen más información para ayudar a responder las preguntas enumeradas en la Tabla 7.1.

Cada uno de estos temas se discutirá por turno.

1. ¿Por qué los errores de requisitos son tan devastadores y cómo pueden ayudar los RA? ¿abordar el problema?

La experiencia de la industria es que los errores asociados con los requisitos no sólo son el tipo de error más común en el desarrollo de sistemas y software, sino también los más costosos de encontrar y corregir. Los expertos de la industria continúan expresando su preocupación de que tanto el número de errores de requisitos como el costo de corregirlos aumentan geométricamente a medida que se descubre el error en el ciclo de vida. La tabla 7.2 proporciona algunos puntos de datos que ilustran por qué los errores de requisitos son tan devastadores.

¿Qué entendemos por error de requisitos? La Tabla 7.3 proporciona información sobre los tipos de errores de requisitos y su frecuencia relativa, proporcionada por la autora y experta de la industria Ivy Hooks en Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos [1].

No sorprende que las dos categorías más importantes sean los hechos incorrectos y la omisión de requisitos. Siguiendo el

Tabla 7.1 Temas de habilidades especiales de RA

Pregunta del tema	Número de página
1 ¿Por qué los errores de requisitos son tan devastadores y cómo pueden ayudar los RA a solucionarlos? ¿el problema?	127
2 ¿Qué necesita saber la RA sobre CM?	129
3 ¿Qué necesita saber la RA sobre el lenguaje de modelado unificado?	135
4 ¿Qué pasa si estoy apoyando un proyecto pequeño? ¿Algo de esto todavía se aplica? Cómo ¿Puedo convencer al primer ministro y a mis compañeros de trabajo para que incorporen cierto grado de disciplina? y proceso en nuestro enfoque?	139
5 ¿Cuál es la diferencia entre una especificación de requisitos y una especificación? ¿los requisitos?	141
6 Nota una "estimación de impacto" en la matriz de habilidades requeridas: ¿qué es? y ¿cómo puedo obtener más información al respecto?	142
7 Parece sugerir que la RA debería ser líder del proyecto. Por qué yo ¿Necesita ser un líder? ¿Cómo puedo ser un líder? ¿Qué debo liderar?	143
8 Ha subrayado el papel de la AR a la hora de facilitar debates, presentaciones, reuniones, sesiones de formación y talleres. ¿Qué puedo hacer para ser mejor? facilitador?	144
9 Ha enfatizado que tener un proceso de prevención de defectos (DP) es Aconsejable para todos los proyectos, quizás necesario. ¿Puede proporcionar un proceso de DP? que puedo implementar fácilmente?	145
10 Indicas que la estimación es una habilidad importante. ¿Qué aspectos de la estimación son críticos para la RA?	149
11 Aconseja realizar inspecciones de todos los documentos relacionados con los requisitos. Por qué ¿No deberíamos conformarnos con hacerles revisiones por pares? Cómo están inspecciones diferentes de las revisiones por pares, y ¿por qué tomarse tantas molestias? ¿Qué tipo de inspección es mejor?	150
12 Ha puesto mucho énfasis en la calidad. ¿Cómo puede ayudar la RA a solicitarla? ¿Principios de calidad en proyectos de desarrollo de sistemas y software?	152
13 Parece haber mucha confusión en nuestra industria con respecto a los términos verificación y validación. ¿Puedes explicar por qué esto es así y también aclarar? ¿Usos sugeridos de los dos términos?	153
14 Los "agilistas" defienden que las metodologías de desarrollo ágiles prometen mayores satisfacción del cliente, menores tasas de defectos, tiempos de desarrollo más rápidos y una solución a necesidades que cambian rápidamente. ¿Debo recomendar que ¿Considerar métodos de desarrollo ágiles en mi proyecto?	154
15 ¿Cuál es el valor del conocimiento práctico? deudas ¿Qué pasa si mi PM, el equipo directivo de nuestra organización o nuestro cliente no ¿No apoya el concepto de mejora de procesos?	154
17 ¿Cómo debe aplicarse la estructura de desglose del trabajo?	156
18 ¿Cuál es un buen enfoque para considerar los riesgos de los requisitos?	159

Enfoque recomendado para trabajar en colaboración con su cliente.
y usuarios para identificar los requisitos reales, utilizando la lista de verificación de recopilación
de requisitos proporcionada en la Tabla 5.1, habrá reducido el riesgo de que estos se produzcan.
tipos de errores. ¿Qué más pueden hacer los RA para reducir los errores de requisitos?

Tabla 7.2 Experiencia de la industria con respecto a errores de requisitos

Alcaparras Jones:

"Los errores que se originan en los requisitos tienden a ser los más costosos y problemático eliminar más tarde. La prevención es más eficaz que la eliminación de defectos".

Experiencia TRW:

"La mayoría de los errores se encontraron después de las pruebas unitarias, y más del 80% fueron requisitos y errores de diseño".

Tom DeMarco:

"Más de la mitad de todos los defectos se deben a errores de requisitos".

Fuerza Aérea de EE. UU.:

"El 41% de todos los errores descubiertos fueron errores de requisitos".

Tabla 7.3 Tipos de errores de requisitos y sus frecuencias relativas

Error de requisitos	Frecuencia relativa (%)
hecho incorrecto	49
Omisión	31
Inconsecuencia	13
Ambigüedad	5
Requisito 2 fuera de lugar	
Total	100

De: Hooks y Farry [1].

La Tabla 7.4 proporciona sugerencias aportadas por los profesionales.¹ Tenga en cuenta que varias de estas actividades pueden mitigar el riesgo de errores en múltiples categorías. Realizar este tipo de actividades tiene un alto retorno del tiempo invertido; son actividades "altamente apalancadas".

2. ¿Qué necesita saber la AR sobre CM?

CM es una disciplina clave en un proyecto de cualquier tamaño. Al carecer de CM eficaz, el El proyecto no está bajo control porque el estado de sus productos de trabajo no está bajo control. El control de configuración es una de las funciones críticas en una tarea o proyecto. que un RA debe conocer, ya que es vital para la integridad general de los requisitos en evolución y otros productos del trabajo de ingeniería. RA debe tomar un papel activo en la gestión de la línea base de requisitos desde su desde su inicio y garantizar que los cambios en la línea de base se identifiquen, documenten, revisen, aprueben y realicen un seguimiento durante la implementación. Estos Los controles y procesos de requisitos son muy similares a los procesos CM.

1. Con un agradecimiento especial a RA Pat Little, quien contribuyó con varias de estas ideas.

Tabla 7.4 Acciones sugeridas para reducir errores de requisitos

Supuestos:

Aproximadamente entre el 8% y el 14% de los costos totales del proyecto se invertirán en el proceso de requisitos.

Se brindará capacitación formal a los RA para explicar cómo redactar buenos requisitos y cómo abordar los diversos tipos de errores típicos en los requisitos.

El cliente estará involucrado durante todo el proceso de desarrollo.

Se establecerá y utilizará un glosario del proyecto para garantizar que se acuerden las definiciones y que las palabras se utilicen de manera consistente en todas las actividades del proyecto.

Tipos de errores de requisitos y sugerencias para abordarlos

Hechos incorrectos:

Proporcione un atributo en la herramienta de requisitos para verificar la base fáctica de cada requisito y realice investigaciones para validar los hechos. Por ejemplo:

V = verificado;

N = aún no investigado; Q =

cuestionable.

Proporcionar a las partes interesadas revisiones de los productos de trabajo de requisitos.

Requerir un enlace a una fuente autorizada (declaración de misión, documento de política, orientación formal, etc.). Si no se puede encontrar una fuente autorizada, escriba una justificación detallada del requisito. La referencia a una fuente autorizada es similar, pero más formal, a la idea de exigir una justificación para cada requisito.

Proporcionar un mecanismo (como una página web o un correo electrónico difundido) o múltiples mecanismos para una revisión amplia de los requisitos y la retroalimentación por parte de los usuarios, clientes, partes interesadas, etc.

Cree un modelo de datos lógico (LDM) desde una perspectiva empresarial.

Omisión:

Solicitar las necesidades del usuario desde una variedad de puntos de vista diferentes.

Realizar modelado de requisitos.

Utilice una metodología estructurada, como el modelado de casos de uso o una técnica comparable.

Desarrolle un OCD o CONOPS antes de profundizar en declaraciones de requisitos específicos.

Si el proceso de negocio que soportará el sistema está documentado, utilice la documentación del proceso para guiar a los usuarios a través del proceso de obtención de requisitos. Si el proceso empresarial no está documentado, considere trabajar con el usuario para crear esta documentación. Puede exceder los límites tradicionales del análisis de requisitos, pero al final tendrá una mejor comprensión de los requisitos y el usuario podrá haber mitigado uno o más riesgos de falla del sistema (incluso si el sistema cumplió con todos los requisitos documentados).

Inconsecuencia:

Realizar inspecciones de productos de trabajo relacionados con requisitos por parte de miembros del equipo del proyecto.

Examinar y analizar el RTM para verificar la coherencia y la ubicación adecuada de los requisitos.

Requerir un enlace a una fuente autorizada (declaración de misión, documento de política, orientación formal, etc.). Si no se puede encontrar una fuente autorizada, escriba una justificación detallada del requisito.

Defina todos los términos y asegúrese de que su uso en las declaraciones de requisitos sea coherente con la definición formalmente reconocida.

Cree un LDM desde una perspectiva empresarial.

Ambigüedad:

realizar inspecciones de productos de trabajo relacionados con requisitos por parte de miembros del equipo del proyecto.

Requerir un enlace a una fuente autorizada (declaración de misión, documento de política, orientación formal, etc.). Si no se puede encontrar una fuente autorizada, escriba una justificación detallada del requisito.

Defina todos los términos y asegúrese de que su uso en las declaraciones de requisitos sea coherente con la definición formalmente reconocida.

Requisito fuera de lugar:

examine y analice el RTM para verificar la coherencia y la ubicación adecuada de los requisitos.

utilizado por otros equipos de proyecto para controlar la evolución de sus productos de trabajo. En absoluto veces, la RA necesitará trabajar con las versiones actuales y correctas bajo Controlar CM y poder dar cuenta de cada cambio y versión de ese trabajo. producto. Por lo tanto, la AR debe tener conocimiento y experiencia con la siguiendo las prácticas de CM.

Planificación CM La planificación CM debe incluir consideraciones relativas a la siguiente: requisitos del contrato, alcance general, CM del cliente, procesos, herramientas y recursos. Desde la perspectiva de los requisitos, una RA necesitaría trabajar con los planificadores de CM para garantizar que los procesos de CM y RM estén coordinados y que los elementos de configuración incluyan requisitos, documentos y otros elementos identificados. Integración de políticas, procesos, procedimientos de CM.

y prácticas con los requisitos y otras políticas de ingeniería, Los procesos y métodos garantizarán un enfoque cohesivo y centrado en el equipo. Formar una estrecha relación de trabajo con todos los grupos del proyecto y con el El cliente fomenta relaciones laborales sólidas y de apoyo mutuo. Las comunicaciones, los procesos y las herramientas eficaces ayudarán a garantizar la integridad general de productos de trabajo entregados.

Herramientas CM El equipo de requisitos debe garantizar que las herramientas seleccionadas herramienta de requisitos no sólo sea adecuada para fines de control, sino que sea también bien integrado con herramientas CM. Algunas herramientas CM comerciales tienen interfaces automatizadas con herramientas de requisitos y otras herramientas de ingeniería. Ellos a su vez debe trabajar en conjunto con y en apoyo del proyecto general. proceso, incluidos los requisitos y CM. La Figura 7.1 describe las relaciones entre una herramienta RM y una herramienta CM en un proyecto de muestra. Tenga en cuenta la identificación de varios documentos y actividades en la herramienta de requisitos, que Los datos se envían directamente desde la herramienta de requisitos a la herramienta CM, que Los requisitos se asignan a los paquetes de software y cómo cada una de las herramientas ayuda con las actividades críticas del proyecto.

Líneas de base de ingeniería En proyectos de ingeniería, CM trabaja temprano en el ciclo de vida del proyecto con los otros grupos de proyecto para identificar productos de trabajo (identificados como elementos de configuración) e incorporarlos en procesos controlados. líneas de base. Los equipos de ingeniería (incluidos los RA) gestionan estas líneas de base como se mueven a través de un ciclo de vida de ingeniería establecido. Estas líneas de base probablemente tendrá identificaciones únicas (al igual que los elementos de configuración individuales) para que los gerentes, clientes, requisitos, diseño, ingenieros de pruebas, control de calidad, y CM tienen todos el mismo entendimiento con respecto al estado de referencia y Contenido, estado del elemento de configuración, versiones y cambios aprobados. incorporado.

Líneas base de operaciones Los proyectos de operaciones y mantenimiento (O&M) lo encuentran crítico para mantener un estado estable de un sistema operativo o grupo de sistemas. Las líneas de base también son importantes aquí, ya que incluyen hardware controlado, software (COTS desarrollado e integrado) y estándar. procedimientos operativos (POE).

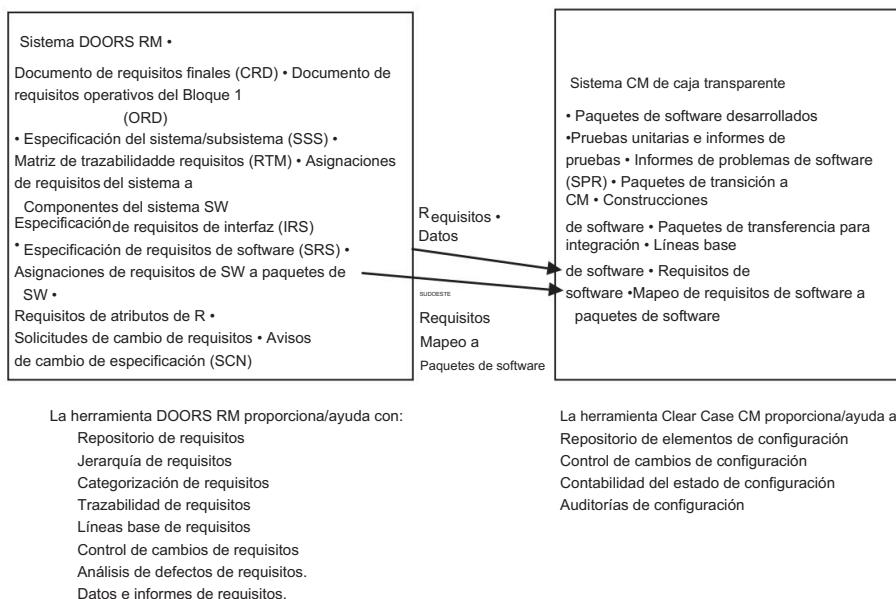


Figura 7.1 Relaciones entre la herramienta de requisitos de un proyecto y su herramienta CM.
(Fuente: Michael Davis. Usado con autorización).

Control de configuración El control proporciona un amplio espectro de actividades y métodos que garantizan que las configuraciones se gestionen adecuadamente y que tengan la mayor integridad posible. Estas actividades incluyen herramientas y repositorios en los que el trabajo se realiza de forma gestionada y la gestión de cambios en esos productos de trabajo. En términos de las tareas de la RA, lo más importante es el uso de una herramienta de requisitos en la gestión y control de los requisitos del proyecto. El RA debe trabajar con CM para garantizar que sus requisitos y otros productos de trabajo (por ejemplo, documentos de requisitos) estén en un entorno controlado (en una herramienta u otro espacio controlado), que estén disponibles para fines laborales y que estén respaldados o archivados adecuadamente y disponible para recreación si es necesario. Dentro de la herramienta o repositorio, el equipo de requisitos debe garantizar que los requisitos individuales sean rastreables (lo cual es en sí mismo una necesidad de CM necesaria para mostrar la contabilidad del estado general). Los requisitos deben tener trazabilidad hacia versiones anteriores del mismo requisito o ser rastreables hasta su origen. La herramienta seleccionada también debe proporcionar trazabilidad directa para mostrar las asignaciones de los requisitos a los componentes del programa (por ejemplo, hardware, software) y las pruebas a realizar, y debe ser capaz de identificar y recuperar versiones anteriores de los requisitos, basándose en cambios aprobados. La herramienta de requisitos también debe poder mostrar indicadores de cambio [por ejemplo, solicitudes de cambio (CR) y propuestas de cambio de ingeniería (ECP)] que proponen cambios a los requisitos. Esto se requiere más adelante en el ciclo de vida cuando CM realiza auditorías de configuración para verificar que los requisitos se hayan satisfecho (probado) a través de los produc-

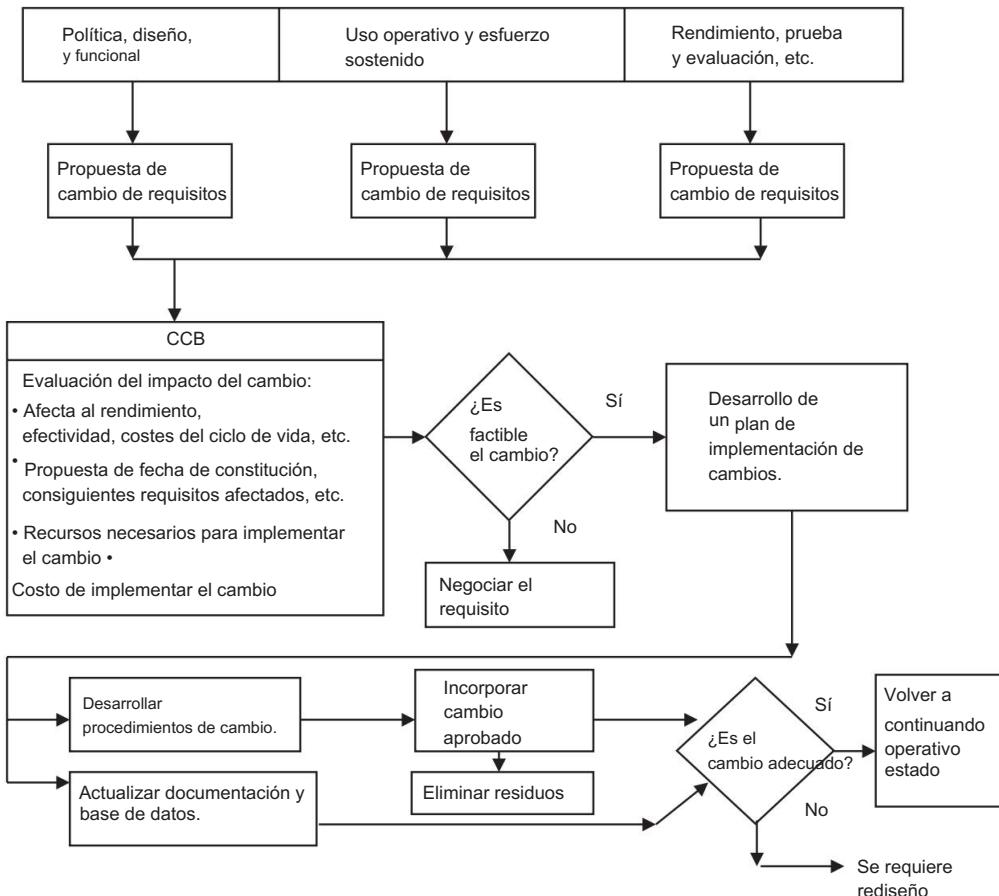


Figura 7.2 Procedimiento de control de cambios de requisitos. (Fuente: Michael Davis. Usado con autorización).

vista de alto nivel de algunas de las interrelaciones entre el CM y las actividades de requisitos en un proyecto, en este caso en relación con las propuestas de cambio de requisitos.²

Junta de Control de Configuración (CCB) Como se indica en la Figura 7.2, a menudo se convocará a la RA para que participe en las reuniones de la CCB y proporcione no sólo el estado, sino también asesoramiento de ingeniería experto y análisis de los impactos de los cambios solicitados. El CCB evalúa si el cambio es factible para garantizar que los cambios en los requisitos sean (1) desaprobados o (2) aprobados, coordinados e implementados de manera coordinada. Esto garantiza que todas las partes interesadas comprendan el cambio y sean capaces de realizar el trabajo necesario, así como la integridad de los productos de trabajo en evolución. Este enfoque de mantener el control de la configuración es esencial para proyectar éxito.

2. Nuestro agradecimiento al RA Michael Davis por brindarnos las experiencias de su proyecto, como se documenta en las Figuras 7.1 y 7.2.

Gestión de cambios operativos Todos los elementos mencionados anteriormente relacionados con las líneas de base operativas sufren cambios frecuentes. Estos cambios también se someten a un proceso de cambio utilizando un CCB para revisar y aprobar el cambio y un proceso de implementación que supervisa la implementación del cambio. Con respecto a estos cambios y la eficacia operativa general, el personal de operaciones también debe ser consciente de sus requisitos. Estos no son menos importantes que los que se encuentran en el entorno de la ingeniería. Deben mantener un nivel de soporte o servicio al cliente a través de acuerdos de nivel de servicio (SLA) documentados. Deben implementarse mediante la operación del sistema implementado y completamente implementado y deben ser monitoreados constantemente. Estos SLA se gestionan, al igual que los requisitos en un proyecto de ingeniería, mediante la recopilación de datos operativos (métricas) que se analizan, rastrean y elaboran informes. Los cambios en los SLA se gestionan a través de solicitudes de cambio y pueden implicar cambios de ingeniería a medida que cambian las necesidades del usuario.

Contabilidad del estado de configuración Aquellos involucrados en la gestión de requisitos y productos de trabajo de requisitos deben comprender sus responsabilidades con respecto al registro del estado de sus actividades. Por ejemplo, si se proponen cambios a los requisitos (después del proceso de cambio), la AR debe documentar o registrar el cambio, registrar su análisis de impacto y dar seguimiento a la implementación del cambio, registrando el estado de su implementación cuando se complete. Para la RA, esto implicaría actualizaciones de la herramienta de requisitos y de los productos de trabajo de requisitos afectados. Si se necesitan informes, el CM y la RA deben coordinar la información necesaria y la mejor manera de recopilar, formatear y entregar la información.

Auditorías de configuración Debido a que su trabajo es integral para la entrega de productos de trabajo que cumplan con los requisitos establecidos por el cliente, los RA deben participar en auditorías CM formales e informales para ayudar a verificar que los requisitos utilizados sean correctos, estén actualizados (que reflejen todos los cambios aprobados) y se hayan cumplido. puestos a disposición de los equipos de diseño, desarrollo y pruebas para su trabajo.

Métricas de CM A menudo, a la oficina de CM se le puede dar la responsabilidad de gestionar las métricas del programa, o al menos proporcionar información al programa de métricas del proyecto. Algunas métricas pueden involucrar requisitos directamente (como en la medición de la volatilidad o el cambio de los requisitos), o pueden tener información sobre las métricas recopiladas para cambios generales.

El RA sabio y proactivo formará una relación sólida y de apoyo mutuo con el equipo de CM. Le sugiero que, si aún no está muy familiarizado con CM, busque a alguien que sí lo esté y pídale ayuda para familiarizarse con las actividades de CM. Además, estudie un buen libro de CM, como el Manual de control de documentación de ingeniería de Frank B. Watts: Gestión de configuración para la industria [2]. Watts se centra en un sistema CM rápido y simplificado que supera los estándares del Departamento de Defensa (DoD) de EE. UU. Se proporcionan estudios de casos. Además, un libro que muchos profesionales de CM de software consideran de lectura obligada es la Guía para la configuración de software de Alexis Leon.

Gestión [3]. Un ejemplo de un buen plan CM está disponible en www.air-time.co.uk/users/wysiwig/cmp.htm. Este enlace también proporciona otro enlace a una breve descripción de los principios de la gestión del cambio. Una discusión más detallada sobre la gestión técnica está disponible en sparc.airtime.co.uk/users/wysiwig/techman.htm.

Véase también la Norma EIA 649, "Estándar de consenso nacional para la configuración". Gestión de la información" [4].

3. ¿Qué necesita saber la RA sobre UML?

La respuesta corta es mucho o poco, dependiendo de si su proyecto utiliza un enfoque OO. El Lenguaje Unificado de Modelado (UML) es una notación de propósito general desarrollada en la década de 1990 queDescribe el comportamiento estático y dinámico de un sistema. Es un modelado visual. lenguaje y no pretende ser un lenguaje de programación visual en el sensación de tener todo el soporte visual y semántico necesario para reemplazar los lenguajes de programación. Por ejemplo, las ramas y uniones complejas son las mejores expresado en un lenguaje de programación textual. UML proporciona un modelo formal y completo para documentar un sistema. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un software.

sistema intensivo. Grady Booch, Ivar Jacobson y Jim Rumbaugh de Rational Corporation inició el UML y fueron sus principales metodólogos originales. Consulte UML destilado de Fowler : aplicación del modelado de objetos estándar Idioma [5] para una buena referencia. Los productos posteriores fueron un esfuerzo de equipo de muchos socios bajo el patrocinio de Object Management Group (OMG), una organización internacional que incluye proveedores de sistemas de información, desarrolladores y usuarios de software. Una "Introducción al UML de OMG" puede descargarse de la Web [6]. La versión actual del UML es también descargable desde la Web [7]. UML se ha convertido en un estándar independiente del proveedor para expresar el diseño de sistemas de software. El UML incorpora casos de uso como medio estándar para capturar y representar requisitos. Los casos de uso son esencialmente funcionales y los RA los utilizan ampliamente para enfoques OO y no OO. Algunos expertos de la industria, incluido Karl Wiegers, creen que los casos de uso se encuentran entre los más efectivos técnicas para capturar requisitos; otros sienten que los casos de uso por sí solos no Proporcionar información adecuada para que los desarrolladores sepan qué construir. El Las siguientes son preocupaciones en el uso de casos de uso: (1) mucha experiencia con ellos. se requiere adquirir familiaridad; (2) es difícil saber cuándo se ha terminado de expresar los requisitos; se pueden seguir haciendo muchos análisis sin terminar; (3) se puede proporcionar información en los casos de uso en el nivel equivocado de abstracción; (4) los casos de uso suelen ser demasiado detallados y casi siempre hay que rehacerlo; y (5) los casos de uso suelen estar llenos de errores. Véase Gottesdiener [8], que cree haber encontrado casi todo tipo de Error que la gente puede cometer al escribir casos de uso. Es interesante notar que los roles en los casos de uso encajan bien con la noción de puntos de vista como vehículo para la obtención de requisitos. Si un grupo de desarrollo de requisitos ha tenido mucho experiencia en el uso de casos de uso, es posible que el equipo desee considerar usarlos para

ayudar a capturar los requisitos. De lo contrario, utilice un enfoque de especificación de requisitos para ayudar a representar los requisitos. Es posible que los ingenieros de sistemas quieran utilizar el estándar IEEE 1233, "Guía IEEE para desarrollar especificaciones de requisitos del sistema" [9], mientras que los ingenieros de software pueden preferir el estándar IEEE 830, "Práctica recomendada IEEE para especificaciones de requisitos de software" [10]. Otra plantilla de especificación de requisitos detallada, la Plantilla de especificación de requisitos de Volere [11], está disponible en línea en www.atlsysguild.com/GuildSite/Robs/Template.html. Se proporcionan varios ejemplos de requisitos.

Quienes opten por utilizar UML deberán familiarizarse con la Arquitectura de gestión de objetos (OMA), que proporciona la infraestructura conceptual en la que se basan todas las especificaciones OMG, con la semántica UML, la Guía de notación UML y los perfiles estándar UML. De esta discusión debería resultar evidente que se necesita mucho estudio y experiencia para utilizar UML de manera efectiva. Por otro lado, es esencial desarrollar un modelo para un sistema de software de potencia industrial. Se necesitan buenos modelos para facilitar la comunicación en todo el equipo del proyecto y garantizar que la arquitectura del sistema sea sólida. La importancia de utilizar buenas técnicas de modelado aumenta con la complejidad del sistema. Los creadores de UML creen que antes de UML, no existía un lenguaje de modelado líder claro. El costo de usar y soportar lenguajes de modelado con diferencias menores motivó los esfuerzos de representantes de más de dos docenas de empresas para asociarse para desarrollar UML. El UML no es un proceso. La experiencia de los desarrolladores de UML es que UML debe aplicarse en el contexto de un proceso, pero diferentes organizaciones y dominios de problemas requieren procesos diferentes. Los desarrolladores de UML reconocieron el valor de un proceso incremental, iterativo, centrado en la arquitectura y basado en casos de uso.

Kulak y Guiney explican los diagramas utilizados en UML en Casos de uso: requisitos en contexto [12]. Asegúrese de analizar su lista detallada de problemas relacionados con el uso de casos de uso [12, págs. 154–165] antes de tomar la decisión de confiar en los casos de uso para representar sus requisitos. Otras buenas referencias son Writing Effective Use Cases [13] de Cockburn (pronunciado "co-burn") y The Misuse of Use Cases [14] de Korson . Creo que el consejo de Korson es sabio: "antes de considerar casos de uso, asegúrese de que exista un proceso de requisitos bien comprendido y gestionado y de que tenga requisitos de alta calidad".³ Como se sugirió anteriormente en este libro, invertir en estas áreas El primero es un requisito previo para un buen enfoque de requisitos.

Un colega con experiencia en el uso de tecnologías OO aboga por el uso de UML en trabajos relacionados con requisitos. Proporcionó un análisis basado en su experiencia; consulte el recuadro 7.1.4. Esta viñeta proporciona ideas y sugerencias.

3. "El uso indebido de casos de uso" y "Construcción de casos de uso útiles" están disponibles en www.korson-mcgregor.com.

Korson también proporciona una plantilla que enumera y describe un conjunto de campos necesarios para un caso de uso bien escrito. Korson cree que los principales problemas en el desarrollo de sistemas no son técnicos sino que tienen que ver con los requisitos: "obtener los requisitos correctos y obtener los requisitos correctos".

4. Nuestro agradecimiento a Wayne O'Brien de Raytheon Corporation, quien amablemente accedió a prestarnos su experiencia.

Inserto 7.1—Experiencia en el uso de UML y casos de uso

Los casos de uso, con las herramientas automatizadas disponibles para admitir UML, proporcionan un vínculo semántico activo desde la vista más detallada de un sistema (por ejemplo, código de software o lógica de hardware, hasta los requisitos de nivel superior, que se capturan en los casos de uso). Este vínculo activo significa que hay información para comprender el sistema que está actualizada y accesible en todo momento para todas las partes interesadas.

Los usuarios interesados incluyen no sólo a los operadores, sino también a los gerentes, inversores, auditores y otros. Las partes interesadas de los desarrolladores incluyen cualquier disciplina que sea necesaria para el sistema bajo consideración, tanto temporal como funcionalmente. Temporalmente, el vínculo semántico activo respalda las disciplinas del desarrollador durante todo el ciclo de vida de un sistema, incluidos los RA, los diseñadores y los evaluadores. El enlace activo también admite disciplinas de desarrollo funcionales, incluidos ingenieros de hardware, software, comunicaciones y factores humanos.

Este amplio apoyo a las partes interesadas que proporcionan los casos de uso depende de las características de UML, así como de las herramientas que soportan UML. Brindar a diversas partes interesadas acceso a información precisa sobre el sistema de su interés aprovecha el aspecto de vista múltiple de UML (una mejor práctica descrita por el estándar IEEE 1471). El concepto de caso de uso dentro de UML captura la semántica de un sistema de una manera que se comunica fácilmente a personas con una amplia gama de intereses y experiencia, en lugar de solo habilidades en un solo campo. Es lo suficientemente preciso como para ser útil para construir sistemas y, sin embargo, fácilmente comprendido por personas que no se ganan la vida construyendo sistemas. Estas mismas comunicaciones características de los casos de uso subyacen en el apoyo interdisciplinario temporal y funcional para los desarrolladores, que se ganan la vida construyendo sistemas.

Wayne O'Brien
Corporación Raytheon
enero de 2003

Tuve la oportunidad de escuchar a Grady Booch (como se señaló anteriormente, uno de los tres principales metodólogos de UML) hablar en una sesión patrocinada por Rational Corporation en el norte de Virginia el 22 de noviembre de 2002. Entre las ideas ofrecidas por Booch se encuentran las siguientes:

Los clientes no saben lo que quieren hasta que se lo entregan. Esto aboga por un enfoque incremental, el valor de la iteración y el enfoque de mostrar productos y planes de trabajo a los usuarios de forma incremental.

Un enfoque incremental nos permite hacer correcciones a mitad de camino.

Existe una gran brecha entre dónde estamos y dónde debemos estar. Un buen enfoque es observar “dónde duelen las cosas” y atacar los puntos dolorosos.

La principal barrera a la que nos enfrentamos hoy es la capacidad de desarrollar software.

Invertir en el proceso de desarrollo de software, porque el software es el alma. El desarrollo de software todavía requiere mucha mano de obra y esto no cambiará. Los mejores procesos se vuelven invisibles: es simplemente "la forma en que hacemos negocios".

La mayoría de las empresas se encuentran en el "nivel 1½" de CMM . El CMM es ortogonal al éxito del proyecto. No es lo suficientemente ágil para el éxito económico.

Hay modos de falla comunes en extremos opuestos del espectro del proceso: en un extremo, tenemos sobreingeniería y sobrediseño de sistemas; en el otro extremo, tenemos la agilidad. (Mantener la magia de un grupo talentoso es difícil porque los desarrolladores no quieren mantener ni crear otra versión. La magia ocurre cuando hay sinergia entre ejecutivos y desarrolladores).

La arquitectura es importante y será cada vez más importante, por ejemplo, los portales que cruzan barreras y silos de información. No existe un diseño perfecto. "Perfecto" es enemigo de "suficientemente bueno".

Se acerca el lenguaje empresarial universal (UBL); permite a otro nivel de abstracción.

Existen varias tendencias globales relativas al futuro del software:

1. Crecimiento y complejidad crecientes.
 2. Subida de andenes. Necesitamos patrones. Lea Patrones de diseño [15] para familiarizarse con el vocabulario. Podemos construir sistemas rápidamente utilizando patrones y marcos; aquí es donde las empresas obtienen su ventaja competitiva.
 3. Crecimiento de los servicios web.
 4. Aumento del software de código abierto (Linux no es un buen modelo para desarrollar sistemas).
 5. Crecimiento del desarrollo impulsado por modelos (MDD). Esto proporcionará la capacidad de depurar a un nivel superior.
 6. Programación orientada a aspectos (observar un sistema desde el aspecto de una parte interesada). Para esto faltan entre tres y cinco años. Se utiliza una herramienta (Aspecto J) para entrelazar los aspectos.
 7. Seguridad. Ésta es un área creciente y emergente y una cuestión general. Se necesitan procesos y tecnologías para ayudar con la seguridad.
 8. Entornos de desarrollo colaborativo y uso de la Web para espacios virtuales de proyectos (la encarnación de la memoria tribal).
 9. Desarrollo basado en componentes.5
5. El proceso de requisitos para el desarrollo basado en componentes (basado en el desarrollo con reutilización) es diferente para sistemas desarrollados desde cero.

El resumen de Booch fue un testimonio de la magnitud del hombre:

Sólo hay catorce millones de programadores en el mundo. ¡Mira lo que hemos hecho! Hemos cambiado el mundo. Qué privilegio es ser parte de la comunidad que ha marcado una gran diferencia. Qué responsabilidad es hacer un mejor trabajo todo el tiempo. Ustedes son las personas que hacen el trabajo pesado.

UML es, sin duda, una excelente metodología para derivar requisitos. Sin embargo, en el momento de escribir este artículo, es relativamente nuevo en el campo del desarrollo de requisitos. Una metodología que existe desde hace más de quince años es la Definición de integración para el modelado de funciones (IDEF). Esto comenzó con el método IDEF0 inicial y continúa hasta IDEF5. Insert 7.2 contiene experiencia del mundo real comparando UML con IDEF0. Estas experiencias fueron proporcionadas por un RA con más de veinte años de experiencia en desarrollo de software e ingeniería de requisitos. Es posible que la AR desee familiarizarse con IDEF0 como alternativa al UML. Excelentes referencias sobre IDEF0 son FIPS PUB 183 [16], el estándar IEEE para lenguaje de modelado funcional [17] y la Guía práctica para la reingeniería de procesos de negocios utilizando IDEF0 de Feldmann [18]. También existen algunas herramientas de simulación de flujo de trabajo IDEF que pueden resultar útiles para derivar requisitos, por ejemplo Business Modeling Workbench.

4. ¿Qué pasa si apoyo un proyecto pequeño? ¿Algo de esto todavía se aplica? ¿Cómo puedo convencer al primer ministro y a mis compañeros de trabajo para que incorporen cierto grado de disciplina y proceso a nuestro enfoque?

¡Buena pregunta! Me han planteado estas preguntas muchas veces y también hay información sobre ellas en la literatura. Primero, comencemos con una definición de proyecto pequeño. Entendiendo que no hay acuerdo sobre la definición de proyecto pequeño, mi definición arbitraria es de uno a seis profesionales trabajando en una sola tarea durante un período de entre tres y seis meses.

Algunos analistas e ingenieros pueden sentir que el CMM (y el más reciente CMMI) son aplicables sólo para proyectos medianos y grandes. Mi experiencia sugiere que CMM y CMMI son escalables a un proyecto de cualquier tamaño: es necesario realizar planificación y seguimiento de proyectos, desarrollo y gestión de requisitos, CM, control de calidad y otros procesos como DP en todos los proyectos.

Mi consejo es instar a los miembros de proyectos pequeños a reducir el enfoque y adaptarlo. Trate de no permitir que dichos proyectos utilicen la pequeñez como excusa para no utilizar procesos, mecanismos, métodos, técnicas y herramientas que les permitan tener éxito. Existen varias buenas publicaciones que brindan consejos sobre cómo implementar procesos en proyectos pequeños. Consulte Uso del software CMM con criterio: pequeños proyectos y pequeñas organizaciones [19] de Paulk para obtener más información sobre este tema y "¿Qué tan escalables son las prácticas clave de CMM?" de Rita Hadden. [20]. La opinión de Hadden, basada en observaciones y experiencia con más de 50 proyectos pequeños, es que el juicio profesional debe

Inserto 7.2—Algunas experiencias del mundo real sobre la IDEF

Creo que en lo que respecta al análisis de requisitos, la preferencia del cliente y el nivel de sofisticación deberían desempeñar un papel importante a la hora de seleccionar qué metodología de modelado utilizar. UML es precisamente el enfoque favorito que se utiliza actualmente en el tema general del modelado. Un analista con el mismo nivel de comprensión de IDEF0/IDEF1X que de UML sería casi igual de eficaz con cualquiera de los dos. Ambos brindan información sobre el espacio del problema, ayudan al analista a saber qué preguntas hacer y brindan una imagen de la funcionalidad del sistema para respaldar el texto de requisitos. (Creo que el análisis de puntos de función (FPA) también lo lleva allí por medios diferentes y menos gráficos).

IDEF es más fácil de comprender que UML. En dos grandes proyectos que apoyé, comenzamos creando requisitos en UML. En ambos casos, alguien se echó atrás al final y decidió que algún grupo importante de partes interesadas del lado del cliente nunca entendería la presentación UML de los requisitos, y volvimos a presentar los requisitos como texto plano. Irónicamente, en ambos casos los clientes ya estaban familiarizados con los modelos IDEF.

Esto fue claramente culpa nuestra. Si va a utilizar modelado de cualquier tipo, asegúrese de que los clientes comprendan adecuadamente el enfoque de modelado. Si no lo hacen, entonces es tu obligación enseñarles, y cuanto antes mejor.

En el mundo del Departamento de Defensa, este problema se ha resuelto principalmente con IDEF, suponiendo que se siga el enfoque descrito en el Apéndice C de FIPS 183. El equipo conjunto está formado por clientes (que generalmente son lectores de IDEF) y contratistas (que generalmente son redactores de IDEF). Los kits (paquetes de revisión IDEF) se revisan tempranamente y con frecuencia. Cuando el modelo está casi terminado, a nadie le sorprende su apariencia o lo que describe.

Por mucho que me guste el modelado como herramienta, no puedo enfatizar lo suficiente que la metodología de modelado debe ser comprendida por todos, no sólo por unos pocos especialistas, para tener éxito. Creo que esta es la causa fundamental de todos los esfuerzos fallidos de modelización, independientemente de la metodología. Después de todo, cada uno de nosotros ha pasado años comunicándose a través de un lenguaje hablado común, pero todavía no logramos entendernos la mayor parte del tiempo. ¿Por qué esperamos que alguien que tiene poca o ninguna experiencia previa en comunicación con modelos los comprenda?

Terry L. Bartolomé

utilizarse para reducir la escala y aplicar prácticas clave para apoyar proyectos pequeños. La CMM personalizada LOGOS de Brodman y Johnson para pequeñas empresas, pequeñas organizaciones y pequeños proyectos [21] también puede resultar útil.

Entonces mi respuesta es sí, esto se aplica a proyectos pequeños: adapte los procesos, mecanismos, métodos, técnicas y herramientas según sea necesario para el proyecto pequeño en su entorno. Convencer a su primer ministro y a sus compañeros de trabajo para que incorporen cierto grado de disciplina y proceso puede resultar una tarea difícil o imposible si tienen opiniones firmes. Mi sugerencia aquí es evitar discusiones; más bien, (1) abordar un “punto problemático” en el proyecto (un área donde el proyecto está experimentando dificultades) y (2) utilizar datos para respaldar su posición. Comenzar poco a poco con una tarea pequeña puede tener beneficios inmediatos para los clientes y servir como una forma de convencer a la gerencia y al personal de la utilidad de un enfoque centrado en los procesos. Esta área bien podría ser la obtención de requisitos: el proyecto puede estar experimentando algunos problemas para identificar los requisitos reales. Los clientes y usuarios pueden no diferenciar entre los requisitos declarados y los requisitos reales. Este es un excelente lugar para comenzar, porque con un poco de esfuerzo en comprender y analizar los requisitos establecidos, es muy probable que pueda identificar formas de ayudar; por ejemplo, aplicando los criterios de un buen requisito a los requisitos establecidos; investigar e identificar la justificación de cada requisito; colaborar con el cliente y los usuarios para priorizar los requisitos en un entorno de taller de requisitos, facilitado por un externo experimentado; crear un equipo conjunto para discutir los requisitos y tomar decisiones compartidas sobre ellos; proporcionar alguna capacitación relacionada con los requisitos basada en los temas sugeridos en la Tabla 5.5; sugerir el uso de una herramienta de requisitos automatizada como Requisite Pro (ReqPro) para asignar atributos a cada requisito (por ejemplo, prioridad) y realizar la trazabilidad de los requisitos para garantizar que cada requisito se aborde en todo el sistema; las posibilidades son infinitas porque las oportunidades de continuidad ¡Abundan las mejoras! Intenta posicionarte como un asesor de confianza. Utilice datos para demostrar su punto e identificar otros puntos débiles. Revise las funciones del AR analizadas en el Capítulo 2. Considere cuál de esas funciones le parece más apropiada para la situación particular. Ser una influencia positiva en el proyecto. Encuentre formas de ayudar y contribuya con más de lo que le corresponde. Tendrá éxito y se convertirá

5. ¿Cuál es la diferencia entre una especificación de requisitos y especificar los requisitos?

¡Gran pregunta! Esto es causa de mucha confusión y es importante que la AR pueda aclarar ésta y otras áreas, no con una actitud de sabelotodo, sino con humildad y comprensión de que es parte de la naturaleza humana que las personas tener diferentes puntos de vista y definiciones de las cosas y diferentes opiniones. Mencioné la especificación de requisitos anteriormente en este capítulo y proporcioné dos buenas referencias al respecto. Una especificación de requisitos es un documento que contiene requisitos. Recordará que en el Capítulo 5 le recomiendo que considere que el repositorio de requisitos consta de varios elementos: la especificación de requisitos (o especificaciones si está desarrollando un sistema grande) es un elemento importante. Algunas personas usarán la verborrea “especificar los requisitos del sistema” para significar algo.

otra cosa, es decir, determinar cuáles deberían ser los requisitos del sistema. Recomiendo que se evite la frase "especificar los requisitos" para reducir la confusión. Una mejor descripción sería definir o documentar los requisitos del sistema.

6. Noto una "estimación de impacto" en la matriz de habilidades requeridas. ¿Qué es y ¿Cómo puedo obtener más información al respecto?

La estimación de impacto (IE) es una técnica analítica defendida por la industria experto y consultor Tom Gilb. Gilb le dedica el capítulo 11 de su libro. Principios de gestión de ingeniería de software [22]. Gilb cree que el IE La técnica proporciona una mejor manera de evaluar el proceso de diseño que cualquier otra. otra técnica. Como se muestra en la Figura 7.3, IE proporciona una herramienta analítica para cuantificar los requerimientos.

Esto permite a las RA evaluar los requisitos cuantitativamente en lugar de sólo subjetivamente. La técnica estima el impacto de las estrategias seleccionadas sobre objetivos específicos. Las tablas IE permiten el análisis de aspectos técnicos u organizativos. ideas en relación con requisitos y costos. Gilb informa [23] que tiene Usé IE para hacer lo siguiente:

Comparar ideas de diseño alternativas;

Estimar el estado de la arquitectura de diseño general;

Soluciones											
Requisitos											
Rentabilidad	20% de retorno de la inversión	5	10	10	10	20	20	75 – 25	Bajo diseño		
Usabilidad	30 minutos.	30	60	25	20	20	40	195	95	Sobre diseño	
Conectividad	5 minutos.	10	40	0	40	40	40	170	70	Sobre diseño	
Disponibilidad	99,98%	30	20	30	10	20	20	130	30	sobre diseño	
Integridad	99,99%	10	10	20	5	20-10		55 – 45	Bajo diseño		
Rendimiento >12 Tr/S		30	5	20	5	5	5	70 – 30	Bajo diseño		
Comerciabilidad 12 idiomas		40	40	40	20	30	40	210	110	Sobre diseño	
Adaptabilidad	10 años.	30	30	10	10	30	30	140	40	Sobre diseño	
Desarrollo, recursos	\$12 millones	20	30	30	10	30	30	150	50	Por encima del presupuesto	
El marketing cuesta 1 millón de dólares al año.		5	10	20	10	5	15	65 – 35	Por debajo del presupuesto		
Relación valor/coste		1,85	1,34	0,78	1,5		1,32	1,03			
Decisión		Mejor solución	No mejor								

Figura 7.3 Un ejemplo de uso de la técnica IE para evaluar requisitos. (Fuente: Tom Gilb. Reimpreso con autorización).

Planificar y controlar los pasos de entrega del proyecto;

Analizar riesgos.

Gilb reconoce que IE está “lleno de posibilidades de errores” (porque el analista proporciona las estimaciones basándose en un enfoque de mejor estimación).

Él cree que estos errores son inevitables y que la técnica IE debería usarse para calcular el “orden de magnitud aproximadamente correcto” [22, p. 184]. Si necesita utilizar esta técnica, le sugiero que estudie el capítulo 11 del libro de Gilb y lea más sobre la técnica en el sitio web de Gilb, www.gilb.com.

7. Parece sugerir que la AR debería ser líder del proyecto. ¿Por qué necesito
¿Sé un líder? ¿Cómo puedo ser un líder? ¿Qué debo liderar?

Aunque el papel de líder no se identificó específicamente en el Capítulo 2, el tono de este libro sugiere que la AR debería aprovechar la oportunidad para brindar liderazgo en los proyectos que usted apoya. El cambio es endémico en los entornos laborales actuales y se necesitan líderes que ayuden a gestionar los cambios y faciliten la adaptación a ellos. Es posible que sienta que demostrar cualidades de líder está “por encima de su nivel salarial”; el propósito de esta discusión es convencerlo de que el AR puede y debe ayudar a liderar el proyecto. El cuadro 7.5 sugiere algunas de las formas en que la AR puede ayudar a liderar.

Tabla 7.5 Algunas de las formas en que las AR pueden ayudar a liderar

Instar a la identificación de las necesidades reales;

Instar a la implementación de un mecanismo para controlar los cambios de requisitos y nuevas requisitos;

Sugiera el uso de un conjunto de reglas de conducta en su proyecto;

Sugerir el desarrollo de un conjunto de pautas para reuniones efectivas para usar en su proyecto;

Recomendar el uso de revisiones por pares de todos los productos de trabajo e inspecciones de productos de trabajo relacionados con requisitos;

Sugerir implementar y utilizar un proceso de PD;

Proporcionar capacitación sobre requisitos para los RA, el personal del proyecto y los clientes y usuarios, incluidas explicaciones de los problemas de la industria relacionados con los requisitos y sus causas;

Instar a tomar medidas proactivas para mejorar la comunicación del proyecto, como presentaciones periódicas de cada grupo de trabajo (para todos los demás en el proyecto) sobre las actividades de trabajo actuales;

Instar a adaptar el enfoque de recopilación de requisitos para su proyecto;

Recomendar el uso de una herramienta de requisitos automatizada sólida en la industria en el proyecto y capacitación formal para quienes usan la herramienta con mayor frecuencia;

Evaluando sus requisitos según los criterios de un buen requisito y la justificación de cada requisito; comparta los resultados de este ejercicio con sus clientes/usuarios.

Estas son sólo algunas de las formas en que un RA puede ayudar a liderar: las posibilidades son infinitas. Todos los días surgen oportunidades para que cada uno de nosotros lidere. Dé el ejemplo y ayude a fomentar formas ganadoras en su proyecto.

Algunas personas sienten que no todo el mundo está hecho para ser líder y que el RA no necesita ser un líder, sino más bien un buen jugador de equipo, dispuesto a compartir sus conocimientos y habilidades para resolver problemas.

8. Ha destacado el papel de la AR a la hora de facilitar debates, presentaciones, reuniones, sesiones de formación y talleres. ¿Qué puedo hacer para convertirme en un mejor facilitador?

Mi consejo para usted con respecto a esto es "simplemente hágalo". He descubierto que cuantas más presentaciones hago durante mi carrera, más fácil se vuelve hacerlas. Todavía me pongo nervioso antes de una presentación, pero está bien. Utilizo esa energía nerviosa para prepararme para la presentación y estar un poco animado durante la presentación. Ninguno de nosotros es más feliz durante una presentación aburrida. Ser optimista, entusiasta, moverse entre la audiencia, invitar a la participación, reconocer los buenos comentarios y contribuciones: todo esto tiene el efecto de involucrar a los participantes y, en mi experiencia, resultan en presentaciones que son bien recibidas. Cuando realmente te paras a pensar en hacer una presentación, ¿a qué temer? ¿Qué es lo peor que puede pasar?

Es posible que algunas personas en la audiencia no tengan una buena opinión de sus comentarios, pero este es el caso incluso cuando oradores experimentados suben al podio: no todos se identificarán con las opiniones y presentaciones de cada orador. He dado cientos de presentaciones y nunca ha habido una en la que al menos una persona del público no haya hecho algún comentario amable y expresión de interés.

Hay algunos pasos que puedes seguir para ayudar:

Asiste a presentaciones de otros y discierne lo que te gusta (y lo que no).
me gusta) sobre ellos.

Forma equipo con alguien que consideres más competente y experimentado en hacer presentaciones o facilitar talleres para estar frente a una audiencia y mejorar tus habilidades y confianza.

Haga arreglos para que lo graben en video cuando haga una presentación. Ver la cinta. Quizás te sorprenda lo que veas. Mi experiencia, después de haber hecho esto varias veces, es que soy bastante diferente de lo que pensé que sería.

Buscar oportunidades para dar charlas. Por ejemplo, las secciones locales de asociaciones profesionales siempre están buscando oradores para almuerzos, reuniones nocturnas y tutorías los sábados. Además, considere hacer una presentación en reuniones y conferencias anuales. Creo que descubrirás que cuantas más presentaciones hagas, más fácil te resultará hacerlo. Incluso podrías organizar sesiones informales en tu oficina y dar algunas de las charlas tú mismo.

Asista a sesiones de capacitación sobre cómo realizar presentaciones o únase a un grupo, como Toastmasters, que brinde apoyo, oportunidades y aliento. No adoptes la actitud de "¡No puedo hacer eso!" Como en otras áreas de la vida, cada uno de nosotros puede hacer mucho más de lo que creemos. Sólo estamos limitados por nuestra propia voluntad de salir y hacer cosas. ¡A por ello!

9. Ha enfatizado que tener un proceso de PD es aconsejable para todos los proyectos, tal vez necesario. ¿Puede proporcionarme un proceso de DP que pueda implementar fácilmente?

DP se preocupa por aprender de los defectos encontrados en los productos de trabajo e implementar formas de mejorar los procesos para evitar que los defectos vuelvan a ocurrir. Es un proceso valioso porque puede reducir la cantidad de errores que se cometan al principio del proceso de desarrollo, ahorrando así retrabajo (tiempo y dinero). Por supuesto, DP se puede utilizar en cualquier momento de cualquier proyecto para mejorar los resultados de nuestras actividades laborales. Un proceso del PD se adapta, documenta, enseña, aprende y aplica fácilmente. Lamentablemente, la mayoría de las organizaciones no se esfuerzan por implementar un proceso de PD. Creo que una de las razones por las que los procesos de DP no se implementan ni se utilizan con mayor frecuencia es que las personas tienen tantas otras cosas que hacer que no quieren asumir otro proceso y actividad más. Otra razón es que muchas personas no aprecian el poder y el valor de las actividades del PD6. Tenga en cuenta que lo principal es que el proyecto tenga éxito. Nuestros criterios de éxito incluyen, entre otros, completar el proyecto dentro del presupuesto y a tiempo. DP puede hacer una valiosa contribución para lograr estos objetivos.

Un proceso de DP La Figura 7.4 proporciona un proceso de DP sencillo con el que puede comenzar y luego agregar características adicionales según lo considere necesario, por ejemplo, métricas para evaluar la efectividad del proceso, un plan de DP, un repositorio de DP, etc. [Tenga en cuenta que a DP se le ha dado otro nombre en el CMMI : "Análisis y resolución causal" (CAR).]

El propósito de DP es identificar las causas fundamentales de los defectos y otros problemas. problemas y tomar medidas para evitar que vuelvan a ocurrir en el futuro.

El primer paso es identificar un producto o proceso de trabajo que esté creando problemas. ¿Dónde está experimentando problemas el proyecto ("puntos débiles")? Por ejemplo, podría decidir que el proyecto está experimentando algunos problemas debido a problemas relacionados con los requisitos. Por lo tanto, tal vez una buena actividad de mejora podría ser evaluar el proceso de requisitos del proyecto. Si está documentado, genial, deja que el proyecto quede documentado.

6. DP es un proceso de alta madurez en el CMMI . Esto sugiere al observador casual que no deberíamos preocuparnos hasta que hayamos abordado todos los procesos en los niveles 1 a 4 de los modelos CMMI /CMMI . Mi experiencia es que DP debería usarse en la mayoría de los proyectos, sin importar el nivel de madurez. La razón es la misma que mi sugerencia anterior de utilizar revisiones por pares en todos los proyectos: estas actividades identifican problemas y defectos en una etapa más temprana del proceso de desarrollo de lo que sería el caso si no usáramos estas técnicas. Como resultado, se ahorra tiempo y dinero, y la calidad de los productos del trabajo es mejor de lo que habría sido si no los usáramos.

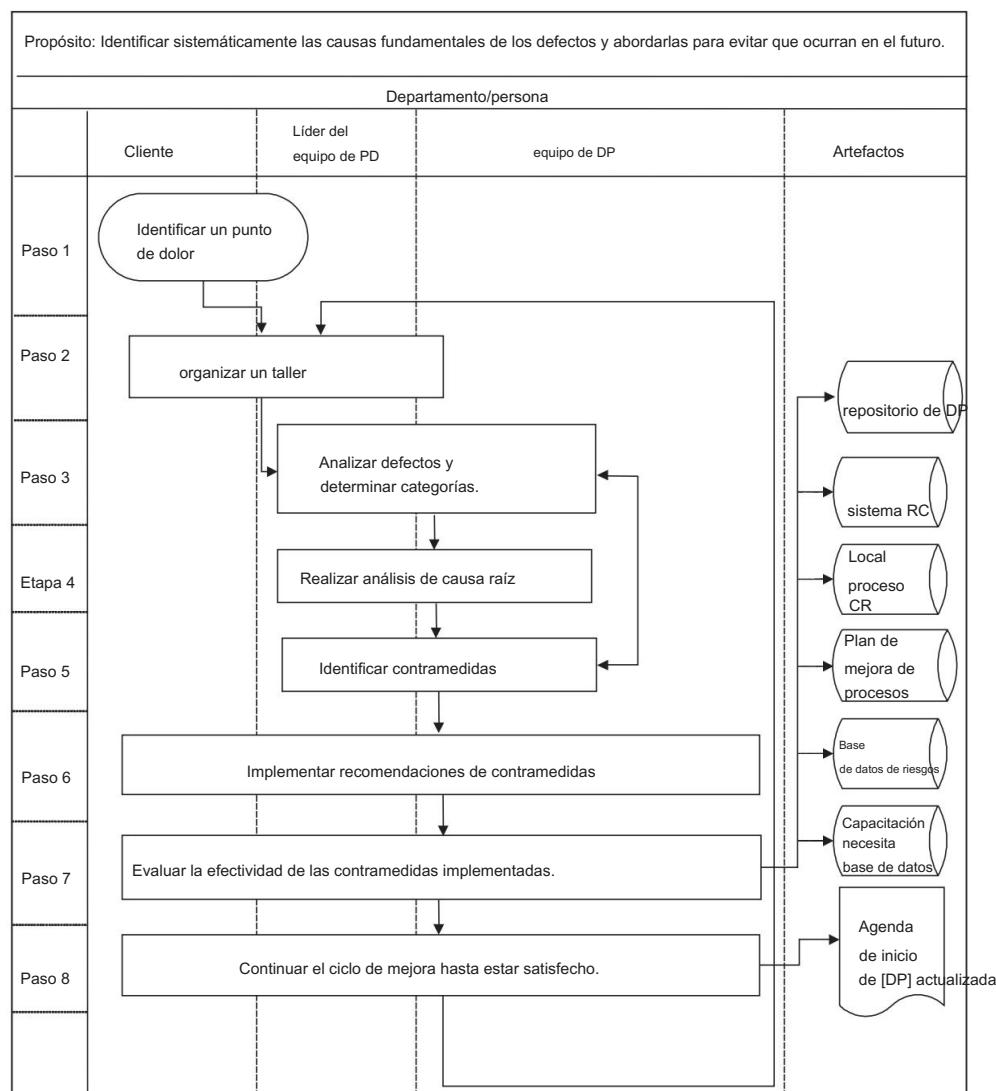


Figura 7.4 Proceso de DP.

El proceso de requisitos será el tema de un “taller del PD”. Si el proceso no está documentado, reúna a algunas partes interesadas y motivadas en una sala durante una o dos horas y documentelo utilizando el procedimiento descrito en el Capítulo 8 para el diseño de un proceso. La documentación del proceso no tiene que ser “perfecta” para permitir el análisis del PD. No te quedes estancado haciendo que los productos de trabajo sean perfectos. (Un día, al observarme hacer la manicura en nuestro jardín, nuestro hijo menor comentó: “Papá, nunca será perfecto”. ¡Tenía razón!) Consulte la Tabla 7.6 para obtener ideas sobre posibles mejoras al proceso de requisitos.

Tabla 7.6 Algunas ideas sobre cómo mejorar un proceso de requisitos

1. Desarrollar un glosario del proyecto (y posiblemente una lista de acrónimos del proyecto).
2. Proporcionar un mecanismo para aumentar la comprensión y la comunicación entre desarrolladores y usuarios.
3. Definir y documentar el proceso de obtención de requisitos.
4. Seleccione algunas métricas para realizar un seguimiento del progreso y proporcionar informes visibles periódicamente.
5. Desarrollar un plan de acción (PA) para la implementación del proceso de ingeniería de requisitos.
6. Mejorar la presentación de informes a la dirección.
7. Tomar medidas para tener mejores reuniones en las que realmente logremos y hagamos las cosas.
8. Implementar un sistema de seguimiento de acciones que se caracterice por criterios de cierre de la acción. elementos.
9. Tome medidas proactivas para lograr una mejor comunicación.
10. Asegurar que las decisiones sean difundidas.
11. Proporcionar capacitación sobre los procesos del proyecto.
12. Desarrollar estrategias para aumentar la "aceptación" de las partes interesadas.
13. Proporcionar y asegurar la participación del cliente en el proceso de requisitos.
14. Descubra cómo identificar los requisitos "reales".
15. Seleccione un método de desarrollo que sea apropiado para el nivel de comprensión de las necesidades del cliente. requisitos.
16. Haz prototipos.
17. Definir un proceso de requisitos.
18. Tomar algunas medidas para fomentar y fomentar un apoyo adicional visible y vocal de la alta dirección. Combine esto con una iniciativa de reducción de costos.
19. Proporcionar el nivel adecuado de detalle con respecto a los requisitos para diferentes grupos de partes interesadas, por ejemplo, un mayor nivel de detalle para los clientes que capture la esencia de lo que necesitan, y un nivel mucho más definido y detallado para los desarrolladores que están a cargo de codificar el requerimiento.
20. A través de discusiones, desarrollar un conjunto acordado de prácticas, enfoques, métodos y herramientas que se utilizarán en el proyecto.
21. Fomentar el compromiso de seguir el conjunto de prácticas, enfoques, métodos y herramientas que se han acepto que.
22. Proporcionar trazabilidad de los requisitos desde la necesidad del cliente hasta la definición del requisito. ya todos los pasos del proceso de desarrollo (por ejemplo, el diseño, el código, el caso de prueba y la verificación de que se ha cumplido el requisito, hasta la inclusión en la documentación del usuario).
23. Mejorar la formación que se imparte.
24. Registrar los requisitos para que exista un entendimiento común y resolver las discrepancias en los comprensión de cada requisito.
25. Reunirse con los usuarios para lograr un significado común de los requisitos del usuario.
26. Aclarar las interpretaciones de documentos formales, como leyes y políticas.
27. Solicitar la "aceptación" de la gerencia en relación con el proceso.
28. Lograr consenso sobre el proceso de desarrollo de requisitos entre todas las partes interesadas.
29. Involucrar a todas las partes interesadas a lo largo del ciclo de vida del desarrollo del sistema.
30. Seleccionar, capacitar e implementar una herramienta de requisitos con anticipación.
31. Reconocer a las personas por sus aportes y esfuerzos.
32. Celebre el progreso.
33. Proporcionar revisiones periódicas de los requisitos.
34. Garantizar una mayor conciencia sobre los requisitos dentro de todos los grupos de proyectos.
35. Proporcionar mecanismos de retroalimentación.
36. Asegurar el flujo hacia abajo.

Tabla 7.6 Algunas ideas sobre cómo mejorar un proceso de requisitos (continuación)

- 37. Documente las cosas que se han hecho en el proyecto.
- 38. Asegurar la comunicación de entendimientos.
- 39. Documente la justificación de cada requisito (¿por qué es necesario?).
- 40. Captar los costos de la volatilidad de los requisitos para las discusiones con el cliente.

Nota: Esta lista se desarrolló a partir de una discusión sobre prácticas efectivas con aportes de 35 profesionales en un taller de requisitos. La lista no tiene prioridades y no todos los elementos están directamente relacionados con los requisitos.

La Tabla 7.6 tiene dos propósitos: primero, ofrecer algunas ideas sobre cómo su proyecto podría mejorar su proceso de requisitos, y segundo, proporcionar temas que podrían considerarse en su taller; por ejemplo, los participantes del taller podrían votar múltiples veces para seleccionar entre estas ideas algunas áreas que podrían justificar contramedidas en su entorno.

El paso 2 es organizar un taller del PD. En un taller de DP, el objetivo (Paso 3) es analizar defectos o problemas para identificar las categorías de problemas o defectos que existen. Es mejor si tiene algunos datos y puede dividir los defectos para determinar distintas categorías de problemas. Pero incluso si no tiene datos, las personas que estén familiarizadas con el proceso podrán identificar cuáles creen que son las causas principales de los problemas. Una vez que hayamos identificado las categorías de problemas, debemos ordenarlas en orden descendente, con la categoría que causa más problemas primero, seguida de la categoría con el siguiente número más grande, y así sucesivamente. Esto se llama análisis de Pareto y es una poderosa técnica de mejora de la calidad.

La figura 7.5 proporciona un ejemplo de un diagrama de Pareto.

El paso 4 es realizar un análisis de causa raíz para identificar las causas raíz de las categorías de problemas o defectos. Una causa raíz se puede definir como la razón principal que resulta en una condición. Queremos tener cuidado de mirar más profundamente que los síntomas del problema o las características del mismo. La razón de esto es que cuando encontramos la causa raíz de un problema, entonces identificamos (Paso 5) y

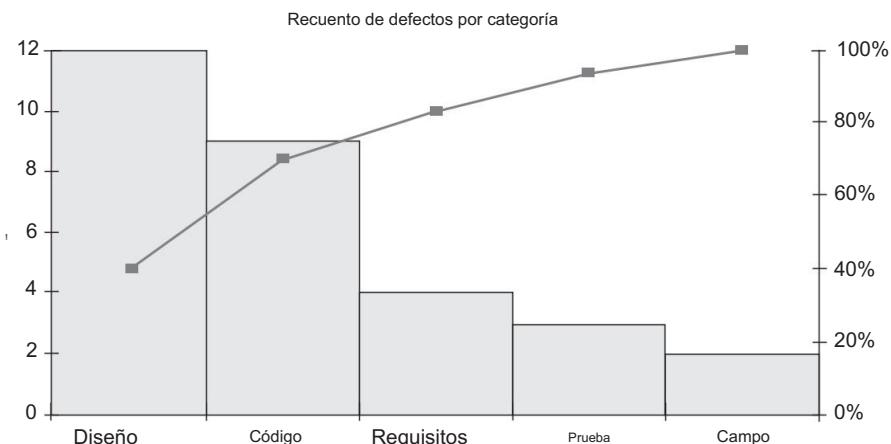


Figura 7.5 Ejemplo de diagrama de Pareto.

implementar (Paso 6) contramedidas (acciones o pasos que eliminan la raíz causas de los problemas) que resuelven eficazmente la causa raíz, ahora tenemos una proceso que produce mejores resultados (la capacidad del proceso ha sido fortalecido o mejorado).

El paso 7 es evaluar la efectividad de las contramedidas implementadas. Después de un período de tiempo razonable, reúna al mismo grupo de partes interesadas que participaron en el Paso 2 y analice si las cosas han mejorado.

Nuevamente, tener datos es lo mejor: una de las contramedidas que puede tomar identificado en el Paso 5 podría haber sido recopilar datos para que podamos mejorar nuestra capacidad de gestionar por hechos y no por intuición. Después de completar todos los pasos del proceso del PD, es probable que desee emprender otro "ciclo de mejora", es decir, recorrer todo el proceso del PD. proceso nuevamente. Este enfoque encarna la mejora continua deseada. ética o valor y brinda la oportunidad de fortalecer y fortalecer aún más mejorar el punto de dolor.

Alternativamente, el grupo puede decidir que simplemente quiere mantener el contramedidas definidas en su lugar, tal vez haciendo algunos ajustes menores ("ajustes") basados en comentarios y observaciones de las partes interesadas. Es importante seguir haciendo hincapié en las actividades de mejora de procesos para garantizar que no sólo se implementan, sino que también se implementan e institucionalizan de manera efectiva. Queremos que las mejoras se conviertan en "la forma en que hacemos negocios". en lugar de algo que requiera un esfuerzo especial. Mi experiencia es que Las mejoras en los procesos deben ser fomentadas y apoyadas, o desaparecerán. De hecho, las mejoras en los procesos desaparecen mucho más rápido de lo que parecen. ven, mi experiencia al respecto es que desaparecen diez veces más rápido que vienen. Puede que no sea tan rápido, ¡pero es rápido! Hacer esfuerzos para reforzar mejoras, como desarrollar y utilizar listas de verificación y procedimientos, brindar capacitación, realizar un seguimiento de los resultados y garantizar el patrocinio y el apoyo de la gerencia. (Un aspecto importante del patrocinio de la dirección es Expresión vocal de que las mejoras en los procesos están ayudando a la organización a alcanzar sus objetivos comerciales. Falta de patrocinio ejecutivo activo y soporte, cualquier proceso o esfuerzo de mejora de la calidad decaerá y eventualmente morirá).

10. Indicas que la estimación es una habilidad importante. ¿Qué aspectos de la estimación son? ¿Crítico para la RA?

Planificar y estimar con precisión proyectos de software es una función de gestión de software extremadamente difícil. Pocas organizaciones han establecido procesos formales de estimación, a pesar de la evidencia que sugiere que las organizaciones sin una estimación formal tienen cuatro veces más probabilidades de sufrir retrasos o proyectos cancelados. Un recurso valioso, en caso de que usted participe En los esfuerzos de su proyecto por mejorar la estimación, está la productividad del software. Research (SPR), Inc. Resuma la información en el sitio web de SPR [24] y no extensas lecturas e investigaciones. Hay mucha información disponible. Quizás el 20% sea realmente útil. Antes de hacer sugerencias, asegúrese de Han digerido la experiencia de la industria.

Existen herramientas, métodos y técnicas que se pueden aplicar para apoyar el desarrollo de estimaciones, como MS Project, modelos de estimación como Constructive Cost Model (COCOMO) y SLIM que utilizan líneas de código, y Knowledge PLAN y COCOMO II que utilizan puntos de función (FP). Se puede invertir mucho esfuerzo en la estimación. La cuestión es que la base de las estimaciones (BOE) debe ser lo más válida posible, o de lo contrario todo el trabajo aplicado al proceso de estimación se desperdicia. La experiencia de la industria es que los requisitos de mayor calidad al inicio de un proyecto contribuyen significativamente a la estimación.

La compañía de Capers Jones, SPR, ha estado recopilando datos sobre el desarrollo de software desde 1984. Jones recomienda encarecidamente el uso de FP y enfatiza el valor de utilizarlos en relación con los requisitos; consulte el Inserto 7.3 para obtener ideas y fuentes de información adicional.

Como se sugirió al comienzo de este capítulo, considere leer y estudiar dos libros que brindan información sobre la estimación: Introducción al proceso de software personal [28] e Introducción al proceso de software en equipo [29], ambos de Watts Humphrey. Proporciona información basada en la experiencia sobre la estimación y lo que hace que los equipos trabajen de forma eficaz. Estos libros ofrecen enfoques que cualquier proyecto y todo PM debería considerar.

11. Aconseja hacer “inspecciones” de todos los documentos relacionados con los requisitos. ¿Por qué no deberíamos conformarnos con realizar revisiones por pares de ellos? ¿En qué se diferencian las inspecciones de las revisiones por pares y por qué tomarse tantas molestias? ¿Qué tipo de inspección es mejor?

Primero, algunos antecedentes sobre las inspecciones: las inspecciones agregan el proceso de detección de defectos al proceso de DP discutido anteriormente en este capítulo. El proceso de detección de defectos tiene que ver con la calidad del documento, particularmente identificar y medir defectos en la documentación y utilizar esta información para decidir cuál es la mejor manera de proceder en consecuencia. Un defecto importante es aquel que probablemente tendrá un orden de magnitud o un impacto mayor en los costos si no se descubre en la etapa de requisitos o diseño de un esfuerzo de desarrollo. Según los datos de Tom Gilb [30], en promedio, el costo de encontrar y reparar defectos importantes es de una hora de trabajo en sentido ascendente, ¡pero nueve horas en sentido descendente! ¡Estos son datos poderosos! Ellos (y quizás algunas lecturas y estudios adicionales) deberían convencerlo de recomendar el uso de inspecciones para todos los documentos relacionados con los requisitos de su proyecto.

La experiencia de Gilb es que, con el apoyo de la gestión, una organización puede mejorar de 20 o más defectos importantes por página a entre cero y dos defectos importantes por página en un año [30].⁷ Él cree que el retorno de la inversión al incorporar el proceso de inspección en un El esfuerzo de desarrollo es de 10:1.

(¡Estos son datos más poderosos que lo ayudarán a convencerlo de recomendar que se utilicen inspecciones en su proyecto!) Otros beneficios incluyen una reducción del retrabajo

7. En realidad, el libro trata sobre inspecciones de cualquier producto de trabajo, no solo de software. El enfoque de los autores es muy riguroso y, por lo tanto, requiere más capacitación y su realización es más costosa que otros tipos de inspecciones. Gilb ahora aboga por otra forma que él llama “inspección extrema”. Consulte su sitio web para obtener más información.

Inserto 7.3—FP y requisitos (Agradecemos a Capers Jones por los datos proporcionados a continuación)

Un FP es una medida de la complejidad del desarrollo de software.

Las métricas de FP se han utilizado para estudiar los tamaños de los requisitos, los costos, los defectos y las tasas de cambio. Capers Jones cree que es útil que los RA comprendan los resultados expresados en términos de FP, pero no necesitan conocer los detalles reales del conteo. Por ejemplo, las especificaciones de requisitos para muchos proyectos tienen un promedio de alrededor de 0,5 páginas por PM. Las pruebas no son eficientes para encontrar problemas de requisitos, pero las inspecciones formales de requisitos promedian alrededor del 65% de éxito en encontrarlos. Al medir los totales de FP después de que se reúnen los requisitos por primera vez y luego nuevamente al final del proyecto, la experiencia de la industria muestra que la tasa promedio a la que surgen nuevos requisitos durante las fases de diseño y codificación es del 2% mensual. La tasa máxima ha llegado al 5% mensual. Con dicha información, es posible planificar con anticipación enfoques eficaces de gestión del cambio. Algunos de los métodos que se han encontrado útiles para deducir cambios de requisitos son la recopilación de requisitos de JAD (talleres de requisitos), prototipos y juntas de revisión de requisitos.

Varios proveedores subcontratados están utilizando los PF como base para los contratos. Algunos utilizan una escala móvil, como que los requisitos iniciales cuestan \$500 por FP; los cambios durante los primeros tres meses después de completar los requisitos iniciales cuestan \$600 por FP; y los cambios posteriores cuestan \$1,000 por FP. Esta escala móvil fomenta el análisis temprano y completo de los requisitos y resuelve algunas áreas difíciles en los contratos de subcontratación.

Fuentes de información adicional:

Garmus, David y David Herron, Análisis de puntos de función: medidas Prácticas de construcción para proyectos de software exitosos [25].

El sitio web del Grupo Internacional de Usuarios de Puntos de Función (IFPUG), en www.ifpug.org [26].

Jones, Capers, Evaluaciones de software, puntos de referencia y mejores prácticas. tic [27].

De: Alcaparras Jones. Usado con permiso.

costos, mayor previsibilidad, mayor productividad y mejor calidad de los documentos.⁸ Los expertos de la industria tienen diferentes puntos de vista sobre cómo se debe realizar el proceso de inspección y la cantidad de capacitación que se requiere para implementar un programa de inspecciones efectivo. Rob Sabourin ofrece una opción económica

8. Gilb, Tom, "Planificación para aprovechar al máximo la inspección", en Conceptos fundamentales para el software de Daughtrey. Ingeniero de Calidad, [31, pág. 178].

enfoque de capacitación e implementación de inspecciones. Contáctelo en rsa-bourin@amibug.com [32].

12. Has puesto mucho énfasis en la calidad. ¿Cómo puede la RA ayudar a aplicar principios de calidad en proyectos de desarrollo de sistemas y software?

El astuto RA estudiará los principios de calidad y se esforzará constantemente por lograr una alta calidad. Taz Daughtrey proporciona un buen resumen de las formas de hacer esto en Conceptos fundamentales para el ingeniero de calidad de software [31], una colección de trabajos recopilados de 32 autores con una gran experiencia en calidad. Los trabajos son informes de prácticas basados en experiencias que han demostrado ser efectivas en una variedad de industrias, aplicaciones y organizaciones. El estudio de estos artículos le proporcionará conocimientos, crecimiento profesional y mejora de sus habilidades. Varios de ellos ofrecen más información sobre los temas tratados en este libro:

Gestión de riesgos;

Satisfacción del cliente;

Uso de un enfoque evolutivo;

Reducción de defectos;

Justificar la mejora de los procesos ante los gerentes (ver tema 16, a continuación);

Elementos humanos en el desarrollo de sistemas;

Utilizar el control estadístico de procesos (SPC) para medir la capacidad del proceso;

Gestionar con métricas;

Seleccionar herramientas;

Evaluación de productos de software;

Aplicar métodos cuantitativos.

Esfuércese por asegurarse de tener un control de calidad proactivo en su proyecto. Por control de calidad proactivo me refiero a especialistas en garantía de calidad que participan en el desarrollo y aplicación de procesos y esfuerzos de prevención de defectos que incorporan calidad a los productos de trabajo. La experiencia nos ha demostrado que no es posible eliminar la mala calidad de los productos de trabajo. Por ejemplo, aunque las pruebas son importantes, la experiencia de la industria muestra que las pruebas encuentran menos del 50% de los defectos en un producto de software.⁹ Se debe tener software de calidad en el proceso de prueba para obtener resultados de calidad aún mejor.

9. Watts Humphrey, tutorial de PSP en Northrop Grumman IT DES el 20 de febrero de 2003. Humphrey enfatizó que la práctica actual de desarrollo de software que se utiliza en la industria actual se preocupa por eliminar defectos.

Su experiencia es que muchas organizaciones gastan el 40% del esfuerzo y los costos del proyecto en pruebas, mientras que las organizaciones con procesos más maduros pueden reducir los costos de las pruebas al 10%.

de las actividades de prueba. Los especialistas en control de calidad pueden y deben tener un enorme impacto positivo en un proyecto de desarrollo de software o sistema. Ellos (como el resto de nosotros) deben estar facultados por la dirección para realizar una contribución valiosa [33].

Me gustaría analizar un método adicional antes de dejar este tema: Implementación de funciones de calidad (QFD). QFD comenzó a mediados de la década de 1960 en Japón como un sistema de calidad enfocado en desarrollar productos y servicios que satisfagan a los clientes. Jones proporciona una descripción de QFD y un informe de estado sobre su uso en su libro Calidad del software: análisis y directrices para el éxito [34]. Señala que QFD es una actividad grupal estructurada que involucra a clientes y personal de desarrollo de productos. A veces se le llama "la casa de la calidad" porque uno de los principales tipos de matrices de planificación se asemeja al techo puntiagudo de una casa. En *Developing Products in Half the Time*, 2.^a ed., Smith y Reinertsen analizan "Blitz QFD" [35, págs. 102-103]. La hoja de ruta hacia el éxito repetible: uso de QFD para implementar el cambio [36] de Bicknell y Bicknell proporciona una revisión integral de QFD y muestra cómo se puede utilizar en todos los niveles de una organización. Ramaswami de Global Technology Operations en India ha utilizado QFD en varios proyectos (consulte su sitio web en www.gemedicalsystems.com). Mark Paulk informa que nueve organizaciones en la base de datos de SEI de nivel 4 y 5 utilizan QFD, cinco en India y cuatro en Estados Unidos.

Zultner advierte que el uso del software QFD está aumentando. Zultner ha realizado presentaciones sobre software QFD en la conferencia QAI en Bangalore, en la conferencia de la Organización Europea para la Calidad en Budapest y en el Segundo Congreso Mundial para la Calidad del Software en Yokohama. Informa que algunas empresas estadounidenses (como Andersen Consulting) llevan años utilizando el software QFD a nivel mundial. Él advierte que la mejor manera de comenzar con QFD para software es Blitz QFD. Zultner cree que ninguna otra herramienta, técnica o método se acerca siquiera al QFD para abordar la "frontal confusa de la definición de requisitos". La experiencia de Capers Jones es que QFD funciona mejor en empresas que tienen departamentos de control de calidad bastante sofisticados. En resumen, en lo que respecta a QFD, desde mi punto de vista, la conclusión es que primero hay que descubrir y desarrollar los requisitos reales para que el método sea eficaz.

13. Parece haber mucha confusión en nuestra industria con respecto a los términos verificación y validación. ¿Puede explicar por qué esto es así y también aclarar los usos sugeridos de los dos términos?

Sí. Como señala Jeff Grady en sus materiales de capacitación, libros y presentaciones, la gente de software y de ingeniería de sistemas tiende a usar las dos palabras a la inversa, y esto causa una tremenda confusión [37]. Lo más importante a este respecto es aclarar cómo se utilizarán las dos palabras en su proyecto.

Aquí están las definiciones que sugiero:

Verificación: un proceso para garantizar que la solución de diseño satisfaga los requisitos;

Validación: proceso para confirmar que los requisitos reales son implementado en el sistema entregado.

Debe aclarar estas dos palabras para ayudar a que su proyecto evite interminables y frustrantes discusiones sobre lo que estas palabras "realmente significan" y quién tiene "razón".

14. Los "agilistas" defienden que las metodologías de desarrollo ágiles prometen una mayor satisfacción del cliente, menores tasas de defectos, tiempos de desarrollo más rápidos y una solución a requisitos que cambian rápidamente. ¿Debería recomendar que consideremos métodos de desarrollo ágiles en mi proyecto?

Creo que hay algo que decir a favor de los enfoques tanto disciplinados como ágiles. Me preocupa cuando leo y escucho opiniones que se inclinan fuertemente hacia un lado o hacia el otro, especialmente cuando uno critica al otro.

Ambos enfoques tienen deficiencias que, si no se toman en cuenta, pueden llevar al fracaso del proyecto. Cada enfoque tiene un "terreno de partida" donde es preferible al otro; consulte la Tabla 7.7.

Mi sugerencia aquí es que lea y digiera "Observaciones sobre el equilibrio entre disciplina y agilidad" [38] de Barry Boehm y Richard Turner. Creen que están surgiendo estrategias para integrar los dos enfoques de una manera que aproveche sus fortalezas y evite sus debilidades.

15. ¿Cuál es el valor del conocimiento práctico?

Mientras escribía este libro, el consultor y experto en ingeniería de requisitos de la industria, Ian Alexander [39], fue un asesor valioso y, como resultado, se convirtió en un buen amigo. Ian compartió conmigo una idea sobre el conocimiento práctico:¹⁰

El énfasis excesivo en el conocimiento intelectual en comparación con la habilidad práctica se remonta a Aristóteles: el "Fuego" creativo y el "Aire" comunicativo eran elementos masculinos, mientras que la "Tierra" pesada y fértil y el "Agua" emocional eran femeninos. Se despreciaba la terrenalidad práctica. Pero hoy podríamos argumentar lo contrario: habilidad práctica que integra la inteligencia emocional (cf.

Daniel Goleman) y cantidades apropiadas de creatividad e intelecto es lo que hace que los proyectos funcionen. (Por ejemplo: ¿los académicos saben más que las personas que realmente hacen el trabajo?)

16. ¿Qué pasa si mi PM, y/o el equipo directivo de nuestra organización, o nuestro cliente?
¿No apoya el concepto de mejora de procesos?

Mi experiencia en hacer sugerencias de mejora en proyectos y organizaciones durante más de 30 años es que quizás un tercio de todos los gerentes están abiertos a la mejora de procesos. En mi experiencia, la mayoría de los gerentes no apoyan mucho la mejora de procesos, especialmente cuando se trata de

Tabla 7.7 Bases para los enfoques ágiles y disciplinados

Característica	Ágil	Disciplinado
Solicitud:		
Metas primarias	Valor rápido; respondiendo al cambio	Previsibilidad, estabilidad, alta garantía
Tamaño	Equipos y proyectos más pequeños	Equipos y proyectos más grandes
Ambiente	Turbulento; alto cambio; centrado en el proyecto	Estable; cambio bajo; centrado en el proyecto/organización
Gestión:		
Relaciones con los clientes	Clientes dedicados in situ; enfocado en incrementos priorizados	Interacciones con el cliente según sea necesario; centrado en las disposiciones del contrato
Planificación y control	Planes internalizados; control cualitativo	Planes documentados, cuantitativos control
Comunicaciones	Conocimiento interpersonal tácito	Conocimiento explícito documentado
Técnico:		
Requisitos	Historias informales priorizadas y prueba. casos; pasando por lo imprevisible cambiar	Proyecto formalizado, capacidad, interfaz, calidad, previsible requisitos de evolución
Desarrollo	Diseño simple; incremento corto; la refactorización se supone económica	Diseño amplio; incrementos más largos; la refactorización se supone costosa
Prueba	Los casos de prueba ejecutables definen requisitos, pruebas	Planes de prueba documentados y procedimientos
Personal:		
Clientes	CRACK* dedicado y ubicado en el mismo lugar <small>artistas intérpretes o ejecutantes</small>	Artistas CRACK*, no siempre colocado
Desarrolladores	Al menos 30% del nivel Cockburn a tiempo completo 2 y 3 peritos; sin nivel 0 o -1 personal	50% Cockburn Nivel 3 temprano; 10% a lo largo de; 30% de Nivel 0 viable; sin nivel -1s**
Cultura:	Consuelo y empoderamiento a través de muchos grados de libertad	Comodidad y empoderamiento a través de marco de políticas y procedimientos

* Colaborador, representativo, autorizado, comprometido, conocedor.

** Estos números variarán particularmente según la complejidad de la aplicación.

De: Barry Boehm. Reimpreso con permiso.

presentado usando esas palabras. Las causas fundamentales de este comportamiento son una misterio para mí. Una posible causa simple es que los PM estén demasiado ocupados para considerar seriamente las oportunidades de mejora. Otra posibilidad es que sus creencias van en contra de la idea. Una tercera explicación posible es que los PM pueden ser reacios a recomendar iniciativas a menos que sean sugeridas, o al menos respaldadas, por el cliente, y muchos clientes no lo están. conscientes del poder y el valor de la mejora de procesos. Otra posibilidad es que la cultura organizacional del proyecto no está comprometida con la continuidad mejora. Otra posible explicación es que los gerentes no han Tenía una manera de determinar cuánta mejora se debe al proceso versus otros factores. La conclusión desde la perspectiva de la RA es que probablemente puedes tener mucho más impacto si te alineas en un entorno que apoya la mejora continua del proceso. Mi creencia es que tú también lo harás

disfrute de mucha más satisfacción laboral y diviértase más en el trabajo haciendo esto. Véase "Efectos de la madurez del proceso en el esfuerzo de desarrollo" de Bradford Clark [40] para obtener información valiosa. Su estudio, utilizando una muestra de 112 proyectos, concluyó que un cambio en un nivel de madurez del proceso usando el marco CMM® resultó en una reducción del esfuerzo de desarrollo del 10 al 32 por ciento. Esto es fenomenal y merece la atención de cada proyecto. Véase también (1) el estudio de Brodmann y Johnson sobre el retorno de la inversión del SPI medido por la industria estadounidense [41]; (2) "Los beneficios económicos de SPI" de Butler [42]; (3) Dion, "Mejora de procesos y balance corporativo" [43]; (4) "Beneficios del SPI basado en CMM: resultados iniciales" de Herbsleb et al. [44]; (5) "SPI en Hughes Aircraft" [45] de Humphrey et al.; (6) "Un caso de negocio para la mejora de procesos de software revisado: Medición del retorno de la inversión desde la ingeniería y gestión de software" de McGibbon. [46]; y (7) Informes técnicos del DACS [47]. No estoy de acuerdo con Grady Booch en que "CMM es ortogonal al éxito del proyecto y no lo suficientemente ágil para el éxito económico". Esto es contrario a mi experiencia personal. Le pedí a Booch que explicara su perspectiva al respecto; aclaró que, según su experiencia, algunos proyectos con "proceso maduro" no tienen éxito porque el equipo del proyecto está tan absorto en el proceso que descuidan la importancia de ser ágiles en el mercado.¹¹ Booch Estaba aclarando que el CMM mide la madurez del proceso, mientras que el éxito en los negocios se mide por el ROI.

Entonces, en respuesta a la pregunta, le aconsejo que, primero, intente alentar a su gerente, su proyecto y su organización a adoptar la mejora de procesos. Si descubre que se está topando contra una pared de ladrillos (nadie apoyará las iniciativas de mejora de procesos), cambie su entorno (muévase a un proyecto u organización diferente que sí apoye la mejora de procesos). La vida es demasiado corta y pasamos demasiado tiempo en el trabajo como para no sentirnos satisfechos con lo que hacemos. Nuestra actitud hacia nuestro trabajo impacta nuestras familias y nuestras vidas. Tome medidas si es necesario. Nuestro Señor provee.

17. ¿Cómo se debe aplicar la estructura de desglose del trabajo?

La estructura de desglose del trabajo (WBS) es una herramienta de planificación que descompone las actividades de un proyecto en categorías de tareas de trabajo a partir de las cuales se pueden asignar y rastrear los costos y el esfuerzo. El desarrollo de una EDT depende de la cultura de la organización, el estilo de gestión de proyectos, las preferencias del cliente y otros factores específicos del proyecto. Un enfoque típico es descomponer el trabajo en subsistemas, componentes, funciones, unidades organizativas y fases del ciclo de vida. Mi experiencia es que la estructura de la EDT de un proyecto es a menudo arbitraria, y el uso de las EDT es defectuoso porque se estructuran prematuramente en torno al diseño del producto y se definen con demasiado o muy poco detalle. Un enfoque alternativo es organizar los elementos de planificación en torno al marco del proceso. El Inserto 7.4 ofrece un enfoque que relaciona los requisitos con los productos del trabajo.

11. Comunicación personal por correo electrónico con el autor, 12 de marzo de 2003.

Inserto 7.4—Estructura de desglose de productos

La mayoría de las orientaciones sobre gestión de proyectos sugieren que la planificación de un proyecto es una secuencia de eventos relativamente sencilla. Primero, define los requisitos y luego identifica los productos de trabajo finales que se entregarán al cliente. A continuación, se identifican los productos de trabajo intermedios necesarios para desarrollar los productos de trabajo finales y luego se desarrolla una EDT, que indica qué trabajo se debe realizar. Con la WBS, debería ser un proceso sencillo estimar el esfuerzo necesario para realizar el trabajo y asignar los recursos disponibles para desarrollar un cronograma.

Con un calendario en la mano, todo está bien. Su proyecto está planificado y el éxito está asegurado. Incluso puede utilizar herramientas de planificación de proyectos sofisticadas y económicas para imprimir gráficos de barras muy grandes e impresionantes que detallan hasta el último día exactamente en que se realizará todo el trabajo.

Desafortunadamente, el mundo real de la gestión de proyectos a veces no es tan sencillo. Una dificultad importante puede ser el salto de los requisitos a la EDT a través de esos escalones llamados “productos de trabajo”. Claro, los principales productos de trabajo que entregará al cliente al final del proyecto deberían ser relativamente sencillos de determinar. Después de todo, eso es lo que el esfuerzo de requisitos debería explicar con suficiente detalle para que usted sepa lo que quiere su cliente. Pero son esos molestos productos de trabajo intermedios y otros resultados potencialmente ocultos que no se hacen obvios durante las primeras etapas de la planificación del proyecto los que pueden hacer que el camino sea resbaladizo, si no francamente peligroso. Si el proyecto implica algún tipo de innovación, desde algo que usted o su equipo nunca hayan hecho antes hasta algo que nadie más haya hecho antes, entonces estos productos de trabajo intermedios podrían no ser del todo obvios.

Luego está el desafío de la propia WBS. El estándar definido en los Fundamentos de Gestión de Programas es que una EDT es un “árbol genealógico de actividades que organiza, define y muestra gráficamente el trabajo total a realizar para lograr los objetivos finales de un proyecto”. [48] La orientación estándar es que la WBS debe basarse en productos, lo que significa que su jerarquía debe estar impulsada por los principales productos finales del trabajo. Pero casi cualquiera que haya trabajado en un proyecto de TI sabe que muchas WBS están organizadas por fases en los niveles superiores, con algo similar a Definir requisitos, Diseño, Construcción y pruebas unitarias, Pruebas de integración y Entrega de productos que representan los niveles superiores de la jerarquía. . Estos no están relacionados en absoluto con el producto, sino que representan un flujo de trabajo que, con suerte, dará como resultado el resultado deseado.

Desafortunadamente, muchos PM adoptan esta estructura estándar muy temprano en el proyecto, sin saber realmente qué subcomponentes deben construirse o qué pasos intermedios deben tomarse. Puede proporcionar la falsa seguridad de que sabemos lo que estamos haciendo, cuando en realidad aún existen importantes incógnitas sobre cuál será el trabajo final o intermedio.

Inserto 7.4—Estructura de desglose de productos (continuación)

Serán los productos y qué tareas serán necesarias para construirlos. Luego está el problema de que una vez organizada la EDT, existe una gran reticencia a cambiarla. Como ha señalado Noel Harroff, "una vez que se elabora la EDT, puede ser una "trabajo difícil de deshacer" [49].

Desafortunadamente, muchos proyectos fracasan porque están trabajando según un cronograma, basado en una EDT que no proporciona una hoja de ruta realista para realizar el trabajo. Lo que parecía bueno sobre el papel no es del todo factible en el mundo real. ¿Hay algo que pueda ayudar?

La metodología de gestión de proyectos PRINCE2 desarrollada para el gobierno británico ofrece algunas ideas que pueden ayudar. La planificación basada en productos es una característica clave de PRINCE2. Se centra en los productos a entregar y su calidad. Un paso clave en la planificación de proyectos es el desarrollo de una estructura de desglose de productos (PBS). En el método PRINCE2, la planificación se realiza en tres pasos: (1) desarrollar un PBS, (2) documentar las descripciones de los productos y (3) producir un diagrama de flujo del producto que dé como resultado una red de actividades laborales. Max Wideman [50] proporciona una excelente comparación de los enfoques PRINCE2 y PMBOK para la gestión de proyectos. La metodología PRINCE2 también considera la planificación como una actividad continua que ocurre durante todo el ciclo de vida del proyecto. De esa manera, a medida que se obtiene nueva información sobre los requisitos o la posible solución, el proyecto tiene un proceso para reincorporar esta información al plan del proyecto.

No es necesario cambiar a toda la metodología PRINCE2 para beneficiarse de algunos de sus conceptos, especialmente la idea de un PBS. Una ventaja clave de desarrollar un PBS es que puede centrar la atención en los resultados al nivel práctico más pequeño, en lo que se necesita para definir los productos finales. Este proceso necesariamente identificará los subcomponentes y componentes intermedios necesarios para producir el producto final.

Este enfoque en el producto en lugar del trabajo necesario para producirlo puede ayudar a despejar el camino e identificar áreas que no se comprenden bien y necesitan aclaración. Esto puede resultar en el reconocimiento de áreas que necesitan estudios de viabilidad o investigaciones adicionales.

¿En qué momento se detiene la PBS y comienza el trabajo, identificado en una red de actividades laborales o una WBS? Por ejemplo, un enfoque común en proyectos de TI es desarrollar un sistema como una serie de compilaciones. Utilizando el enfoque PBS, cada compilación sería un producto entregable al cliente y el flujo de trabajo resultante reflejaría esto en el nivel superior de la jerarquía. Sin embargo, al comienzo de la planificación del proyecto, es posible que ni siquiera se considere el concepto de construcciones separadas. La distinción entre producto y trabajo no es tan crítica como el reconocimiento de que la planificación es un proceso creativo para descubrir los componentes, las actividades y las relaciones entre ellos. Lo que comienza como el PBS inicial, que necesariamente se centrará estrictamente en los entregables inmediatos al cliente, evolucionará a medida que la planificación identifique actividades de reducción de riesgos, que podrían ser

Inserto 7.4—Estructura de desglose de productos (continuación)

manifestado como compilaciones separadas, para finalmente entregar el producto. El punto clave es que un PBS puede servir como otra herramienta en la planificación de proyectos. que minimizará los errores en aquellos productos de trabajo mal definidos o productos de trabajo intermedios que podrían causar retrasos o desastres.

Juan E. Moore

ingeniero de gestión de proyectos

18. ¿Cuál es un buen enfoque para considerar los riesgos de los requisitos?

Algunas organizaciones utilizan el Cuestionario basado en taxonomía (TBQ) del SEI como una herramienta para la identificación de riesgos. Varias de las preguntas TBQ fueron diseñadas para obtener riesgos relacionados con los requisitos; estos se proporcionan en la Tabla 7.8. Aquellos descritos en la Sección A.1 se relacionan con la calidad de los propios requisitos. Los descritos en la Sección A.2 son riesgos relacionados con los requisitos asociados con el diseño, y los descritos en la Sección A.4 están asociados con integración y prueba. Los descritos en el apartado A.5 son riesgos asociados a las especialidades de ingeniería. Los descritos en la Sección B se relacionan con el proceso de desarrollo. Estas preguntas proporcionan una buena base para evaluar riesgos relacionados con los requisitos asociados con su tarea o proyecto.

Ian Sommerville ofrece una buena visión general de la gestión de riesgos. fuentes, probabilidades y efectos de procesos y riesgos en Ingeniería de Software, 6to ed. [52, págs. 84–90]. Barry Boehm proporciona ideas pioneras sobre economía de la ingeniería de software [53, págs. 279–288, 297–300 y 588–590].

Resumen

Este capítulo proporciona una discusión sobre habilidades adicionales e información útil. a la RA. Es probable que no necesites la información relativa a todos estos temas ya sea inmediatamente o en cualquier momento determinado. También es probable que necesitarás conocer sobre la mayoría de estas áreas en algún momento de tu trabajar. Podría considerar probar técnicas como DP e inspecciones. si (1) no los ha utilizado, o (2) su proyecto no se aplica actualmente a ellos. Los enfoques y técnicas discutidos en muchos de estos temas. debe usarse continuamente en todos los proyectos, por ejemplo, reduciendo errores de requisitos, CM, comprensión de V&V e inspecciones de productos de trabajo relacionados con requisitos. Otros están más en la categoría “es bueno tenerlos pero son muy importantes”, como la estimación, la mejora y el refinamiento de su facilitación. habilidades, ser líder en su proyecto y buscar la mejora continua. Otros más, como la IE y el uso de FP, son aplicables a situaciones especiales. Por ejemplo, considere usar IE cuando sea necesario estimar requisitos cuantitativamente para tomar decisiones de diseño. Los PF pueden ayudar cuando medir los totales de FP en diferentes puntos del proyecto ayudará a planificar

Tabla 7.8 Análisis de riesgos relacionados con los requisitos

A.1	Requisitos: ¿Existen riesgos que puedan surgir de la imposición de requisitos a la producto?
A.1.a	Requisitos: estabilidad ¿Están cambiando los requisitos incluso mientras se fabrica el producto? [1] ¿Son estables los requisitos? [1.a] En caso negativo: ¿Cuál es el efecto en el sistema? Calidad/Funcionalidad/Calendario/Integración/Pruebas de diseño [2] ¿Están cambiando las interfaces externas?
A.1.b	Requisitos: integridad ¿Faltan requisitos o no están especificados por completo? [3] ¿Hay algún TBD en las especificaciones? [4] ¿Hay requisitos que usted sabe que deberían estar en la especificación, pero no lo están? [4.a] En caso afirmativo: ¿Podrá incorporar estos requisitos al sistema? [5] ¿Tiene el cliente requisitos o expectativas no escritos? [5.a] En caso afirmativo: ¿Existe alguna manera de capturar estos requisitos? [6] ¿Están las interfaces externas completamente definidas?
A.1.c	Requisitos: claridad ¿Los requisitos no están claros o necesitan interpretación? [7] ¿Puede comprender los requisitos tal como están escritos? [7.a] En caso negativo: ¿Se están resolviendo satisfactoriamente las ambigüedades? [7.b] En caso afirmativo: ¿No hay ambigüedades ni problemas de interpretación?
A.1.d	Requisitos: validez ¿Los requisitos conducirán al producto que el cliente tiene en mente? [8] ¿Existe algún requisito que pueda no especificar lo que el cliente realmente quiere? [8.a] En caso afirmativo: ¿Cómo está resolviendo esto? [9] ¿Usted y el cliente entienden lo mismo por los requisitos? [9.a] En caso afirmativo: ¿Existe algún proceso mediante el cual determinar esto? [10] ¿Cómo se validan los requisitos? Creación de prototipos/Análisis/Simulaciones
A.1.e	Requisitos: viabilidad ¿Son los requisitos inviables desde un punto de vista analítico? [11] ¿Hay algún requisito que sea técnicamente difícil de implementar? [11.a] En caso afirmativo: ¿Cuáles son? [11.b] En caso afirmativo: ¿Por qué son difíciles de implementar? [11.c] En caso negativo: ¿Se realizaron estudios de viabilidad para estos requisitos? [11.c.1] En caso afirmativo: ¿Qué tan seguro está de las suposiciones hechas en los estudios?
A.1.f	Requisitos: precedente ¿Los requisitos especifican algo que nunca antes se había hecho o algo que su empresa no lo ha hecho antes? [12] ¿Existen requisitos de última generación? Tecnologías/Métodos/Idiomas/Hardware [12.a] En caso negativo: ¿Alguno de estos es nuevo para usted? [12.b] En caso afirmativo: ¿Tiene el programa conocimientos suficientes en estas áreas? [12.b.1] En caso negativo: ¿Existe un plan para adquirir conocimientos en estas áreas?

Tabla 7.8 Análisis de riesgos relacionados con los requisitos (continuación)

A.1.g	Requisitos: escala	
	¿Los requisitos especifican un producto más grande, más complejo o que requiere una mayor organización que en la experiencia de la empresa?	
[13]	¿Es el tamaño y la complejidad del sistema una preocupación?	
[13.a]	En caso negativo: ¿Ha hecho algo de este tamaño y complejidad antes?	
[14]	¿El tamaño requiere una organización mayor a la habitual en su empresa?	
A.2	Diseño: ¿Existen riesgos que puedan surgir del diseño que el proyecto ha elegido cumplir sus requisitos?	
A.2.a	Diseño—Funcionalidad	
	¿Existe algún problema potencial al cumplir con los requisitos de funcionalidad?	
[15]	¿Existe algún algoritmo específico que pueda no satisfacer los requisitos?	
[15.a]	En caso negativo: ¿Alguno de los algoritmos o diseños es marginal con respecto al cumplimiento requisitos?	
[diecisésis]	¿Cómo se determina la viabilidad de algoritmos y diseños?	
	Creación de prototipos/modelado/análisis/simulaciones	
[18]	¿Existen requisitos o funciones que sean difíciles de diseñar?	
[18.a]	En caso negativo: ¿Tienen soluciones para todos los requisitos?	
[18.b]	En caso afirmativo: ¿Cuáles son los requisitos? ¿Por qué son difíciles?	
A.2.d	Diseño—Rendimiento	
	¿Existen requisitos estrictos de tiempo de respuesta o rendimiento?	
[22]	¿Hay algún problema con el rendimiento?	
	Rendimiento	Programación asincrónica
	Respuesta en tiempo real	eventos en tiempo real
	Respuesta, contención o acceso a la base de datos	Tiempo de respuesta
		Cronogramas de recuperación
[23]	¿Se ha realizado un análisis de desempeño?	
[23.a]	En caso afirmativo: ¿Cuál es su nivel de confianza en el análisis de desempeño?	
[23.b]	En caso afirmativo: ¿Tiene un modelo para realizar un seguimiento del desempeño a través del diseño y la implementación?	
A.2.e	Diseño: comprobabilidad	
	¿Es el producto difícil o imposible de probar?	
[26]	¿Los evaluadores participan en el análisis de requisitos?	
A.2.f	Diseño: restricciones de hardware	
	¿Existen restricciones estrictas en el hardware de destino?	
[27]	¿El hardware limita su capacidad para cumplir algún requisito?	
	Arquitectura	Rendimiento
	Fiabilidad	Capacidad de memoria
	Funcionalidad	Respuesta en tiempo real
		Cronogramas de recuperación
		Disponibilidad
A.4	Integración y prueba: ¿Existen riesgos que puedan surgir de la forma en que el proyecto elige reunir las piezas y demostrar que funcionan en conjunto?	
A.4.a	Integración y prueba: entorno	
	¿Es adecuado el entorno de integración y prueba?	
[48]	¿La instrumentación de hardware y software facilita las pruebas?	
[48.a]	En caso afirmativo: ¿Es suficiente para todas las pruebas?	
A.4.b	Integración y prueba—Producto	
	¿La definición de la interfaz es inadecuada? ¿Las instalaciones son inadecuadas? ¿El tiempo es insuficiente?	

Tabla 7.8 Análisis de riesgos relacionados con los requisitos (continuación)

Tabla 7.8 Análisis de riesgos relacionados con los requisitos (continuación)

B.2	Sistema de desarrollo: ¿Existen riesgos que puedan surgir de las herramientas de hardware y software que el proyecto ha elegido para controlar y facilitar su proceso de desarrollo?		
B.2.b	Sistema de desarrollo: idoneidad ¿El sistema de desarrollo soporta todas las fases, actividades y funciones?		
[94]	¿El sistema de desarrollo apoya todos los aspectos del programa?		
Análisis de requerimientos	Análisis de rendimiento	CM	
Diseño	Codificación	Trazabilidad de requisitos	
Prueba	Documentación	Seguimiento de la gestión	

Siguiendo el modelo del SEI TBQ [51].

anticiparse para enfoques efectivos de gestión de cambios (para anticipar y controlar los cambios de requisitos) o cuando el uso de FP puede facilitar el cálculo de costos o la comprensión y reducción de la complejidad del código. Algunos expertos en ingeniería de requisitos de la industria siguen siendo escépticos con respecto a la aplicabilidad general de IE y FP.

Vale la pena reiterar que los proyectos pequeños deben evitar utilizar la pequeñez como excusa para no utilizar la disciplina y los procesos. La mayoría de las cosas que deben hacerse en proyectos medianos y grandes también deben hacerse en proyectos pequeños; es una cuestión de escala.

Lo más importante es que te sientas realizado en tu trabajo como RA.
Confío en que la discusión en este capítulo le ayudará.

Caso de estudio

Un artículo del Washington Post¹² informó sobre una situación que involucra a desarrolladores e integradores de sistemas. Una junta federal despidió al contratista que contrató en 1997 para crear un nuevo sistema informático para realizar un seguimiento de 100 mil millones de dólares en ahorros para la jubilación, citando repetidas demoras y software que contenía muchos defectos. La junta declaró que el contratista era “incapaz de cumplir sus compromisos” y presentó una demanda por 350 millones de dólares en daños y perjuicios. La demanda alegaba que el sistema requería 40 horas para registrar transacciones, en lugar del requisito de 11 horas, y que no podía distinguir entre un participante que vivía en Delaware y uno que vivía en Alemania, lo que requeriría toda la correspondencia. abordarse manualmente, en lugar de hacerlo por computadora.

La rescisión del contrato y la demanda “sorprendieron a los funcionarios contratistas”. quien explicó que su cliente solicitó repetidamente cambios de diseño, lo que hizo que los cronogramas de entrega del nuevo sistema fueran un objetivo cambiante. “Después de más de tres años, la junta aún no ha determinado cuáles serán sus sistemas.

12. Washington Post, 18 de julio de 2001, págs. B1 y B5.

las necesidades son. Hemos desarrollado más de 1,2 millones de líneas de código de software (cinco veces la estimación original) y más del 70 por ciento del software se ha completado y probado en su totalidad".

Originalmente estaba previsto que el nuevo sistema informático estuviera en funcionamiento Mayo de 2000. La fecha de entrega se pospuso varias veces.

Se contrató a otro contratista para hacerse cargo del proyecto, pero no estaba claro cuándo podría entrar en funcionamiento el nuevo sistema.

Este ejemplo de requisitos que salieron mal puso en peligro la reputación de la empresa contratista, desperdició millones de dólares, afectó negativamente la misión y las operaciones del cliente e impactó negativamente las vidas de muchas personas y familias involucradas en ambos lados del conflicto.

contrato.

¿Qué se podría haber hecho para evitar este desastre? Lo más fundamental era crear, desarrollar y mantener una relación de asociación que estableciera el éxito del proyecto como objetivo. El cliente y el contratista de desarrollo deberían haber implementado al comienzo del contrato un conjunto de mecanismos para mantener el tren en las vías en lugar de arriesgarse a permitir que las cosas salgan mal. Específicamente, se podría hacer lo siguiente para evitar un desastre similar:

Mediante el uso de un taller de colaboración facilitado por un experto externo, se debe definir, documentar y firmar una visión conjunta del éxito del proyecto. El documento debe incluir un conjunto de objetivos mutuamente acordados que definen los requisitos de alto nivel para el contrato. Lo más importante es que desde el principio se establezca el compromiso de ambas partes de apoyarse mutuamente.

El compromiso señalado aquí debe incluir un compromiso de resolver problemas y cuestiones tan pronto como surjan. Se debe crear una "escalera de resolución de problemas" en el taller de asociación inicial y se debe lograr el compromiso de las personas involucradas en ambos lados del contrato de utilizar este mecanismo para mantener el tren en las vías mientras avanza el trabajo.

Se debe desarrollar en colaboración un conjunto de principios rectores sobre cómo las partes se tratarán entre sí durante el contrato. Algunos ejemplos de estos principios rectores son la honestidad, la comunicación abierta, el liderazgo proactivo, la pronta finalización de las acciones asignadas y la revisión de los documentos, el apoyo mutuo entre sí y el establecimiento de un conjunto de "reglas de conducta" para las reuniones, el correo electrónico comunicaciones y cómo las personas deben tratarse entre sí.

Referencias

[1] Hooks, IF y KA Farry, Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos, Nueva York: AMACOM, 2001, p. 6.

- [2] Watts, FB, Manual de control de documentos de ingeniería: gestión de configuración en Industria, 2^a ed., Park Ridge, Nueva Jersey: Noyes Publications, 2000.
- [3] León, AA, Guía para la gestión de la configuración de software, Norwood, MA: Artech Casa, 2000.
- [4] Estándar EIA 649, "Estándar de consenso nacional para la configuración Gestión."
- [5] Fowler, M., UML Distilled: Aplicación del lenguaje de modelado de objetos estándar. Lectura, MA: Addison-Wesley, 1997.
- [6] Object Management Group (OMG), "Introducción a UML", en www.omg.org/gettingstarted/what_is.uml.htm.
- [7] Diós mio, "Especificación del lenguaje de modelado unificado OMG", versión 1.4, septiembre de 2001, en www.omg.org/technology/documents/formal/mof.htm.
- [8] Gottesdiener, E., "Diez formas principales en que los equipos de proyecto hacen un mal uso de los casos de uso y cómo corregirlos", en www.therationaledge.com/content/jun_02/t_misuseUseCases_eg.jsp.
- [9] Comité de Estándares de Ingeniería de Software del IEEE, Estándar IEEE 1233a-1998, "Guía IEEE para el desarrollo de especificaciones de requisitos del sistema", IEEE Computer Society, 8 de diciembre de 1998.
- [10] Comité de Estándares de Ingeniería de Software del IEEE, Estándar IEEE 830, "Práctica recomendada del IEEE para especificaciones de requisitos de software", IEEE Computer Society, 2 de diciembre de 1993.
- [11] Robertson, S. y J. Robertson, Dominar el proceso de requisitos, Harlow, Reino Unido: Addison-Wesley, 1999.
- [12] Kulak, D. y E. Guiney, Casos de uso: requisitos en contexto, Nueva York: ACM Prensa, 2000.
- [13] Cockburn, A., Redacción de casos de uso eficaces, Boston, MA: Addison-Wesley, 2001.
- [14] Korson, T., "The Misuse of Use Cases: Management Requisitos", documento técnico, copyright Korson-McGregor, 2000, en www.korson-mcgregor.com/publications/korson/Korson9803om.htm.
- [15] Gamma, E., et al., Patrones de diseño, Reading, MA: Addison-Wesley, 1995.
- [16] Publicaciones de estándares federales de procesamiento de información (FIPS PUBS) 183, "Definición de integración para el modelado de funciones (IDEF0)". Este estándar describe el lenguaje de modelado IDEF0 (semántica y sintaxis) y las reglas y técnicas asociadas para desarrollar representaciones gráficas estructuradas de un sistema o empresa. El uso de este estándar permite la construcción de modelos que comprenden funciones del sistema (actividades, acciones, procesos, operaciones), relaciones funcionales y datos (información u objetos) que respaldan la integración de sistemas. Disponible en www.itl.nist.gov/fipspubs/idef02.doc.
- [17] IEEE, IEEE 1320.1. "Estándar IEEE para lenguaje de modelado funcional: sintaxis y semántica para IDEF0". Sociedad de Computación IEEE, 1998.
- [18] Feldmann, CG, La guía práctica para la reingeniería de procesos de negocios utilizando IDEF0, Nueva York: Dorset House, 1998.
- [19] Paulk, MC, "Using the Software CMM with Good Judgment", Software Quality Professional 1(3) (1999), en www.sei.cmu.edu/publications/articles/paulk/judgment.html.

- [20] Hadden, R., "¿Cuán escalables son las prácticas clave de CMM?" CrossTalk (abril de 1998): 18–23.
Véase también www.ppc.com.
- [21] Brodman, JG y DL Johnson, The LOGOS Tailored CMM for Small Businesses, Small Organizations and Small Projects, LOGOS International, Inc., en www.tiac.net/usuarios/johnson.
- [22] Gilb, T., Principios de gestión de ingeniería de software, Harlow, Reino Unido: Addison-Wesley, 1988.
- [23] Gilb, T., "Impact Estimation Tables: Understanding Complex Technology Quantitatively", documento técnico, noviembre de 1997, en www.gilb.com.
- [24] SPR, sitio web, en www.spr.com.
- [25] Garmus, D. y D. Herron, Análisis de puntos de función: prácticas de medición para proyectos de software exitosos. Boston, MA: Addison-Wesley, 2001.
- [26] IFPUG, sitio web, en www.ifpug.org.
- [27] Jones, C., Evaluaciones de software, puntos de referencia y mejores prácticas. Boston, Massachusetts: Addison-Wesley, 2000.
- [28] Humphrey, WS, Introducción al proceso de software personal, Reading, MA: Addison-Wesley, 1997.
- [29] Humphrey, WS, Introducción al proceso de software de equipo, Reading, MA: Addison-Wesley, 2000.
- [30] Gilb, T. y D. Graham, Inspección de software, Boston, MA: Addison-Wesley, 1993.
- [31] Daugherty, T. (ed.), Conceptos fundamentales para el ingeniero de calidad de software, Milwaukee, WI: ASQ Quality Press, 2002.
- [32] Sabourin, R., sitio web, en www.amibug.com/index.shtml.
- [33] Walton, M., El método de gestión de Deming, Nueva York: The Putnam Publishing Grupo, 1986.
- [34] Jones, C., Calidad del software: análisis y directrices para el éxito, Londres: Prensa internacional de computadoras Thomson, 1997.
- [35] Smith, PG y DG Reinertsen, Desarrollo de productos en la mitad del tiempo, 2^a ed., Nueva York: John Wiley & Sons, Inc., 1998.
- [36] Bicknell, BA y KD Bicknell, La hoja de ruta hacia el éxito repetible: uso de QFD para implementar el cambio, Boca Ratón: CRC Press, 1995.
- [37] Grady, JO, Validación y verificación del sistema, Boca Ratón: CRC Press, 1997.
- [38] Boehm, B. y R. Turner, "Observaciones sobre el equilibrio de la disciplina y la agilidad", en Balancing Agility and Discipline: A Guide to the Perplexed, Boston, MA: Addison-Wesley, 2003.
- [39] Alexander, I., sitio web, en easyweb.easynet.co.uk/~iany/index.htm.
- [40] Clark, BK, "Efectos de la madurez del proceso en el esfuerzo de desarrollo", en www.ralphyoung.net/goodarticles.
- [41] Brodman, JG y DL Johnson, "Retorno de la inversión (ROI) de la mejora de procesos de software medido por la industria estadounidense", Práctica y mejora de procesos de software. Sussex, Reino Unido: John Wiley & Sons Ltd., 1995, 35–47.
- [42] Butler, K., "Los beneficios económicos de la mejora de procesos de software", CrossTalk (1995): 28–35.

- [43] Dion, R., "Mejora de procesos y balance corporativo", IEEE Software (octubre de 1993): 28–35.
- [44] Herbsleb, J., et al., Beneficios de la mejora de procesos de software basado en CMM: resultados iniciales. Informe Técnico CMU/SEI-94-TR-013. Pittsburgh, PA: Instituto de Ingeniería de Software, agosto de 1994.
- [45] Humphrey, WS, et al., "Mejora de procesos de software en Hughes Aircraft", Software IEEE (agosto de 1991): 11–23.
- [46] McGibbon, T., "Un caso de negocio para la mejora de procesos de software revisado: medición del retorno de la inversión desde la ingeniería y la gestión de software", número de contrato SP0700-98-4000, Centro de análisis y datos para software (DACS), ITT Industries, División de Ciencias e Ingeniería Avanzada, Roma, Nueva York, 30 de septiembre de 1999, en www.dacs.dtic.mil/techs/roisp12.
- [47] DACS, Informes técnicos de DACS, en www.dacs.dtic.mil/techs/tr.shtml.
- [48] Project Management Institute, Guía de los conocimientos sobre gestión de proyectos (PMBOK), 1996.
- [49] Harroff, Noel, "Estructura de desglose del trabajo (WBS)", en www.nnh.com/ev/wbs2.html.
- [50] Wideman, Max, "Comparing PRINCE with PMBOK", AEW Services, Vancouver BC, Canadá, en www.pfforum.org/library/papers/Prince2vsGuide3.htm, 2002.
- [51] SEI, "Identificación de riesgos basada en taxonomía", Informe Técnico CMU/SEI-93-TR-6. Pittsburgh, PA: SEI, junio de 1993, en www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf.
- [52] Sommerville, I., Ingeniería de software, 6^a ed., Reading, MA: Addison-Wesley, 2001.
- [53] Boehm, BW, Economía de la ingeniería de software, Englewood Cliffs, Nueva Jersey: Prentice Salón, 1981.

Machine Translated by Google

CAPÍTULO

8

Contenido

[Impulsores empresariales para la calidad](#)[El papel de la gerencia](#)[Principios rectores](#)[Gestión de prioridades](#)[Los componentes de un Enfoque de calidad integrado](#)[Mejora de calidad](#)[Técnicas](#)[El ciclo PDCA](#)[Cómo diseñar un proceso](#)[Trabajo en equipo](#)[Resumen](#)[Estudio de caso: un ejemplo de](#)[Mejora de calidad](#)[Desviado](#)[Referencias](#)

Un enfoque de calidad integrado

A lo largo de los años se han propuesto muchos "enfoques de calidad", como la Gestión de la Calidad Total (TQM), los Criterios del Premio Malcolm Baldrige, Six Sigma, Quality Is Free, Zero Defects, el Balanced Scorecard, estándares desarrollados por la Organización Internacional de Normalización (ISO) y otros. Uno de los problemas al implementar programas de calidad ha sido convencer a la gerencia y a la organización de que valen la pena. Otro problema es que, a menudo, las iniciativas de calidad son difíciles de sostener.

Según mi experiencia, la calidad en una organización o en un proyecto es más una forma de vida que un programa independiente, y la elección del modelo de calidad a utilizar es menos importante que centrarse en satisfacer las necesidades reales de los clientes. La calidad es nuestra forma de trabajar, no una función separada. Este capítulo es importante para la RA porque existe una conexión directa entre satisfacer las necesidades reales de los clientes y un enfoque de calidad. Hay un conjunto de impulsores de negocio (necesidades y expectativas de alto nivel del cliente) que en realidad son requisitos de alto nivel del cliente, que los RA deben abordar. La gestión tiene un papel crítico en relación con la calidad. Si la dirección no valora la calidad, la calidad no se producirá. Existe un conjunto de principios en cualquier organización que sirven para proporcionar valores rectores para el trabajo que se realiza en la organización. Esto requiere un conjunto de técnicas de mejora de la calidad (que se describen a continuación). Los individuos son responsables de la calidad de sus productos y servicios. Este capítulo explica un enfoque de calidad integrado que facilita y apoya el trabajo de la AR. Mostrará que es necesario un proceso de requisitos eficaz para tener un enfoque de calidad integrado y que se requiere un enfoque de calidad integrado para que el proceso funcione mejor. Por enfoque de calidad integrada me refiero al uso de técnicas de mejora de la calidad que se incorporan al trabajo diario realizado en un proyecto y en una organización con el objetivo de lograr la satisfacción del cliente. Esto es más fa-

y una expectativa de gestión (a través de sus valores y principios declarados) que respalda el trabajo del analista. Una advertencia es que no importa cuán comprometidos estén las personas o los equipos con la calidad y con el uso efectivo de técnicas de mejora de la calidad, es posible que no tengan éxito si otros equipos, miembros de su propio equipo y la gerencia no comparten ese compromiso.

Impulsores empresariales para la calidad

Existe un conjunto de factores que pueden considerarse requisitos de alto nivel del cliente, pero que a menudo no se expresan. Estos incluyen lo siguiente:

- Éxito empresarial continuo;
- Productos o servicios de alta calidad;
- Cumplir con los requisitos del cliente;
- Reducción de costos;
- Fidelización de clientes y empleados;
- Rendimiento mejorado;
- Eliminación de defectos;
- Eficiencia;
- Tiempo de ciclo reducido;
- Soluciones innovadoras.

Es importante que los proyectos y las organizaciones consideren estas necesidades de los clientes.

El papel de la gerencia

En las organizaciones de alto rendimiento, los objetivos estratégicos y los impulsores comerciales están vinculados a los objetivos y actividades de mejora de procesos. Los valores y principios rectores se documentan y comunican, y la alta dirección establece sus expectativas de que todos los niveles de la organización respeten estos principios en sus hábitos de trabajo diarios. En relación con la calidad y la mejora de procesos, el papel de la dirección es hacer lo siguiente:

- Definir objetivos estratégicos, como ingresos, ganancias, satisfacción del cliente, y retención de empleados;
- Habilitar la mejora de procesos (PI):
 - Actuar como patrocinador y defensor de PI;
 - Verbalizar el valor y los resultados de PI;
 - Proporcionar recursos para PI;

Establecer un ciclo de mejora.

Establecer valores, como los siguientes:

Centrarse en arreglar el proceso, no las personas; Medir

y actualizar periódicamente los procesos en uso; Promover la mejora continua como esencial para mantener y

hacer crecer la organización;

Promover la visión de que las personas de la organización son lo más importante. recurso tantante.

Tomar decisiones oportunas;

Determinar las áreas que (a) son más importantes para los clientes y (b) necesitan mejora, como por ejemplo:

Proporcionar soluciones competitivas;

Apropiarse de los proyectos; Ser flexible en cuanto a las modalidades de trabajo; Responder bien a los cambios;

Ser receptivo;

Emplear personas de alta calidad;

Proporcionar un buen valor;

Cumplir los compromisos;

Escuchar bien y comprender las necesidades;

Ser competente.

Principios rectores

En cualquier organización, un conjunto de principios rectores establece valores sobre cómo se deben hacer las cosas. Estos principios pueden o no estar articulados. En algunas organizaciones, el valor rector es que simplemente llevarse bien está bien. En otros, existe un conjunto de valores que sirven para proporcionar estándares elevados y eficaces sobre cómo se deben hacer las cosas, por ejemplo:

La satisfacción del cliente es imperativa para nuestra existencia continua.

Gestionaremos por hechos (utilizando datos) en lugar de por intuición o por la asiento de nuestros pantalones.

Contamos con un conjunto de reglas de conducta utilizadas por las personas de nuestra organización que reflejan respeto por las personas.

La mejora continua es fundamental para todos nuestros procesos.

En mi experiencia, tener un conjunto de principios rectores que la gerencia y las personas de la organización practican y valoran crean un sentido de propósito para los empleados y enfatizan que los empleados son clave.

partes interesadas en el éxito de la empresa. Como resultado, las personas se sienten valoradas y motivadas a contribuir con sus mejores esfuerzos.

Gestión de prioridades

El proceso utilizado por la dirección para decidir lo que quiere lograr se muestra en la Figura 8.1. Tenga en cuenta que el proceso utiliza equipos de QI para abordar actividades o problemas prioritarios y desarrollar planes y recomendaciones para su revisión y aprobación por parte de los ejecutivos responsables. Este enfoque aprovecha la experiencia, los conocimientos y el compromiso de los empleados.

Los componentes de un enfoque integrado de calidad

Los componentes de un enfoque de calidad integrado pueden describirse como gestión, clientes, proyectos y tareas, equipos de calidad y garantía de calidad. La Figura 8.2 describe cómo estos componentes trabajan juntos. Tenga en cuenta lo siguiente:

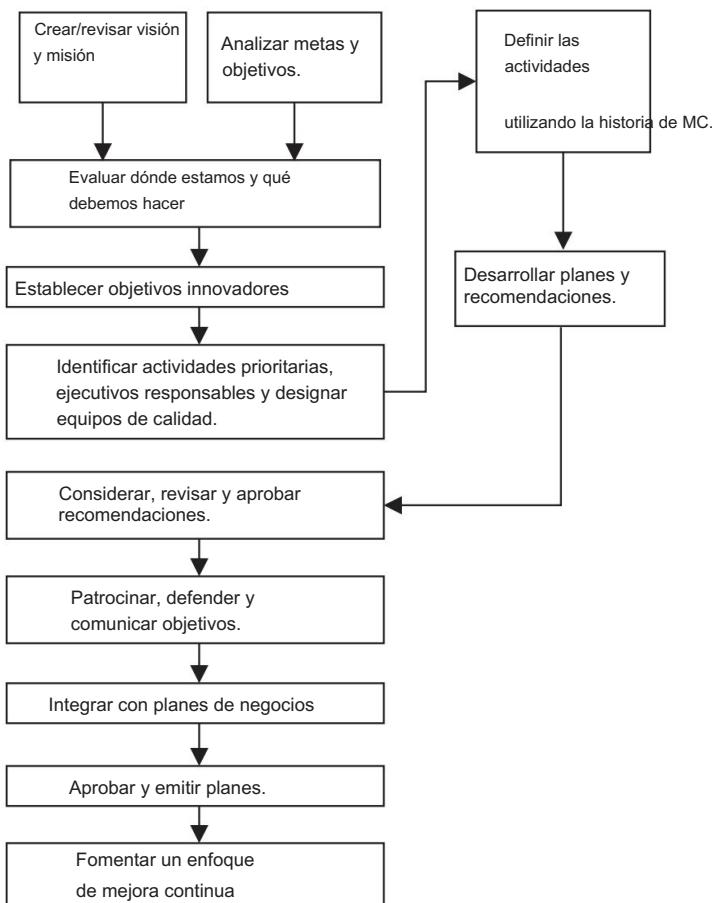


Figura 8.1 Decidir lo que queremos lograr.

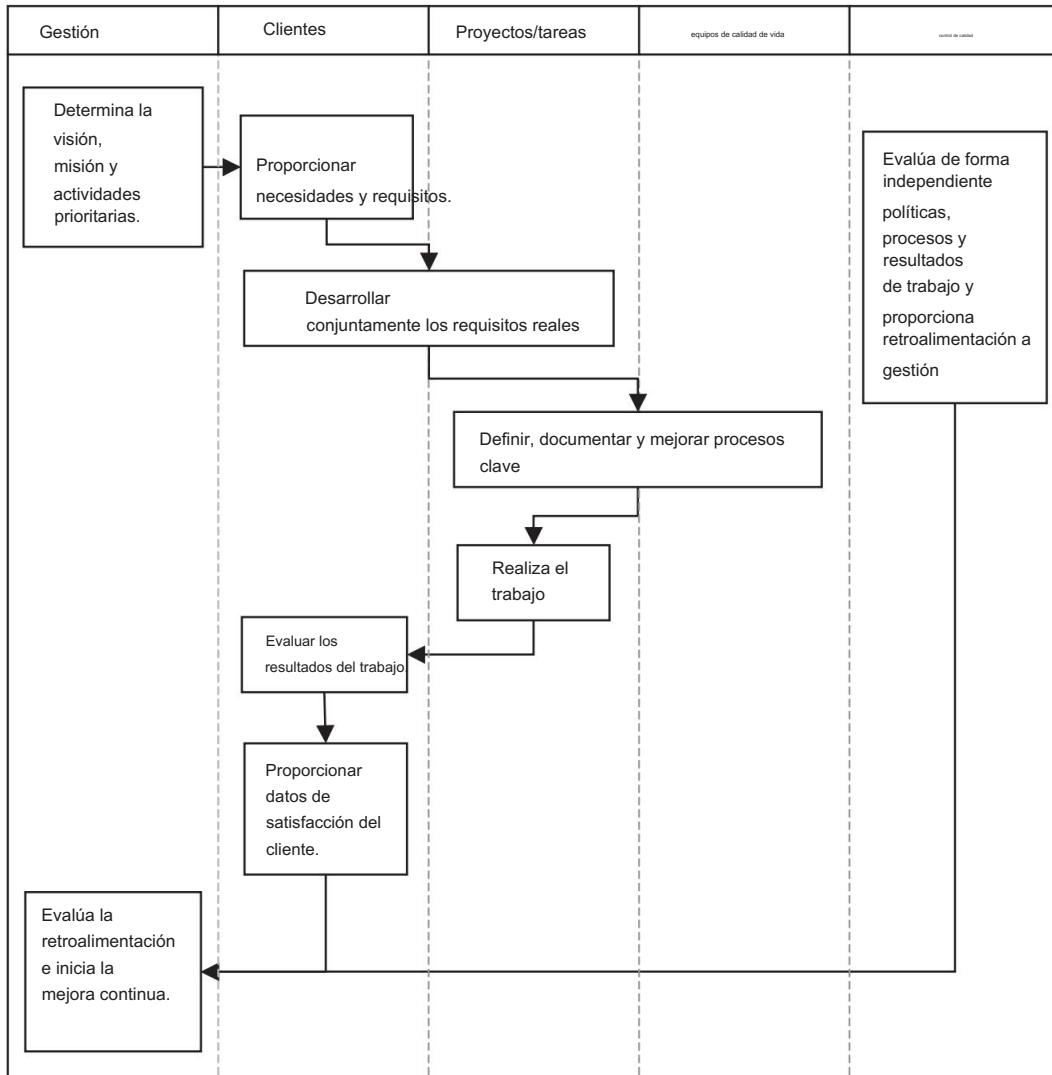


Figura 8.2 Cómo funcionan juntos los componentes de un enfoque integrado de calidad.

Un aspecto crítico es el uso del equipo conjunto para desarrollar los requisitos reales, como se analiza en el Capítulo 1.

Los datos de satisfacción de los empleados son otra forma de retroalimentación proporcionada a gestión que ayuda a guiar la organización.

Se pueden utilizar estándares como el CMM para medir la calidad organizacional. eficacia.

Técnicas de mejora de la calidad

Independientemente del tamaño de sus proyectos, los buenos PM pueden realizar mejoras en proyectos locales bajo su propia iniciativa utilizando uno o más de los

técnicas de mejora que se describen a continuación. Pero, si una empresa quiere mejorar la calidad de sus productos y servicios, el enfoque más eficaz es hacerlo desde arriba e implementar estas técnicas en todos los proyectos. La mejor manera de gestionar la mejora de la calidad es establecer un consejo de gestión de la calidad (QMB, por sus siglas en inglés) como un equipo directivo que pueda implementar estas ideas desde arriba y, así, guiar a la empresa por el camino hacia la calidad con el ejemplo. En el QMB, los gerentes pueden determinar políticas organizacionales apropiadas y objetivos de calidad e PI. Para lograr sus objetivos, implementan planes de mejora y proporcionan los recursos y habilidades necesarios.

Realizan un seguimiento del estado de las mejoras, recompensan a los equipos y a las personas por sus éxitos e identifican formas de transferir las lecciones aprendidas y las mejoras a otras partes de la organización. Independientemente del tamaño de la organización, el QMB establece su agenda a través de objetivos de calidad fáciles de entender y proporciona dirección con respecto a la implementación de las siguientes técnicas de mejora de la calidad:

Modelos de mejora de calidad y mejora de procesos: considere adoptar un marco (como CMM o CMMI) como estándar para sistemas o ingeniería de software y realizar evaluaciones anuales de la situación actual en comparación con el estándar (ver Figura 8.3). La experiencia ha demostrado que las CMM, en uso desde 1987, permiten que un proyecto u organización de ingeniería de sistemas o software perciba cómo se compara.

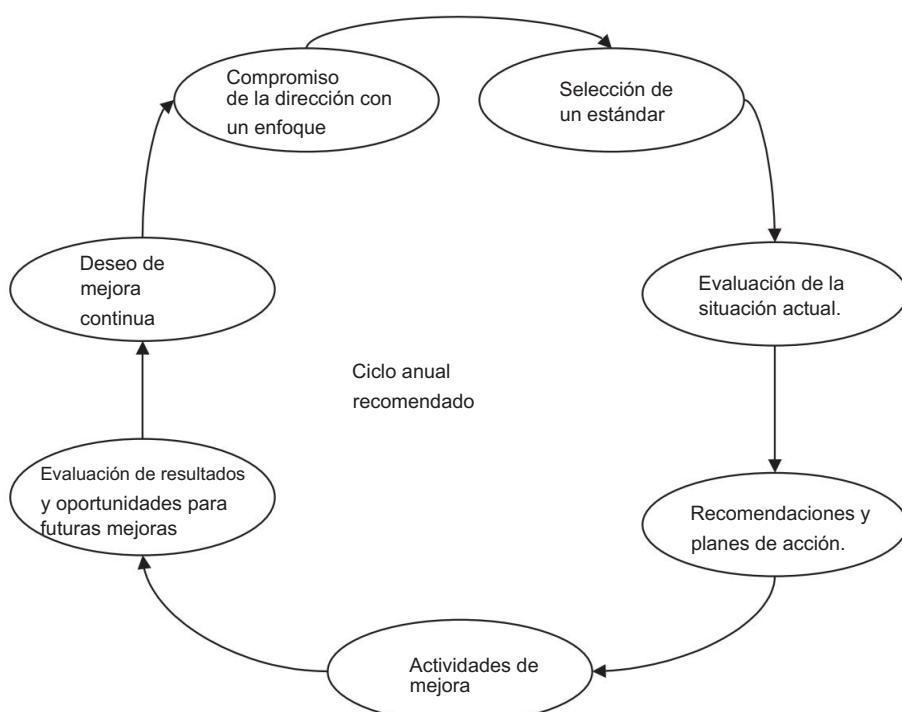


Figura 8.3 El ciclo de mejora de la calidad.

contra lo que la industria considera una mejor práctica y si las mejoras que ha implementado están teniendo el efecto deseado. Usar los resultados para identificar áreas prioritarias para actividades de mejora y iniciar iniciativas de mejora continua, incluidos equipos de calidad. donde corresponda. Aunque puede requerir algo de planificación, recursos y tiempo para determinar cómo la organización encaja con el proceso elegido. modelo, esta inversión proporcionará información valiosa que puede ayudar a orientar las mejoras. Podría ser más fácil para las organizaciones más grandes absorber el gasto de evaluaciones y mejoras, pero de menor tamaño. Las organizaciones pueden ver su programa de mejora en términos estratégicos. términos y buscar otras formas de implementar mejoras. Para Por ejemplo, es posible que deseen asociarse con otros contratistas o clientes para lograr sus objetivos.

Formación: la formación periódica y adecuada es siempre una inversión valiosa para empresas de cualquier tamaño. Los clientes exigen que los trabajadores tengan las habilidades tecnológicas de última generación y los empleados quieren mejorar su desarrollo y desempeño profesional. Si las organizaciones esperan continuar satisfaciendo las necesidades cambiantes de sus clientes y las de sus empleados deseo de crecimiento personal y profesional, entonces la gerencia debe tomar responsabilidad de garantizar que esto suceda. Las organizaciones más pequeñas o medianas pueden tener dificultades (por falta de recursos) para establecer una programa de formación organizacional. Asociarse con colegios o universidades locales para obtener tarifas más bajas puede ser una opción que podrían explorar.

Equipos de MC que utilizan la historia de MC: aunque siempre conscientes del fondo. En esta línea, las grandes empresas tienen más margen de maniobra en términos de recursos para personal y financiar equipos de QI. Debido a que tienen menos personal y presupuestos más pequeños, los proyectos o empresas pequeñas y medianas deben ser juiciosos. en el establecimiento de equipos. Al mismo tiempo, porque deben operar sus negocios y responder a las demandas de los clientes y tratar con problemas crónicos de calidad, deberían establecer equipos de QI para atacar. Sólo los que los altos directivos consideran los problemas más críticos de la empresa. En otras palabras, las organizaciones más pequeñas deberían establecer equipos para abordar aquellos problemas que obstaculizan su capacidad para hacer negocios o satisfacer las demandas de los clientes. (Consulte a continuación para obtener más información sobre el QI proceso de historia que puede ser utilizado por los equipos de QI para fomentar la continuidad mejora.)

Encuestas de satisfacción del cliente (p. ej., por teléfono): considere proporcionar una mecanismo para abordar inmediatamente la insatisfacción del cliente, como "procedimientos de alerta roja" para escalar inquietudes, abordarlas y brindar retroalimentación al cliente. Grandes proyectos o empresas han establecido mecanismos y herramientas para rastrear y gestionar la satisfacción del cliente. asuntos. Los administradores pueden aprovechar los recursos existentes para recopilar y Analizar la información y utilizarla para mejorar o conseguir nuevos negocios. Aunque es posible que no tengan la ventaja de un enfoque corporativo

o infraestructura, las empresas más pequeñas también tienen una ventaja. Porque pueden trabajar con un grupo más pequeño de clientes, es más fácil tomar la enfoque personal y discutir problemas uno a uno con los clientes y para desarrollar acciones correctivas inmediatas. Independientemente del tamaño de la organización, si los gerentes no realizan un seguimiento o control de la satisfacción de sus clientes problemas, pagarán su precio en financiación reducida, contratos cancelados, o premios a los competidores.

Encuestas de satisfacción de los empleados: conocer las inquietudes de los empleados a través de datos objetivos y actuar sobre los resultados genera lealtad de los empleados y mejora la retención. Al igual que con las encuestas de satisfacción del cliente, mayores empresas utilizan recursos considerables para abordar los problemas de los empleados y para retener su fuerza laboral. Saben que estos recursos tienen pies y Puede salir en cualquier momento para encontrar un mejor ambiente de trabajo u oportunidades de avance. Esto también es válido para las empresas más pequeñas, pero pueden ser más críticos para su negocio si unos pocos trabajadores calificados El núcleo de su experiencia en cualquier área dada se une a la competencia. Los gerentes de pequeñas empresas harían bien en desarrollar un informe breve y Encuesta de 20 preguntas para saber qué piensa su personal, para determinar qué funciona bien en la empresa y qué no, y identificar áreas de mejora. No es necesario que una encuesta de este tipo sea científica, sino que sirva de base para las decisiones de gestión.

Control de calidad: en la mayoría de las grandes empresas, tener una visión objetiva independiente sobre el cumplimiento y uso de políticas y procesos en la organización Proporciona comentarios valiosos sobre la mejora de la calidad y el proceso. esfuerzos de mejora o señala otros problemas que podrían no haber sido su atención inmediata. Esto hace posible tener un control de calidad capacitado. personal disponible para apoyar proyectos o equipos de cualquier tamaño como una función matricial. Es posible que las organizaciones más pequeñas no tengan el personal capacitado o presupuesto para apoyar a un personal de control de calidad a tiempo completo o una organización matricial desde a los que pueden asignar control de calidad. En esos casos, se recomienda que el Todo el equipo del proyecto adopta una estrategia de equipo de calidad, donde todos los miembros del equipo son responsables de la calidad del trabajo que realizan. y los productos y servicios que ofrecen. En este entorno, cada Un miembro del personal debe realizar revisiones o auditorías del producto o servicio de trabajo de otro miembro del equipo y del proceso utilizado. Este enfoque Requiere que cada miembro del equipo esté capacitado para su función de control de calidad y que exista un sistema de acciones correctivas que rastree el estado de los problemas encontrados en las revisiones de control de calidad y mantenga informada a la gerencia. Gerentes debe tener especial cuidado para garantizar que los revisores puedan demostrar su objetividad en su rol como QA (un requisito principal del CMMI) y que no están directamente involucrados en el proceso o producto que están revisando.

Diseño, gestión y mejora de procesos: Un proceso es un conjunto de actividades que dan como resultado la realización de una tarea o el logro de un resultado. Las organizaciones de cualquier tamaño deberían utilizar el proceso como uno de los

pilares fundamentales de su labor. Los proyectos más grandes y complejos requieren más detalles en sus procesos documentados y muestran los roles y responsabilidades de todos los grupos que participan en el proceso. Por razones obvias, lograr que grandes grupos de personas trabajen juntas para alcanzar objetivos compartidos puede ser un desafío mayor. Por otro lado, los equipos de proyectos más pequeños tienen una ventaja, ya que pueden depender de menos detalles en su proceso documentado. Por ejemplo, pueden utilizar listas de verificación o diagramas de flujo de procesos simples. Tener menos personas en el equipo del proyecto hace que sea más fácil determinar cuál es el resultado deseado del proceso, qué entradas y salidas se requieren y los pasos específicos del proceso que deben seguirse. Tener equipos pequeños también facilita la capacitación y la realización de los cambios deseados.

Si no puedes describir lo que estás haciendo como un proceso, no sabes lo que estás haciendo.

—William Edwards Deming, consultor de gestión, 1900–1993.

Monitorear el desempeño a través de métricas: los gerentes deben tomar decisiones basadas en datos. A veces estos datos pueden ser cualitativos o cuantitativos. Para que una organización de cualquier tamaño mejore, debe tener datos cuantitativos en los que basar sus mejoras. Es un gerente imprudente el que decide gastar recursos para implementar una mejora de la calidad cuando no sabe si dicha mejora es necesaria. Para mejorar la calidad y los procesos, los gerentes deben establecer objetivos razonables (que tengan buenas posibilidades de lograr) e identificar medidas que puedan utilizar para determinar si han cumplido o no esa meta. Ejemplos de métricas potencialmente útiles incluyen tasas de éxito empresarial, índices de satisfacción del cliente y un índice de fidelidad del cliente.

Esto último puede generarse de forma tan sencilla como formular tres preguntas: (1) ¿Cómo califica nuestra calidad? (2) ¿Cuál es la probabilidad de que su negocio continúe? (3) ¿Cuál es la probabilidad de que nos recomiende a un nuevo cliente?

Técnicas de MC: Las técnicas de MC como la lluvia de ideas, la votación múltiple, el análisis de Pareto, el análisis de barreras y ayudas, los planes de acción, el análisis de causa y efecto, las listas de verificación, la historia de la MC y el PDCA (que se analiza a continuación) son fáciles de aprender y de un valor incalculable en un organización con visión de futuro.1 Estas técnicas pueden funcionar bien en proyectos u organizaciones de cualquier tamaño, siempre y cuando el grupo esté capacitado para utilizar la técnica y los resultados del ejercicio.

Historia de QI: La historia de QI, desarrollada por Qualtec Quality Services (ahora parte de Six Sigma Qualtec), proporciona una estructura para abordar actividades y problemas prioritarios. Como se mencionó anteriormente, debido a que tienen menos

1. Una guía excelente y económica para utilizar estas técnicas es QI Story: Herramientas y técnicas, una guía para la resolución de problemas de Six Sigma Qualtec, [1]. Otro recurso para materiales de mejora organizacional es GOAL/QPC, disponible en goalqpc.com o llamando al (800) 643-4316.

los recursos, los proyectos más pequeños o las empresas deben ser prudentes en el uso de la historia de la MC para resolver problemas; deben identificar los problemas que serán más rentables de resolver. Un conjunto modificado de pasos es el siguiente:

1. Identificar el motivo de la mejora.

Determinar la naturaleza del problema.

Recopilar datos.

Identificar procesos clave.

Desarrollar un plan y un cronograma.

2. Analizar la situación actual.

Identificar los requisitos reales de los clientes.

Establecer un objetivo de mejora.

3. Realizar análisis.

Identificar las probables causas fundamentales del problema.

Seleccione las causas fundamentales que parezcan tener el mayor impacto.

Verificar las causas raíz seleccionadas con datos.

4. Seleccionar contramedidas que ataquen las causas fundamentales verificadas.

Evaluando si las contramedidas lograrán lo siguiente:

Abordar las causas fundamentales verificadas;

Impactar los requisitos de los clientes;

Demostrar ser rentable.

5. Desarrollar un plan de acción para implementar las contramedidas seleccionadas. Obtener

la aprobación de la dirección.

Coordinar con las partes interesadas para obtener apoyo y cooperación.

6. Implementar las contramedidas.

7. Mida los resultados.

¿Funcionaron las contramedidas?

¿Están mejorando las cosas?

¿Se están impactando las causas fundamentales?

Evaluando los resultados respecto al objetivo de mejora.

Implementar contramedidas adicionales si es necesario.

8. Estandarizar un enfoque basado en los resultados. ¿Qué se

puede cambiar para garantizar que el problema no vuelve a ocurrir (por ejemplo, una política, un procedimiento, un proceso de trabajo, una norma o una capacitación nuevos o revisados)?

9. Considere qué lecciones se han aprendido al realizar el esfuerzo de mejora de la calidad.

¿ Deben abordarse los problemas relacionados? ¿ Debería revisarse el enfoque para realizar la historia de MC?

El ciclo PDCA

Un paradigma popular y útil utilizado para la mejora de la calidad es el ciclo PDCA [2] mencionado en el Capítulo 3 en relación con la evaluación del valor y la utilidad de las reuniones. La idea es planificar el enfoque, implementarlo (“hacerlo”), verificar cómo funcionan las cosas, actuar sobre los resultados de esa verificación y continuar el ciclo. El ciclo PDCA se muestra en la Figura 8.4.

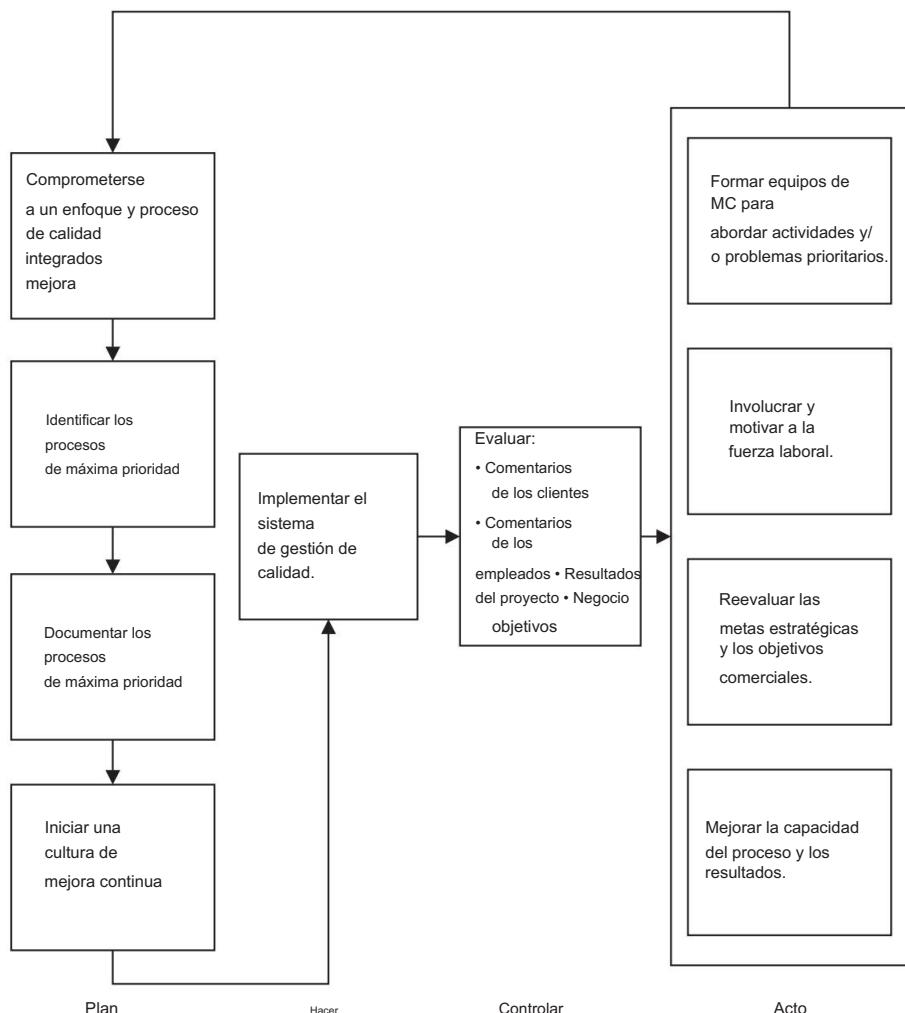


Figura 8.4 El ciclo PDCA.

Cómo diseñar un proceso

Una habilidad crítica que necesita el RA es la capacidad de diseñar y mejorar procesos, más específicamente, facilitar el diseño de procesos y las sesiones de mejora. Recuerde que definimos un proceso como un conjunto de actividades que resultan en la realización de una tarea o el logro de un resultado. Por ejemplo, en este libro nos hemos referido al proceso de requisitos, un conjunto de actividades del ciclo de vida completo del sistema que incluye lo siguiente:

- Identificación de requisitos;
- Comprender las necesidades y expectativas del cliente;
- Aclarar y reformular los requisitos (evolucionar los requisitos reales);

- Analizar los requisitos;
- Definir los requisitos;
- Especificación de los requisitos;
- Priorizar los requisitos;
- Requisitos derivados;
- Requisitos de partición;
- Asignación de requisitos;
- Requisitos de seguimiento;
- Gestión de requisitos;
- Requisitos de prueba y verificación;
- Validación de requisitos.

El diseño de procesos y la mejora de procesos son actividades que hacen lo siguiente:

Involucran a las partes interesadas (aquellos que tienen interés) en decidir cómo se deben hacer las cosas, obteniendo así su aceptación para la implementación, el uso y la mejora continua del proceso.

Permiten que un proyecto u organización sea cada vez más competente. Una vez que un proceso está documentado, todos pueden entenderlo y se puede realizar repetidamente de la misma manera con los mismos resultados. Además, se pueden sugerir, discutir e incorporar mejoras.

Diseñar un proceso es sencillo. Reúna en una sala a algunas personas inteligentes que estén familiarizadas con el proceso a documentar. Lo mejor es que todos los grupos de partes interesadas estén representados. Usando una pizarra o una hoja de papel grande, cree la plantilla proporcionada en la Figura 8.5.

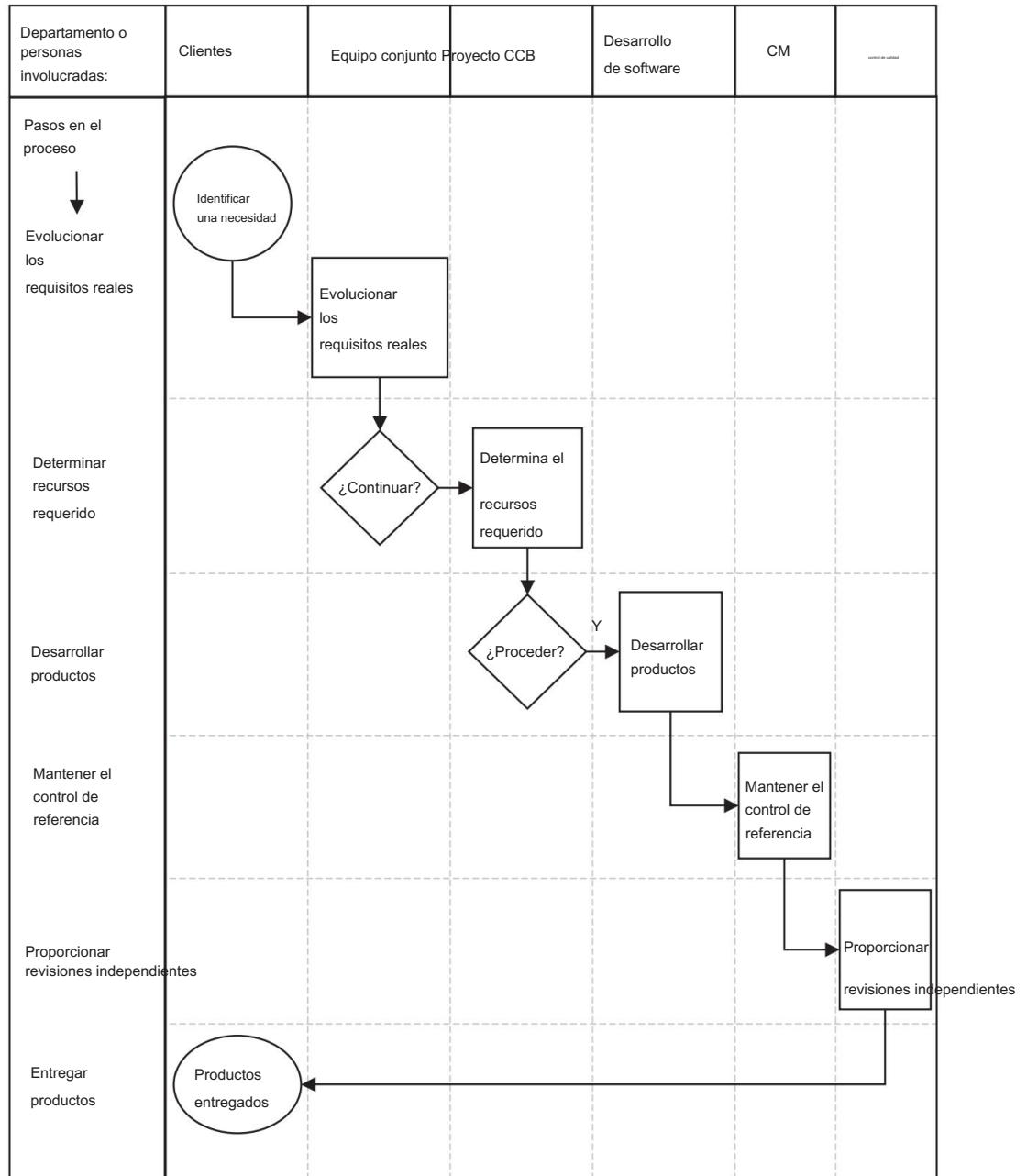


Figura 8.5 Plantilla de diagrama de flujo de diseño de procesos con proceso de desarrollo de productos simplificado.

En la parte superior, enumere los departamentos, organizaciones o personas involucradas en el proceso. A lo largo del eje vertical, enumere los pasos del proceso (numere o describa cada paso). Usando pequeñas hojas adhesivas, pida a las personas que describan qué se debe hacer para realizar las tareas en el proceso en las hojas adhesivas y colóquelas en los lugares apropiados de la plantilla. Un enfoque alternativo es documentar lo que realmente se hace en la situación actual.

proceso como punto de partida para la mejora. Una vez que se hayan identificado la mayoría de las actividades, haga una verificación final para asegurarse de que estén ensambladas en el orden en que deben realizarse y asignadas a la organización o persona que debe realizarlas. Más adelante, utilice símbolos de diagrama de flujo estándar para conectarlos. He proporcionado un proceso de desarrollo de producto simplificado en la plantilla para ayudarle a comprender el concepto. Descubrirá que algunas de las actividades del proceso requerirán una definición más detallada; en este caso, considere diseñar subprocessos para capturar y documentar comprensiones más detalladas del enfoque del “debería ser”.

Como se enfatiza en el Capítulo 3, la AR debe ser competente en el diseño y desarrollo de un proceso. Aprovecha ahora, mientras lo piensas, para diseñar un proceso. Seleccione una actividad importante relacionada con sus responsabilidades laborales actuales que aún no esté documentada. Invite de tres a seis compañeros de trabajo que tengan mucho conocimiento sobre la actividad o que sean partes interesadas a participar con usted en un taller de diseño de procesos. Transcriba la plantilla proporcionada en la Figura 8.5 (sin el diagrama de flujo y sus pasos asociados) en una pizarra. En una libreta grande a un lado, acuerden el nombre del proceso. Luego, piense detenidamente en los objetivos del proceso: ¿qué se supone que debe lograr el proceso? Será útil visualizar los resultados del proceso. Documentar (anotar) los objetivos del proceso. Luego, enumere las partes interesadas del proceso (cualquiera que tenga interés en el proceso o responsabilidades relacionadas con la finalización del proceso).

Usando notas Post-it® de 3 × 5 pulgadas , escriba los nombres de los grupos de partes interesadas y de los grupos de proyectos involucrados en el proceso a documentar (un grupo por nota Post-it®) y colóquelos en la parte superior de la pizarra. Nuestra costumbre es colocar al cliente del proceso en el lado izquierdo (dejar espacio para una columna vertical que será un conjunto de los números sucesivos de los pasos del proceso (empezando por “1”) o nombres especificando estos pasos. Normalmente, los nombres de los grupos que están más involucrados en el proceso se colocan después de los de los clientes, y los nombres de los grupos menos involucrados en el proceso hacia o en el lado derecho. Luego, invite a los participantes a su taller de diseño de procesos para sugerir las tareas que deben realizarse para realizar el proceso. Inicialmente, simplemente enumere cada tarea en una nota Post-it® separada y colóquelas en la pizarra (en cualquier lugar). A medida que avanza con la identificación de tareas , comience a organizar los pasos de acuerdo con cómo se debe realizar el proceso. [Un enfoque alternativo es desarrollar el diagrama de flujo del proceso basándose en la forma existente en que se realiza el proceso (el “proceso tal como está”).] No se preocupe Aún no se han comprometido a conectar los pasos del proceso con líneas o a determinar puntos de decisión; estos refinamientos pueden realizarse fuera de línea, después del taller, por la persona que tiene la responsabilidad principal de ese proceso (nos referimos a esta persona como el proceso). dueño). Una vez que esté satisfecho de haber identificado la mayoría o todas las tareas que deben realizarse en el proceso, refine su organización de las tareas en un flujo lógico (el diagrama de flujo del proceso). Es posible que descubra que es necesario definir una o más tareas con más detalle; puede optar por abordar este subprocesso en un taller posterior o puede decidir que es tan importante que debe definirse primero, antes de completar

flujo que comenzó inicialmente. Sea muy flexible. ¡Divertirse! Disfruten unos de otros. Aprendamos unos de otros. Considere invitar a otras partes interesadas a unirse a su agrupar o crear subgrupos para definir subprocesos (a veces denominados microprocesos del macroporceso). Otro enfoque, si usted o alguien otra persona está muy familiarizada con el proceso, es que esa persona diseñe un borrador de el proceso fluye de forma independiente y llevar ese borrador del producto de trabajo a la Taller de diseño de procesos para su refinamiento por parte del grupo con el fin de alcanzar consenso.

Para ver un ejemplo de un diagrama de flujo de proceso de requisitos completo, consulte Prácticas de requisitos eficaces [3, págs. 110-124]. Escribe un PD narrativo para acompañan cada diagrama de flujo (macro y micro). Consulte la Tabla 8.1 para obtener una plantilla de PD. Los PD para los diagramas de flujo referenciados del proceso de requisitos son disponible en mi sitio web (www.ralphyoung.net). El propietario del proceso debe Desarrollar el PD fuera de línea. Pienso que un proceso definido incluye tanto el diagrama de flujo del proceso y el PD narrativo, porque la información proporcionada Ambos documentos son necesarios para comprender el proceso.

Tenga en cuenta que tener un proceso definido y documentado proporciona una base para la mejora continua. A medida que el proceso se realiza o "ejecuta", se pueden identificar y proporcionar al proceso sugerencias para mejorarlo. dueño. El propietario del proceso debe recopilar estas sugerencias y periódicamente Invitar a las partes interesadas a participar en un taller de perfeccionamiento de procesos para considerar sugerencias. Si se decide cambiar ("actualizar") el diagrama de flujo del proceso o PD, el propietario del proceso debe documentar los cambios y publicar un nuevo versión del proceso. Una convención de nomenclatura sugerida es utilizar la Versión 1.1, 1.2, 1.3, etc., para actualizaciones menores y la versión 2.0, 3.0, 4.0, etc. adelante, para actualizaciones importantes. Esto es importante para que todas las partes interesadas puedan identificar y utilizar la versión actual. Se debe enviar un correo electrónico siguiendo el lanzamiento de cada nueva versión para que todas las partes interesadas sean conscientes de lanzamiento actual.

Habiendo asumido la responsabilidad de facilitar un taller de diseño o desarrollo de procesos y de crear un diagrama de flujo de proceso y PD, usted es un RA o ingeniero mejor calificado. (A veces, este tipo de trabajo y actividades relacionadas se denominan ingeniería de procesos). Además, encontrará que el trabajo en equipo se fortalece con estas actividades, porque los compañeros de trabajo Se vuelven cada vez más conscientes de que dependen de otros para trabajar. actividades y para ideas y sugerencias de mejora. Es más, la gente aprenderá mucho sobre cómo se debe realizar el trabajo.

La literatura de la industria informa un retorno de la inversión de 7:1 de las actividades de ingeniería de procesos. Véase [4-8] para varios informes excelentes. Consulte [1, 9] para obtener materiales de ayuda. Introducir y utilizar técnicas de MC. He experimentado este retorno y otros beneficios en la unidad de negocios IT DES de Northrop Grumman, y esto ha fortaleció mi compromiso con la ingeniería de procesos y con el logro madurez del proceso en los últimos doce años [10].

Habiendo facilitado exitosamente el taller de diseño de procesos, sigalo ideamos otro lugar: un almuerzo informal en el que Presenta información sobre varias técnicas de MC simples pero poderosas. (estos se describen en [1]):

Tabla 8.1 Plantilla para un PD

Nombre	Nombre del proceso o subprocesso
información general	
Identificación de proceso	Identificador de proceso único
Meta	Proporcione una breve descripción del propósito y objetivo de la actividad.
Estándares	Identificar los estándares de procesos y productos aplicables, incluidos referencias del modelo de madurez.
Procesos relacionados	Identificar los procesos que están relacionados con este proceso, especialmente si este proceso forma parte de un conjunto que normalmente se considera como un todo. Enumere los procesos que producir insumos o consumir productos de este proceso.
Número de versión	<p>Incluir historial de versiones. Para cada versión, incluya el número de versión, fecha de aprobación y un resumen de los cambios. Por ejemplo, las versiones para esto Si siguió:</p> <p>3.0 amplió SW-CMM a modelos de madurez como estándar; agregado orientación sobre el uso; cambiado a un formato de tabla; orientación adicional sobre indicadores; pie de página modificado para incluir la versión y la fecha (14.10.02)</p> <p>Versión 2.0 en PAL con la etiqueta 2.0.</p> <p>1.1 razones agregadas por las cuales las metas y mediciones de la organización o proyecto podría cambiar</p> <p>Proceso actualizado 1.0 basado en la revisión por pares del 7.1.99 (7.12.99)</p> <p>0.1 versión inicial (1995)</p>
Descripción del cliente	
Cliente	Identificar las partes interesadas internas y externas que se benefician directamente (recibir productos/servicios) a partir de los resultados (outputs) de este proceso.
Requisitos	Enumere cada uno de los requisitos legítimos que se han negociado y acordado con el identificado. Estos requisitos deben seguir las Criterios de RUMBA en el sentido de que deben ser razonables, comprensibles, mensurables, creíbles y alcanzables.
Descripción de la interfaz	
Criterios de entrada	Identificar los criterios que deben cumplirse antes de que la actividad pueda realizarse iniciado. Los criterios podrían decir cómo saber cuándo se puede iniciar un proceso, por ejemplo, al concluir otra actividad o proceso.
Entradas	Identificar los productos de trabajo que se utilizan en cualquier punto del proceso.
Salidas	Identificar los productos de trabajo que se producen durante el proceso.
Criterio de salida	Identificar los criterios que deben cumplirse antes de que la actividad pueda realizarse considerado completo. Los criterios de salida resumen las tareas medibles más destacadas del proceso.
Qué hacer	
Responsabilidades	Describe los grupos que participan en el proceso.
Tareas	Describir las tareas que deben realizarse dentro del proceso. Para la facilidad de referencia, las tareas deben seguir la calidad en el trabajo diario (QIDW) diagrama de proceso al que se hace referencia como exhibición principal. Si el proceso es procesal, describir las tareas en el orden en que deben realizarse, numerando cada paso de la tarea. Poner entre paréntesis al responsable grupo a la izquierda de la tarea, como se muestra a continuación:
(<Grupo participante>) <Descripción de la tarea>	
Herramientas	Utilice verbos de acción para describir las tareas. Referencia por ID de proceso todas las tareas que se describen con más detalle en otra parte. Tenga en cuenta cualquier procedimiento particular, prácticas o métodos que se emplean en cualquier paso.
Recursos	Describa las herramientas sugeridas u obligatorias utilizadas durante cualquier paso del proceso.

Tabla 8.1 Plantilla para un PD (continuación)

Nombre	Nombre del proceso o subproceso
Descripción de la medida	
Indicadores de calidad	Enumere y describa brevemente aquellas mediciones que se utilizarán para rastrear el desempeño (o resultado) de este proceso en términos del producto o servicio entregado al cliente interno o externo. Estos indicadores deben estar estrechamente vinculados a los requisitos válidos del cliente y deben usarse para monitorear el desempeño de todo el proceso. Estas medidas deben ser mensurables, verificables y rentables.
Indicadores de proceso	Enumere y describa brevemente aquellas mediciones que se tomarán en puntos críticos durante el proceso y se utilizarán para rastrear y evaluar la efectividad del proceso en sí. Estas medidas durante el proceso también deben ser mensurables, verificables y rentables.

Nota: En los procesos estándar organizacionales, los indicadores de calidad y de proceso se sugieren, pero no son obligatorios.

Nota: Los proyectos determinan las mediciones del proceso durante la planificación del proyecto y los esfuerzos de mejora del proceso. Para reducir los gastos generales, un proyecto puede designar indicadores de proceso como opcionales, que se recopilarán si se requiere un diagnóstico de problemas.

(Fuente: Craig Hollenbach, miembro de varios de los equipos de desarrollo de productos CMMI que contribuyeron enormemente al desarrollo de los productos CMMI).

Lluvia de ideas;
 Votación múltiple;
 Diagramas de espina de pescado (Ishikawa);
 Diagramas de Pareto;
 PDCA;
 Normas de conducta;
 Prevención de defectos;
 revisiones por pares;
 Inspecciones
 Planes de acción.

Invita a miembros de tu proyecto u organización a comenzar a utilizarlas o a aumentar la frecuencia de uso si estas técnicas ya se conocen y se utilizan.

Su preparación y facilitación de estos talleres es en sí misma una experiencia de crecimiento y desarrollo profesional. He descubierto que cuanto más presento, más fácil es hacerlo.

Considera iniciar un grupo de trabajo sobre requisitos organizacionales (RWG). Invita a participar a quienes realizan trabajos relacionados con requisitos en cualquiera de los proyectos de la organización. El RWG proporciona un mecanismo (una forma de hacer algo o lograr un resultado) para coordinar actividades relacionadas con los requisitos y para que las AR se ayuden entre sí. Consulte la Tabla 8.2 para obtener una lista de las ventajas de tener un RWG y consulte [8] para obtener más información.

Tabla 8.2 Ventajas de tener un RWG organizacional

Permite que la organización se beneficie de la experiencia de sus proyectos y de los conocimientos de los miembros clave del personal;
Siembra la organización con personas que comparten un conjunto común de conocimientos y han llegado a un consenso sobre temas clave;
Proporciona a través de sus miembros un recurso para el resto de la organización;
Facilita el uso del conocimiento y los artefactos desarrollados para su uso en la obtención de nuevos negocios (propuestas, sesiones informativas de marketing para líderes, etc.);
Fomenta una forma común de hacer las cosas y apoya la repetibilidad y la reutilización;
Alienta y facilita la selección de métodos y herramientas apropiados, así como su despliegue e implementación;
Alienta la medición de la efectividad del proceso y los beneficios de la institucionalización;
Permite la participación en esfuerzos de vanguardia de la industria (paquetes de transición).

Northrop Grumman IT ha utilizado este mecanismo durante siete años y ha demostrado ser invaluable. Es posible que las empresas más pequeñas no tengan suficientes proyectos o AR para formar un RWG. En cualquier caso, las empresas deberían considerar trabajar con la SEI, colegios o universidades locales o consorcios de calidad para compartir las mejores prácticas de requisitos locales (por ejemplo, herramientas, técnicas, ideas) entre los miembros y servir como un foro de networking.

Inicialmente, unos pocos participantes activos (aproximadamente 6 a 8 de los 15 a 20 miembros del RWG, dependiendo de la reunión en particular) diseñaron un proceso de requisitos (RE) actualizado durante un período de aproximadamente cuatro meses. Usamos la abreviatura RE para distinguir el proceso actualizado de la versión anterior, que era un proceso RM (queríamos que el proceso actualizado abordara el ciclo de vida completo del sistema, en lugar de solo la parte del software). Posteriormente, el RWG recibió a varios proveedores durante un período de cuatro meses, quienes brindaron demostraciones de las populares herramientas de requisitos automatizados, incluidos DOORS (de Telelogic), RequisitePro ("ReqPro") de IBM (anteriormente Rational) Corporation, Calibre RM de Technology Builders (TBI) [ahora Requisitos del sistema Star Team de Starbase], CORE (de VITECH Corporation), RTM Workshop (de Integrated Chipware) y Vital Link (de Compliance Automation). Como se enfatizó anteriormente, la AR debe familiarizarse y tener experiencia en el uso de una herramienta automatizada de requisitos sólida en la industria. (Por fortaleza de la industria, me refiero a una herramienta de requisitos que proporciona las capacidades necesarias para desarrollar sistemas y software). La participación en estas demostraciones proporcionó a los miembros de nuestro RWG conocimientos adicionales sobre el uso y el valor del soporte de herramientas automatizadas para actividades relacionadas con los requisitos. Tenga en cuenta que a menudo puede descargar versiones de prueba de muchas herramientas desde los sitios web de los proveedores. Asegúrese de estar dispuesto a dedicar algo de tiempo a experimentar con la herramienta antes de descargarla; en mi experiencia, el período de evaluación de prueba pasa muy rápido, dadas sus otras responsabilidades.

Con el tiempo, nuestro RWG organizacional ha evolucionado hasta convertirse principalmente en un grupo de correo electrónico que proporciona un mecanismo para hacer preguntas, obtener respuestas, compartir información, buscar ayuda con problemas, propuestas o proyectos, encontrar RA para

nuevas asignaciones, etc. El mecanismo del GTR sigue siendo invaluable en muchos sentidos. Un informe que presenté en una conferencia de la industria sobre nuestro RWG está disponible en mi sitio web [8].

Trabajo en equipo

He tenido la suerte de ser miembro de muchos equipos empoderados y de alto rendimiento en diferentes organizaciones. Un equipo de alto rendimiento realmente es más que la suma de sus miembros. Un equipo de alto rendimiento puede lograr casi cualquier cosa que se proponga.

El trabajo en equipo evoluciona (o no) en función de una variedad de factores. Estos son algunos de los factores más importantes, según mi experiencia:

1. Existe un sentimiento de confianza. Sé que mi jefe cree que mis intenciones son buenas, incluso cuando cometo errores.
2. Los compañeros de trabajo se apoyan unos a otros. Se tratan unos a otros como clientes. Una pregunta de un compañero de trabajo no es una interrupción; más bien, es una de las razones por las que estoy allí y una oportunidad de ayudar a alguien que es muy importante para mí.
3. Se asignan al equipo objetivos significativos, realistas (alcanzables) e importantes. Los objetivos pueden ser muy ambiciosos, pero son alcanzables. Los objetivos son importantes para los objetivos comerciales de la organización y del proyecto.
4. Los miembros del equipo se dan cuenta de que cada persona tiene un papel único en el equipo y tiene habilidades especiales (considérelas como dones) que aporta al esfuerzo del equipo.
5. Los miembros del equipo se respetan unos a otros. Se tienen en alta estima. Hablan muy bien unos de otros con los demás.
6. Se lleva a cabo una reunión inicial para proporcionar un inicio oficial a los esfuerzos del equipo y para ayudar a informar a otros que el esfuerzo del equipo está en marcha.
7. El equipo desarrolla un plan de acción para sus esfuerzos, que define tareas específicas, fechas de finalización planificadas y quién tiene el liderazgo (la responsabilidad) de cada tarea.
8. El equipo rastrea e informa el progreso en comparación con el plan. Se considera a sí mismo un “mini proyecto” responsable de utilizar técnicas adecuadas de planificación y seguimiento del proyecto.
9. Los miembros del equipo se sienten apoyados por el proyecto y la organización en el desempeño de su trabajo. Las personas saben lo que está haciendo el equipo y están dispuestas a brindar apoyo (“ayudar a tirar de los remos”) cuando el equipo o un miembro del equipo se lo solicita.
10. El equipo acuerda un conjunto de reglas de conducta sobre cómo se tratarán entre sí (consulte Prácticas de requisitos efectivos [3, p. 41] para obtener un ejemplo de conjunto de reglas de conducta).

11. La dirección está interesada en el progreso del equipo. Gestión proporciona tiempo para informes periódicos de estado del equipo y quiere saber qué puede hacer para apoyar las actividades y esfuerzos del equipo. La gerencia busca activamente oportunidades para ayudar al equipo y da seguimiento a las sugerencias y compromisos que asume en reuniones de revisión y estado.
12. El equipo y sus miembros desarrollan "soluciones alternativas" cuando se encuentran obstáculos y dificultades. No dejarán barreras disuadirlos de alcanzar los objetivos establecidos (o al menos una meta realizable) subconjunto de los objetivos).
13. El equipo y sus miembros se toman el tiempo para celebrar el progreso y logros en el camino.
14. El equipo utiliza parafernalia para fomentar un mayor espíritu, motivación y compromiso, por ejemplo, tazas de café o camisetas con el equipo logo. El equipo está orgulloso de utilizarlos y proporcionárselos a las personas fuera del equipo que están ayudando.
15. Los miembros del equipo utilizan técnicas comprobadas de QI como forma de ellos hacen su trabajo (ver la discusión sobre técnicas de MC anteriormente en este capítulo).
16. Se llevan a cabo celebraciones especiales cuando el equipo logra un hito importante, por ejemplo, una recepción de final de día, un almuerzo o una cena. reconocimiento del equipo en una reunión de alta dirección y cartas de agradecimiento o elogio por parte de los gerentes o altos directivos.

Confío en que a partir de este debate usted tenga la idea de que contar con políticas efectivas El trabajo en equipo en un proyecto y en una organización es poderoso. Realmente empodera al equipo, al proyecto y a la organización. Un equipo de alto rendimiento (en mi experiencia) logrará objetivos aparentemente imposibles. Además, Es un placer ser miembro de un equipo de alto desempeño por la satisfacción y plenitud que uno obtiene.

La gerencia puede y debe preparar el escenario para permitir que sus equipos sean efectivos. Estas son algunas de las formas en que la administración hace esto:

La gerencia comunica los objetivos comerciales de la organización, articulando claramente que cada miembro de la organización es necesarios y de los que se depende para lograr estos objetivos. La gerencia expresa de manera convincente que realmente necesita el apoyo capaz de cada miembro de la organización. La gerencia solicita los mejores esfuerzos, ideas, sugerencias y energía de cada persona.

La alta dirección involucra a los gerentes subordinados en la decisión. proceso. Las creencias e inquietudes de los directivos se valoran y utilizan en tomando decisiones. Los gerentes se sienten necesarios y valorados.

La dirección brinda apoyo en forma de patrocinio, recursos, su participación personal en reuniones de inicio y revisión, tiempo para conocer el estado

informes, reconocimiento de esfuerzos e hitos, compromiso de dar seguimiento a las solicitudes, asistencia para resolver problemas y conflictos y uso de buenas habilidades para escuchar.

La gerencia mantiene su interés en las asignaciones asignadas a personas y equipos y mantiene a las personas informadas a medida que cambian las prioridades, de modo que las personas no sientan que están haciendo un trabajo innecesario o sin importancia.

La dirección se comunica de forma concertada con representantes de otras áreas, como marketing o ingeniería de sistemas, por ejemplo, enviando conjuntamente un correo electrónico a toda la organización. Esto expresa un enfoque que podría explicar o comunicar más claramente un tema de una manera que supere las agendas percibidas de los departamentos individuales.

Cuando la dirección decide no ayudar de estas y otras formas relacionadas, la probabilidad de que los equipos de proyecto de la organización tengan un alto rendimiento se reduce considerablemente, como todos sabemos por nuestra propia experiencia. Pensamos: si a mi jefe no le importa y no reconoce lo que intento hacer, ¿por qué molestarse?

Como RA, puede trabajar para fomentar el trabajo en equipo eficaz en los diversos roles descritos en el Capítulo 2. El trabajo en equipo es una de las formas en que se implementa un enfoque de calidad integrado.

Resumen

Es necesario un proceso de requisitos eficaz para tener un enfoque de calidad integrado, y se requiere un enfoque de calidad integrado para que el proceso de requisitos (o cualquier otro) funcione mejor. La implementación de un enfoque de calidad integrado implica:

Familiarizar a todos en la organización con la preocupación de la gerencia por la calidad y el valor y la importancia de la mejora de la calidad para la razón de ser de la organización;

Capacitar personas para liderar equipos de MC;

Capacitar a las personas sobre cómo diseñar procesos de trabajo;

Gestionar utilizando un enfoque de mejora de la calidad.

Las AR se encuentran en una posición estratégica para ayudar a proyectos y organizaciones a implementar y utilizar un enfoque de calidad integrado (revise las funciones de la AR sugeridas en el Capítulo 2 en este contexto). Al hacerlo, ayudan a garantizar un proceso de requisitos eficaz.

Estudio de caso: un ejemplo de mejora de la calidad Desviado

Se pidió a una ingeniera de procesos que ayudara a una división canadiense de una gran empresa multinacional a aprovechar su amplia experiencia y conocimientos.

en ingeniería de procesos y gestión de proyectos. La empresa multinacional se había comprometido recientemente a utilizar un marco de ingeniería de procesos. El consultor en ingeniería de procesos participó en una sesión informativa presentada por un vicepresidente de la empresa matriz al personal técnico de la división, compuesto por aproximadamente 40 personas. La sesión informativa comunicó claramente la importancia crítica de la mejora de procesos, demostró el patrocinio y el compromiso de la alta dirección y describió el marco probado de mejora de procesos que se utilizaría. La audiencia fue extremadamente receptiva y pareció darse cuenta de que se les ofrecía algo nuevo y diferente para ayudarlos a ser más productivos y efectivos. El consultor notó miradas de emoción en los rostros del personal técnico. Quedó impresionada por el grado de atención brindada al presentador y por la cantidad y consideración de las preguntas y comentarios posteriores. El escenario parecía estar preparado para una iniciativa de mejora de procesos productiva y eficaz.

Sin embargo, no se lograron avances significativos en la mejora de procesos durante los siguientes tres años. ¿Por qué no? Un análisis de este estudio de caso indicó los siguientes problemas:

Aunque la alta dirección de la empresa matriz estaba convencida de que la mejora de procesos ayudaría a la división a alcanzar sus objetivos comerciales, la dirección de la división no estaba tan convencida. No hubo participación de la gerencia de la división en la decisión de la empresa matriz de iniciar actividades de mejora de procesos y no hubo patrocinio o compromiso para la implementación. La iniciativa de mejora de procesos fue, más o menos, alentada desde arriba hacia abajo sin suficiente tiempo o preparación para la aceptación en los niveles inferiores. Había una falta de trabajo en equipo que impedía que las personas de todos los niveles de la organización se concentraran en el objetivo común.

La dirección de la división utilizó tecnicismos para retrasar la mejora del proceso. Por ejemplo, una excusa para el retraso fue que "el plan de mejora de procesos de la empresa matriz no se había finalizado". En otras palabras, el espíritu de mejora de procesos, que podría haber seguido adelante de muchas maneras con o sin un plan final de mejora de procesos "aprobado" a nivel de empresa, fue fácilmente desviado por el cumplimiento de la letra burocrática.

La empresa matriz gastó una gran cantidad de dinero para desarrollar un plan de mejora de procesos (PIP) y contratar consultores experimentados para que prestaran su experiencia y conocimientos. Se brindó capacitación a la división para familiarizar a los empleados con el plan y con la mejora de procesos en general. Sin embargo, estas inversiones no dieron sus frutos. No se realizaron cambios significativos en la organización y no se produjeron mejoras importantes en la productividad o la calidad.

La mejora de procesos y la mejora de la calidad son tanto cuestiones de cambio cultural como cuestiones de cambio técnico o de gestión. Un mejor

El enfoque podría haber sido que la empresa matriz emitiera un conjunto de planes de mejora de procesos y principios generales por etapas. Las primeras versiones podrían haber fomentado actividades de mejora de procesos simples y efectivas, con planes más detallados publicados más adelante. Entonces, la empresa podría haber monitoreado estos primeros pasos y alentado a los niveles inferiores de la división a adoptar la solución como su idea, en lugar de algo que fue impuesto desde arriba. La empresa también podría haber utilizado la retroalimentación generada al monitorear de cerca los resultados de estos pasos iniciales para guiar el desarrollo posterior e identificar barreras técnicas o culturales específicas que deben superarse dentro de la división.

El diseño de su proceso y su enfoque de mejora se volverán más detallados a medida que madure su experiencia con él. ¡Genial! Lo importante es que los procesos esenciales del proyecto u organización sean diseñados, documentados y mejorados continuamente por sus partes interesadas. Al tener un proceso organizacional estándar para las actividades necesarias, como requisitos, planificación de proyectos, seguimiento de proyectos, revisiones por pares, CM, QA, DP, gestión de cambios tecnológicos y otros procesos necesarios, los proyectos pueden reutilizarlo y adaptarlo según sea necesario. Tener los procesos (diagramas de flujo, PD narrativos y artefactos relacionados, como plantillas para planes) disponibles en una biblioteca electrónica automatizada facilita lograr la repetibilidad en una organización. El uso del control de versiones facilita la mejora continua de los artefactos.

Referencias

- [1] Six Sigma Qualtec, Historia de QI: Herramientas y técnicas, una guía para la resolución de problemas, 3^a ed., 1999. Llame al (480) 586-2600 para obtener información. Véase también www.ssqi.com/homepage.asp.
- [2] Walton, M., The Deming Management Method, Nueva York: The Putnam Publishing Group, 1986. El ciclo PDCA a menudo se atribuye a Deming porque lo introdujo en Japón. Walter A. Shewhart lo concibió originalmente. Véanse las págs. 86–88.
- [3] Young, RR, Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.
- [4] Clark, BK, "Efectos de la madurez del proceso en el esfuerzo de desarrollo", Centro de Ingeniería de Software, Universidad del Sur de California, 1999, en www.ralphyoung.net.
- [5] Northrop Grumman IT DES, "The Road to CMM(I) Level 3", informe técnico, disponible en ralph.young@ngc.com.
- [6] Wiegers, KE, Creación de una cultura de ingeniería de software, Nueva York: Dorset House Publishing, 1996.
- [7] Young, RR, "La importancia y el valor de la mejora de procesos", en www.ralphyoung.net.
- [8] Young, RR, "El valor de un grupo de trabajo organizacional", en www.ralphyoung.net.
- [9] Materiales de GOAL/QPC, en www.goalqpc.com.
- [10] Comunicado de prensa de Northrop Grumman IT DES sobre CMMI Nivel 5, en www.irconnect.com/noc/pages/news_releases.mhtml?d=35405.

Machine Translated by Google

CAPÍTULO

9

Contenido

¿Cómo debemos apoyar a los PM?

¿Cómo debemos apoyar a los clientes?

¿Cómo deberíamos apoyar a los desarrolladores?

Resumen

Caso de estudio

Referencias

Una visión de los requisitos Ingeniería

Una visión razonable para la ingeniería de requisitos es lograr los siguientes objetivos:

1. Los clientes reconocen que el sistema cumple con sus Expectativas.
2. Los ingenieros de sistemas y software se sienten satisfechos con su trabajo.
3. Se reconoce que los AR capacitados y experimentados han marcado una diferencia positiva.
4. Los requisitos de las tareas y proyectos están completos y controlados para proporcionar una base estable para el desarrollo.
5. Los proyectos de desarrollo de sistemas y software incurren en menos del 15% de retrabajo no planificado y menos del 10% de desperdicio. recursos.
6. Se desarrolla un mercado informado, donde se valora la capacidad de un proveedor potencial para definir, implementar y respaldar el desarrollo basado en requisitos.
7. Quienes dominan los requisitos, la ingeniería de sistemas y las tareas de gestión de proyectos asumen la responsabilidad personal de orientar a otros.

Estos objetivos son alcanzables. De hecho, existen focos de excelencia donde las tareas, los proyectos e incluso las organizaciones ya están logrando estos objetivos. Hay organizaciones, proyectos y tareas que se desempeñan con altos niveles de madurez de procesos, según lo medido por estándares de la industria como CMM [1], CMMITM [2] y Six Sigma [3]. El SEI proporciona una página web (www.sei.cmu.edu/sema) que informa sobre el progreso logrado por las organizaciones que utilizan CMM y CMMI. Sin embargo, el promedio de la industria es mucho más bajo que el de las organizaciones de alta madurez que sigue el SEI, según lo informado por

organizaciones como el Grupo Gartner y el Grupo Standish Internacional.

Está claro que las organizaciones tienen la capacidad de desempeñarse a niveles más altos de logro, medidos por muchas de las metas proporcionadas anteriormente. Por qué ¿no es así? Evidentemente, la respuesta a esta pregunta es compleja. Del Dr.

Desde la perspectiva de Deming [4], es una cuestión de gestión: la gestión a menudo no no empodera a su fuerza laboral ni permite un desempeño bueno o excelente en sus organizaciones. Sin duda, se puede ayudar a la gestión. Cada persona puede hacer un diferencia.

Este libro sugiere que los ingenieros y analistas de requisitos pueden ayudar mucho. Los requisitos son la base de todo el trabajo que se realiza en ingeniería de sistemas y software. Los resultados de la industria [5, 6] sugieren que existe

Hay mucho margen de mejora en la práctica de la ingeniería de requisitos. Está claro que hay suficiente información disponible para permitir una mejora significativa en el logro de los objetivos que he enumerado. La necesidad obvia

Es que los profesionales utilicen prácticas de requisitos más efectivas. Esto requiere conocimientos y experiencia. Creo que se puede fortalecer y mejorar aún más. sus habilidades y ser aún más efectivo en los roles que le brinda.

Como indiqué en el Capítulo 6, esto requiere mucho trabajo duro. Como hablo en talleres y seminarios [7], las cosas mejorarán sólo si los practicantes hacemos algunas cosas de manera diferente. No es que no sepamos (o no podamos averiguar) qué hacer; es que no perseveramos en hacer las cosas tan bien como podemos hacerlos. En mis seminarios, animo a los participantes a crear su propia lista de compromiso personal. Los desafío a escuchar atentamente y Identifique al menos tres cosas que estén dispuestos a comprometerse a hacer de manera diferente en sus situaciones laborales. Obviamente, la palabra clave aquí es "comprometerse". I desafíos a trabajar con sus pares y gerentes para cambiar algunos cosas que marcarán la diferencia. Si ha asimilado este libro, tengo la esperanza y la seguridad de que habrá notado algunas cosas que puede hacer de manera diferente en su propio entorno de trabajo. Aprovecha para seguir hasta el final—para hacer el trabajo duro que mejorará los resultados de sus tareas, proyecto y organización.

La Tabla 9.1 proporciona algunas oportunidades para que los RA y los ingenieros mejoren tasas de éxito del proyecto. Piense en ellos como facilitadores de la excelencia que buscamos. Más fundamentalmente, la experiencia confirma que son condiciones necesarias para alcanzar la excelencia en la ingeniería de requisitos.

La ingeniería de requisitos es difícil. No se trata simplemente de escribir lo que el cliente dice que quiere. Clientes y

Los usuarios no han pensado en sus necesidades reales y no son capaces de articular a ellos. Otro problema fundamental en los negocios es que los requisitos son inherentemente dinámicos; cambiarán con el tiempo a medida que cambie nuestra comprensión del problema que intentamos resolver. La importancia de buenos requisitos y la naturaleza dinámica subyacente del proceso significa que debemos ser lo más precisos posible y, al mismo tiempo, flexibles. Flexible no significa "débil", sino más bien que tenemos un proceso para adaptar los requisitos modificados a medida que aclaramos los requisitos reales de clientes.

Tabla 9.1 Algunas oportunidades para que los RA e ingenieros mejoren las tasas de éxito de los proyectos

<p>Reconocer que un ambiente de trabajo productivo significa apoyar a las personas, lograr un trabajo en equipo efectivo y establecer un valor de mejora continua. Tome medidas para crear un entorno de trabajo más productivo para que el trabajo relacionado con los requisitos sea eficaz.</p>
<p>Los proyectos (de desarrollo) deben gestionarse (se necesita una supervisión continua para garantizar que se estén haciendo las cosas correctas, adecuada y correctamente). Debemos gestionar mejor los proyectos y, al hacerlo, reducir los defectos relacionados con los requisitos.</p>
<p>Formar y utilizar especialistas en ingeniería de requisitos.</p>
<p>Tener y utilizar un proceso de requisitos que aborde las actividades relacionadas con los requisitos del ciclo de vida completo. Invertir del 8% al 14% del costo total del esfuerzo de desarrollo del proyecto en actividades relacionadas con los requisitos durante todo el ciclo de vida del proyecto.</p>
<p>Invierta más tiempo en identificar los requisitos reales.</p>
<p>Utilice prácticas de requisitos efectivas.</p>
<p>Utilice una herramienta de requisitos automatizada para respaldar el esfuerzo de desarrollo.</p>
<p>Proporcionar un proceso y mecanismo eficaz para gestionar los cambios en los requisitos.</p>
<p>Actúe cuando las cosas no funcionen. Sabemos cuando las cosas no funcionan.</p>
<p>Asegúrese de que los esfuerzos relacionados con los requisitos sean efectivos.</p>
<p>Proporcione modelos a seguir que demuestren consistentemente hábitos y disciplinas de trabajo efectivos, por ejemplo, creando y utilizando un equipo conjunto para ser responsable de los requisitos.</p>

Este no es un trabajo para el miembro más joven del equipo ni para el miembro menos talentoso del grupo. Requiere la combinación de excelentes habilidades técnicas con conocimiento del dominio, buena gente y habilidades de comunicación.

Los siguientes son algunos de los desafíos que enfrentamos para mejorar los requisitos. ingeniería de elementos:

PM que se concentran en las actividades diarias y son incapaces de abordar adecuadamente las necesidades humanas subyacentes y los problemas a largo plazo. Un estudio de la industria realizado por Standish Group concluyó que el 15% de los proyectos fracasaron rotundamente en 2002, y otro 51% se consideraron "cuestionados", es decir, retrasados, por encima del presupuesto o completados con una funcionalidad reducida [5]. Sólo el 54% de las características originalmente definidas de un proyecto se entregan y, de las que se entregan, ¡el 45% nunca se utiliza! El Grupo Standish cree que el problema tiene su origen en unos requisitos mal definidos. Importantes causas fundamentales de los fracasos de los proyectos son atribuibles a la planificación y el seguimiento de los mismos. Algunos PM solo hablan de labios para afuera sobre la calidad, el trabajo en equipo y la mejora continua y no están dispuestos a invertir en capacitación y práctica para crear una "cultura de mejora de la calidad".

1. Una buena referencia es Karl E. Wiegers, Creando una cultura de ingeniería de software (Nueva York: Dorset House Publishing, 1996). Véase también Rita Hadden, Liderando el cambio cultural en su organización de software (Viena, VA: Management Concepts, 2003).

Otros PM no se centran en satisfacer los requisitos del proyecto porque se centran en ser receptivos a sus tres jefes: sus gerentes (que realizan revisiones de desempeño y esperan que los PM alcancen objetivos de ingresos y ganancias); sus clientes (que se preocupan por las expectativas de los usuarios); y el personal de su proyecto (que busca dirección). El PM debe equilibrar todas estas necesidades y garantizar que todas las partes participen en el esfuerzo del proyecto. La figura 9.1 proporciona algunas ideas sobre el complejo papel del PM. El PM debe mantener un enfoque en el cliente, responder a la dirección ejecutiva y proporcionar un entorno eficaz para los desarrolladores. Un buen PM puede tener que depender de un gerente subordinado, como el ingeniero de sistemas, para que sea responsable de los requisitos.

Desarrolladores que se niegan a utilizar técnicas mejoradas, incluso cuando se ha demostrado que estas técnicas producen mejores resultados; que no hayan aprendido técnicas de ingeniería de requisitos; y a quienes no se les han proporcionado expectativas sobre cómo deben trabajar y qué deben hacer en diferentes situaciones.

Clientes que creen que comprenden sus necesidades (pero dejan que sus egos los obliguen a adoptar una posición inamovible), están impacientes por obtener resultados a corto plazo, requieren el uso de técnicas arriesgadas y tienen experiencia limitada trabajando en modo de asociación² o de equipo . eso requiere un compromiso común para el éxito del proyecto.

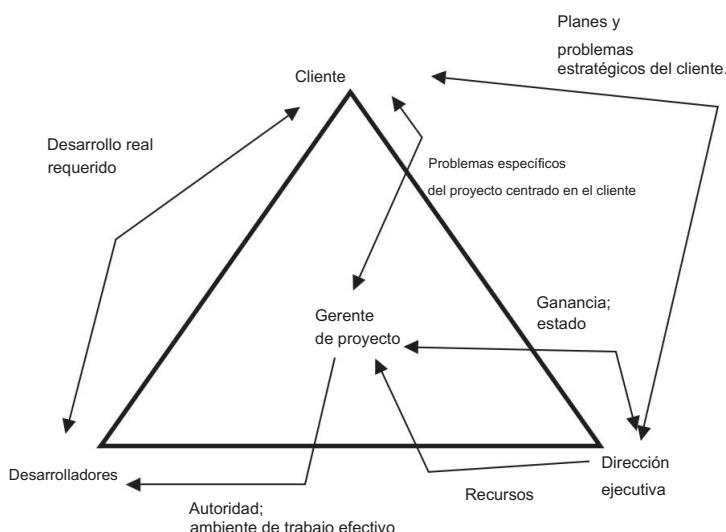


Figura 9.1 Los desafíos del PM. (Fuente: Richard Raphael. Usado con autorización).

2. Véase Frank Carr et al., *Partnering in Construction: A Practical Guide to Project Success* (Chicago: American Bar Association, 1999) para un tratamiento exhaustivo del proceso de asociación. Visite www.facilitationcenter.com para obtener una referencia de un profesional con experiencia en implementar el proceso de asociación de manera efectiva.

Todos estos desafíos obstaculizan la excelencia como estándar de la vida diaria, el trabajo en equipo como enfoque hacia la excelencia y la mejora continua como hábito.

Nosotros (todos los que participamos en la realización de actividades relacionadas con requisitos en sistemas y proyectos de desarrollo de software) tenemos una serie de desafíos que debemos abordar. Debemos decidir si estamos comprometidos o no a mejorar el desempeño y trabajar para lograr un mayor logro de las metas establecidas. A continuación se muestran algunos ejemplos de cosas que debemos hacer en nuestro trabajo diario para vivir nuestro compromiso.

¿Cómo debemos apoyar a los PM?

Deberíamos ayudar a los PM a beneficiarse de las experiencias de otros proyectos. Por ejemplo, en mi experiencia, la mayoría de los PM se beneficiarían si prestaran más atención a las dimensiones humanas de los proyectos. Los PM podrían preguntar a los artistas cómo van las cosas y luego reflexionar e incorporar aportaciones siempre que sea posible. El punto es que existe una gran cantidad de información y consejos disponibles por parte del personal de cualquier proyecto que pueden fortalecer y mejorar aún más las prácticas que se utilizan en el proyecto. Las personas dan sus mejores esfuerzos cuando se sienten realizadas y empoderadas³ en su entorno laboral. Los PM deben descubrir qué quieren hacer los miembros del equipo, fomentar ideas de mejora y buscar consejos sobre cómo hacer que su ambiente de trabajo sea más positivo. Deben crear una atmósfera de confianza y verdadero interés en las opiniones de los miembros del equipo y no ponerse a la defensiva ante comentarios negativos o inesperados. Luego, los PM deben tomar medidas para garantizar que se esté realizando el trabajo necesario y que la gente salga del trabajo la mayoría de los días sintiendo que han logrado cosas.

Otra cosa que debemos hacer por los PM es brindarles recursos sobre la experiencia de la industria en materia de ingeniería de requisitos y hacerles recomendaciones específicas sobre los enfoques que podrían considerar para un proyecto específico. Por ejemplo, recuerde los datos de la NASA anotados en el Capítulo 1 que dan un mensaje claro y poderoso: Los proyectos que gastaron el promedio de la industria del 2% al 3% del costo o esfuerzo total de desarrollo del proyecto en el proceso de requisitos (ciclo de vida completo) experimentaron un Entre el 80% y el 200% de sobrecosto, mientras que los proyectos que invirtieron entre el 8% y el 14% del costo o esfuerzo total del proyecto en el proceso de requisitos tuvieron un sobrecosto del 0% al 60% [8]. Además, debemos aclarar a los PM que una parte del tercio de los co

3. Por empoderado me refiero a que los empleados se sientan responsables no sólo de hacer un buen trabajo, sino también de hacer que toda la organización funcione mejor. Los equipos trabajan juntos para mejorar su desempeño continuamente, logrando mayores niveles de productividad. Las organizaciones están estructuradas de tal manera que las personas sienten que son capaces de lograr los resultados que desean, que pueden hacer lo que hay que hacer, no sólo lo que se les exige. Véase Empowerment: A Practical Guide for Success, de Cynthia Scott y Dennis Jaffe (Menlo Park, CA: Crisp Publications, 1991).
4. Como se documenta en Prácticas efectivas de requisitos (p. 79), el 80% de todos los defectos del producto se insertan en las actividades de definición de requisitos. Los costos de retrabajo se estiman en un 45% de los costos totales. Al tomar el 80% del 45%, aprendemos que el 36% (más de un tercio) de los costos totales del proyecto (según datos de la industria) pueden potencialmente evitarse eliminando los errores de requisitos de los productos de trabajo.

pueden redirigirse para pagar las mejoras necesarias en los procesos y prácticas. Necesitamos mostrarles cómo realizar un seguimiento del retorno de la inversión (ROI) de mejoras específicas, para que tengan los datos necesarios para tomar buenas decisiones (“administrar con base en hechos”).

Los proyectos deben priorizar los requisitos. Como hemos comentado, no todos los requisitos son iguales: algunos son más importantes que otros. Davis proporciona tres estudios de casos y ofrece 14 recomendaciones en su artículo “El arte de la clasificación de requisitos” [9].

Los proyectos necesitan un proceso y mecanismo eficaz para gestionar los cambios en los requisitos. Sabemos por experiencia que aquí es donde perdemos el control de muchos esfuerzos técnicos. Tomemos medidas para que esto no vuelva a suceder. Por ejemplo, será útil utilizar un equipo conjunto de cliente/proveedor para mantener la responsabilidad de los requisitos durante todo el esfuerzo de desarrollo y hacer que todas las solicitudes de cambio pasen a través de un solo canal. Otro mecanismo es controlar las fuentes de requisitos no oficiales [10]. Documentar la justificación de cada requisito también reducirá el número de requisitos (hasta en un 50%, según la experta de la industria Ivy Hooks) [8].

¿Cómo debemos apoyar a los clientes?

Deberíamos familiarizar a los clientes (y a los PM y al equipo del proyecto) con la experiencia de la industria de que el conjunto inicial de requisitos escritos rara vez (quizás nunca) es el conjunto de requisitos reales. Como se explicó en el Capítulo 1, los clientes necesitan la ayuda de RA capacitados para identificar los requisitos reales. La inversión realizada para descubrir y desarrollar los requisitos reales se verá recompensada con creces al evitar el trabajo de desarrollo posterior sobre un conjunto inadecuado de requisitos. ¡No sigamos reaprendiendo esta lección en cada proyecto futuro!

¿Estamos realmente aprovechando el poder del trabajo en equipo eficaz? Debemos establecer un compromiso mutuo para permitir que cualquier esfuerzo tenga éxito. Deberíamos encontrar formas creativas de hacer posible el trabajo en equipo, incluso en entornos que no lo apoyan. Por ejemplo, un enfoque de equipo de producto integrado (IPT) [11] en el que el cliente es miembro del IPT facilita la comunicación y el esfuerzo del equipo.⁵ Deberíamos asesorar a los clientes, PM y organizaciones sobre enfoques exitosos que hayan logrado un trabajo en equipo efectivo. y fomentar el compromiso que se requiere para que cualquier proyecto tenga éxito [12].

¿Cómo deberíamos apoyar a los desarrolladores?

Una forma en que deberíamos ayudar a los desarrolladores es escucharlos y tomar medidas sobre lo que dicen, para hacerles saber que estamos ahí para ayudarlos. por esto yo

5. El Departamento de Defensa ha comenzado a utilizar el término gestión integrada de equipos en lugar de IPT debido a la preocupación de que los IPT sean no funciona tan bien como se necesita.

significa que es trabajo de la gerencia proporcionar un ambiente de trabajo que maximice la efectividad. La conclusión es que el éxito depende de las personas. Nosotros debe proporcionar un entorno en el que las personas puedan ser eficaces y sentirse realizadas, donde puedan crecer y donde sean apreciadas y valoradas.

La gerencia debe preocuparse profundamente y demostrarlo.

Otras formas en las que podemos ayudar a los desarrolladores son poner en práctica procesos y prácticas vigentes (y esperar que se apliquen; consulte la siguiente párrafo), trabajar para reducir el retrabajo, crear un entorno de mejora continua y trabajar para lograr una calidad cada vez mejor en los productos de trabajo. Si no hacemos esto, corremos el riesgo de que la gente se sienta frustrada y abandone el organización y proyectos.

Otra cosa importante que debemos hacer es dejar claro a los desarrolladores que esperamos que utilicen políticas, procesos y políticas mejorados y probados. mecanismos, métodos, técnicas y herramientas que son los estándares en el organización y en el proyecto. “¿Por qué no practican?” de Watts Humphrey Lo que predicamos”⁶ detalla algunas razones por las que los desarrolladores no utilizan técnicas mejoradas, incluso cuando se les proporciona evidencia de que logran mejores resultados. resultados. Debemos dejar claro que no es aceptable que la gente retroceda sobre sus viejas formas de hacer las cosas. Deberíamos proporcionar modelos a seguir que demuestren consistentemente hábitos y disciplinas de trabajo efectivos.⁷ Einstein comentó que la única manera racional de educar es con el ejemplo.

Quienquiera que esté en el proyecto abordando los problemas de requisitos necesita apoyo. Además de una herramienta de requisitos automatizada eficaz, los ingenieros de requisitos necesitan formación en ingeniería de requisitos. ¿Qué es un buen requisito? ¿Por qué los RA no deben tomar decisiones sobre requisitos? ¿Por qué no debería placa de oro (agregar características y capacidades que no son necesarias)? Deberíamos crear un conjunto de expectativas para el personal del proyecto sobre cómo deben trabajo y lo que deben hacer en diferentes situaciones. La mejor manera de lograr esto es mediante una capacitación adaptada al entorno de un proyecto en particular y necesidades y presentadas por personas que realmente pueden ayudar. La formación y el asesoramiento al personal debe presentarse de una manera que respete a las personas: todos tenemos egos, y si el mío está herido, me resulta difícil esforzarme al máximo.

Resumen

Hay muchas cosas que podemos hacer para crear un camino para abordar los problemas relacionados con los requisitos. No pretendo que esto sea fácil o que pueda serlo. logrado rápidamente. Sin embargo, lograr la visión definida para la ingeniería de requisitos requiere que hagamos las cosas de manera diferente. espero que tú se comprometerá a realizar algunos cambios que le ayudarán.

6. Disponible en www.sei.cmu.edu/publications/articles/practice-preach/practice-preach.html.

7. Despues de la fiebre del oro: creación de una verdadera profesión de ingeniería de software, de Steve McConnell (Redmond, WA: Microsoft Press, 1999) está lleno de ideas y sugerencias para ayudar con esta situación. McConnell señala en su epílogo que Los problemas comunes de desarrollo no se podrán evitar sin nuestro apoyo.

Caso de estudio

Existe una percepción errónea común de que los sitios web son simples, se pueden construir rápidamente y no requieren la planificación y gestión de recursos humanos que requieren los sistemas reales o las aplicaciones de software. Esta es una trampa para los propietarios y desarrolladores de sistemas poco sofisticados, y todavía hay muchos de ellos por ahí.

Se llamó a un consultor de ingeniería de requisitos para investigar por qué un sitio web para una agencia gubernamental no se estaba completando a tiempo.

El trabajo se estaba realizando a través de un titular de horario de la Administración de Servicios Generales (GSA), pero había sido subcontratado a otra empresa que había comercializado el negocio. El contratista principal no había revisado detenidamente la propuesta ni el trabajo inicial porque no tenía experiencia interna en desarrollo de aplicaciones web. El trabajo tenía un precio fijo, las relaciones con los usuarios ya eran tensas y el gobierno había emitido una "causa de demostración" y amenazaba con congelar el cronograma GSA del contratista.

Al revisar los materiales contractuales, el consultor descubrió que la solución era simplemente colocar una base de datos de Microsoft Access existente en la Web, probarla y escribir una guía de usuario y documentación del sistema. Originalmente había una tarea para hacer un análisis de la base de datos, pero todas las partes acordaron una modificación para reducir el costo del proyecto eliminando esta tarea. No era necesario desarrollar un documento de requisitos. Quizás los usuarios de la agencia pensaron que eso habría supuesto un coste y un retraso innecesarios.

Después de todo, tenían un sistema de trabajo que representaba sus necesidades. ¿O lo hicieron?

Cuando el consultor revisó la aplicación (porque se trataba de una aplicación, no simplemente de un sitio web estático), descubrió que la base de datos efectivamente se había colocado en la web. Se había escrito una interfaz para aceptar criterios de selección y producir informes en línea de los datos. Si las selecciones recuperaran grandes cantidades de datos, después de una espera, se generaría una tabla grande en la pantalla que obligaría al usuario a desplazarse hacia la derecha y hacia abajo para verlo todo, a pesar del uso de una fuente pequeña. Si las selecciones fueran demasiado grandes, la recuperación expiraría.

Una selección de menú diferente solicitó una contraseña y luego proporcionó una una larga serie de formularios de entrada de datos para permitir al usuario introducir o actualizar los datos.

Los desarrolladores consideraron que el proyecto estaba esencialmente terminado, y el único trabajo restante era (1) crear una forma de imprimir los informes y (2) corregir los tiempos de espera y otros errores que el usuario señaló durante las pruebas.

La base de datos existente estaba en la Web y proporcionaba una forma de ingresar datos y una capacidad flexible de generación de informes que podía generar cualquier informe que el usuario quisiera.

Desafortunadamente, lo que el usuario quería era una forma de automatizar el proceso previamente manual de recopilación de datos de sus oficinas en el campo. El proceso actual consistía en que las oficinas llenaran formularios o enviaran hojas de cálculo a la sede, y otro contratista introduciría los datos en la base de datos de Access. Los desarrolladores del nuevo sistema no se habían dado cuenta de que el propietario del sistema había asumido que poner la base de datos en la Web les daría la posibilidad de que las oficinas de campo ingresaran directamente los datos a través del

Web. Esto requeriría un sistema más sofisticado que una única contraseña de usuario y un conjunto de pantallas de entrada de datos. De hecho, resultó que el proceso existente incorporaba flujo de trabajo y aprobación antes de que los datos aparecieran en la sede, y la sede podía tener preguntas o incluso devolverlos para reelaborarlos antes de que pasaran a formar parte de la base de datos “oficial” disponible para el público.

Para colmo, muchos de los “errores” de los que se quejaban los usuarios estaban relacionados con problemas de datos. La década de datos existentes estuvo plagada de datos inconsistentes, faltantes y erróneos. El usuario ejecutaría informes en el sistema existente para compararlos con el sistema basado en Web, pero cualquier cambio menor en la consulta podría arrojar resultados diferentes. Resultó que la base de datos de Access se había utilizado tal como existía, sin cambios, y que alguien sin experiencia en bases de datos la había desarrollado. No se había realizado ningún diseño de base de datos real. Existían numerosas tablas de datos similares. Los elementos de datos tenían nombres similares. Para las claves se utilizaron campos de texto con ortografía inconsistente. Los datos fueron realmente una pesadilla. Sin embargo, los informes generados a partir de él se han presentado al Congreso cada año, y los usuarios deberían poder duplicarlos cuando consultan el sitio Web.

Después de un largo proceso de reuniones y negociaciones con el gobierno, el contratista principal acordó construir los aspectos de control de acceso y flujo de trabajo del sistema y gestionar el control de calidad y las pruebas, mientras que el desarrollador continuó mejorando los aspectos de recuperación e ingreso de datos del sistema. Se aplicaron recursos de programación adicionales. Se implementó el control del código fuente. El sistema fue codificado, probado, documentado y finalmente aceptado. El gobierno apreció la inversión y el esfuerzo adicionales realizados por el contratista principal, que no fue despedido. Se restableció el horario de GSA.

El desarrollador afirmó que los errores de datos no eran su responsabilidad contractual, y el gobierno aceptó a regañadientes, pero el contratista principal tuvo que realizar una cierta limpieza de datos para que el sistema fuera aceptado de todos modos. Realmente era necesario limpiar los datos y diseñar la base de datos, pero no se hizo porque esos fondos se habían eliminado del contrato.

Los cambios en la base de datos en ese momento también habrían requerido cambios de codificación, y el contratista principal decidió no hacerlo, aunque habría resultado en un sistema mucho más simple y más fácil de mantener.

Una atención temprana a las necesidades podría haber evitado este desastre. La divergencia entre lo que quería el gobierno (pero nunca se articuló claramente) y lo que el contratista sentía que era responsable era tan grande que la única manera de recuperarse era reelaborar el problema con un sobrecoste de casi el 100%.

El contratista principal, el desarrollador y el cliente gubernamental cometieron los siguientes errores relacionados con los requisitos:

Suponiendo que un sistema existente es una buena representación de los requisitos mentos para su reemplazo;

Asumir que “ponerlo en la Web” no lo cambia ni requiere el desarrollo de ningún nuevo proceso o requisito comercial;

Aceptar omitir el análisis de datos, asumiendo que los datos heredados estaban limpios y bien estructurado;

Olvidarse de la seguridad, el rendimiento, la mantenibilidad y otros tipos de requisitos no funcionales.

La principal lección que se puede aprender de este caso es que los propietarios de sistemas básicamente exigen que un sistema satisfaga la necesidad empresarial prevista, no que cumpla al pie de la letra los requisitos contractuales. Diseñar para la Web no es diferente. Si el sistema no satisface las necesidades del negocio, será un fracaso.

Si los desarrolladores no comprenden completamente la necesidad empresarial, no pueden inferirla a partir de un sistema heredado, porque los procesos empresariales que rodean al sistema heredado no están documentados en el sistema. En este caso, podrían haber adivinado que existía un requisito de flujo de trabajo y aprobación multinivel de los datos presentados, pero tal vez no hubieran adivinado que existía un requisito de replicar ciertos resultados de informes existentes para evitar contradecir presentaciones anteriores del Congreso.

Una lección secundaria es que siempre se debe asumir que los datos heredados están "sucios" al menos hasta cierto punto. Es extremadamente arriesgado suponer que las bases de datos existentes se pueden reutilizar sin rediseñarlas. En este caso, omitir el análisis de datos fue muy costoso para todas las partes y provocó la eventual falla del sistema.

Aunque a algunas empresas y oficinas gubernamentales se les ha hecho creer que se pueden crear prototipos y construir sitios web en "tiempo de Internet", todavía es necesario pensar detenidamente los requisitos. Debido a que este trabajo fue contratado, y particularmente porque tenía un precio fijo, los requisitos deberían haberse elaborado y documentado de manera más explícita. Una forma de evitar este tipo de sorpresas es dividir los requisitos y los análisis de datos en una tarea con un precio separado, cuyos resultados sirvan de base para el diseño y desarrollo posteriores.

Nombre retenido por solicitud
Consultor de ingeniería de requisitos

Referencias

- [1] Paulk, MC, et al., Capability Maturity Model for Software, Versión 1.1, febrero de 1993. SEI, Carnegie-Mellon University, Pittsburgh, PA, 1993, en www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html.
- [2] Equipo de producto CMMI, Integración del modelo de madurez de capacidad, versión 1.1, diciembre de 2001. SEI, Universidad Carnegie-Mellon, Pittsburgh, PA, 1993, en www.sei.cmu.edu/capability/cmmi.
- [3] Eckes, G., Cómo hacer que Six Sigma dure: gestionar el equilibrio entre el cambio cultural y técnico, Nueva York: John Wiley & Sons, Inc., 2001.
- [4] Deming, WE, Fuera de la crisis, Boston: Centro de Ingeniería Avanzada del MIT Estudio, 1986.
- [5] The Standish Group, CHAOS Chronicle 2003 Report, West Yarmouth, MA: The Standish Group International, 2002, en www.standishgroup.com.

- [6] McGibbon, T., "Un caso de negocio para la mejora de procesos de software revisado", Roma, Nueva York: Centro de análisis y datos para software, 30 de septiembre de 1999, en www.dacs.dtic.mil/techs/roispi2.
- [7] Consulte "Tutoriales, conferencias, presentaciones, talleres y debates" en www.ralphyoung.net.
- [8] Hooks, IF y KA Ferry, Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos, Nueva York: AMACOM, 2001.
- [9] Davis, AM, "El arte de la clasificación de requisitos", IEEE Computer (marzo de 2003): 42–49.
- [10] Weinberg, gerente general, "¡Simplemente diga no! Mejorando el proceso de requisitos", American Programador 8(10) (octubre de 1995): 19–23.
- [11] Equipo de productos integrados del Cuerpo de Ingenieros del Ejército de EE. UU., página web, en www.usace.army.mil/ci/lcmis/lcmipt.html.
- [12] Una buena referencia sobre el trabajo en equipo es The Team Handbook de PR Scholtes et al. , 2^a ed. (Madison, Wisconsin: Oriel Inc., 2001). La tesis de los autores es que para tener éxito en el entorno actual, se deben reunir los conocimientos, las habilidades, la experiencia y las perspectivas de una amplia gama de personas.

Machine Translated by Google

CAPÍTULO

10

Contenido

A dónde ir desde aquí

Avanzando

Un mandala de requisitos

Resumen

Caso de estudio

Referencias

Avanzando: Conocible Requisitos, manejables Riesgo

En este libro hemos abordado temas que son de gran importancia para la RA:

Los requisitos son importantes y aprovechar las actividades relacionadas con los requisitos en un proyecto tiene gran poder y efecto (Capítulo 1).

Se necesitan prácticas efectivas y procesos documentados y bien entendidos en proyectos de todos los tamaños, no sólo para el desarrollo a gran escala (Capítulo 1).

La experiencia de la industria demuestra el valor de invertir del 8% al 14% de los costos totales del proyecto en el proceso de requisitos del ciclo de vida del sistema; Los datos confirman que este nivel de inversión produce los mejores resultados (Capítulo 1).

Se deben desarrollar criterios para un buen requisito y cada requisito debe evaluarse con respecto a esta lista para aclarar y reformular el requisito (Capítulo 1).

La documentación de la justificación de cada requisito (por qué es necesario) puede eliminar hasta la mitad de los requisitos establecidos (y muchos costos) (Capítulo 1).

Diferenciar entre los requisitos declarados y los requisitos reales , y trabajar en colaboración con sus clientes y usuarios para identificar los requisitos reales (Capítulo 1).

Planificar el esfuerzo de requisitos y redactar un plan de requisitos generará buenos dividendos (Capítulo 1). (Los libros de Watts Humphrey, Introducción al proceso de software personal [1] y

La introducción al proceso de software de equipo^[2] proporciona buenos conocimientos sobre el valor de la planificación.¹)

Todos los proyectos, con la posible excepción de los proyectos "pequeños", requieren una herramienta de requisitos automatizada de potencia industrial (Capítulos 2 y 5).

Inicie el proceso de selección de la herramienta de su elección temprano en la fase de planificación del proyecto, guiado por un estudio comercial para garantizar que la herramienta seleccionada respalde el proceso de su proyecto y las necesidades de su cliente. Invierta en capacitación formal para las personas que más utilizarán la herramienta.

El mayor problema en informática, ingeniería de sistemas y software es la falta de identificación de los requisitos reales antes de iniciar otros trabajos, lo que resulta en costosos retrabajos (un promedio del 40% al 50% de los costos totales del proyecto, pero la cantidad puede ser mucho más alto, llegando directamente a la cancelación del proyecto), incumplimiento de presupuestos y cronogramas, mala calidad de los productos de trabajo, insatisfacción del cliente y fracaso del proyecto (Capítulo 5). Un equipo conjunto puede ayudar a superar este problema. El segundo mayor problema en la industria de desarrollo de sistemas y software es la falta de control de los nuevos requisitos y los cambios en los mismos (Capítulo 5). Proporcionar un mecanismo formal puede ayudar a superar este problema.

Aunque algunos requisitos no son "conocibles" al comienzo de un esfuerzo de desarrollo de sistema o software, se puede emplear un ciclo de vida de desarrollo incremental para gestionar la situación (Capítulo 5). Además, utilizar un desarrollo rápido de aplicaciones y un enfoque en espiral puede ayudar a identificar requisitos reales.

Se identificaron nueve funciones de la AR. Se proporcionaron sugerencias e ideas sobre cómo desempeñar cada función (Capítulo 2).

Se describieron las habilidades y características que necesita un AR eficaz (Capítulo 3). Una "matriz de habilidades de RA" le ayudará a guiar su desarrollo profesional. Continuar su educación puede ayudarlo a adquirir conocimientos expertos en ingeniería de requisitos y utilizar prácticas de requisitos efectivas.

Se proporcionaron varios estudios de casos que ayudan a ilustrar cómo las tareas de requisitos pueden salir mal (todos los capítulos). El desarrollo y la gestión de requisitos son difíciles. Aplicar procesos, métodos, técnicas y herramientas en el mundo real con diversos clientes y usuarios es particularmente desafiante.

Se identificaron varios tipos de requisitos (Capítulo 4). Se sugirió la terminología recomendada para usar y no usar, basada en

1. Una de las mejores prácticas de la industria que podría sugerir a su organización es el Proceso de software personal (PSP) y el Proceso de software de equipo (TSP). Watts Humphrey ha demostrado que los datos y la planificación son muy poderosos a nivel de desarrollador individual y para equipos de desarrolladores.

experiencia de la industria en el uso de los distintos términos, para reducir la confusión y facilitar una comunicación efectiva. Será útil utilizar un glosario de proyectos y una lista de acrónimos de proyectos.

Se proporcionó una lista de verificación de recopilación de requisitos que ayudará a su proyecto a realizar el trabajo necesario de manera eficiente y efectiva (Capítulo 5). Será útil tener un procedimiento documentado que usted adapte para satisfacer las necesidades de su proyecto. Le sugerimos que se planifique el esfuerzo y que se tome un tiempo periódicamente para evaluar cómo le está yendo en comparación con el plan (PDCA).

Es posible que necesite volver a planificar. En cualquier caso, queremos establecer un valor de mejora continua y captar las ideas y sugerencias de los compañeros de trabajo en cada paso del camino, fomentando el trabajo en equipo (Capítulo 8).

Su proyecto debe adoptar un conjunto de mejores prácticas para las AR (Capítulo 6). Se habla y se escribe mucho sobre las mejores prácticas, pero no hay suficiente implementación e institucionalización real de su uso. De hecho, un problema endémico en informática, ingeniería de sistemas y software es que no practicamos lo que predicamos: necesitamos hacer esfuerzos concertados para utilizar (no sólo reconocer) buenas prácticas, procesos, mecanismos, métodos, técnicas y herramientas. a nivel individual, de proyecto y organizacional. La razón por la que no hacemos esto es que requiere mucho trabajo. La cuestión es que debemos esforzarnos por hacer que nuestro trabajo sea eficiente y eficaz, de modo que podamos apoyar mejor los objetivos empresariales y organizacionales.

La discusión sobre un conjunto de habilidades especiales de RA proporciona información sobre áreas que pueden ser críticas en algún momento de su trabajo (Capítulo 7).

Se necesita un enfoque de calidad integrado para que los procesos (incluido el proceso de requisitos) funcionen de manera más efectiva (Capítulo 8). Los RA se encuentran en una posición estratégica para ayudar a proyectos y organizaciones.

Se proporcionó una visión para la ingeniería de requisitos que incluye objetivos específicos que son alcanzables (Capítulo 9).

A dónde ir desde aquí

Hemos cubierto mucho terreno en este libro. La Figura 10.1 muestra al AR, ingeniero o gerente considerando algunas de las herramientas disponibles para ayudar.

Considere los siguientes temas en términos de algunos de los próximos pasos. Estos son algunos cursos de acción que espero que consideres:

1. Considere los roles que proporciona en sus asignaciones de trabajo actuales en el contexto del Capítulo 2. ¿Es posible que pueda ampliarlos, haciendo así una mayor contribución, aprendiendo más y sintiéndose más realizado en sus actividades laborales?

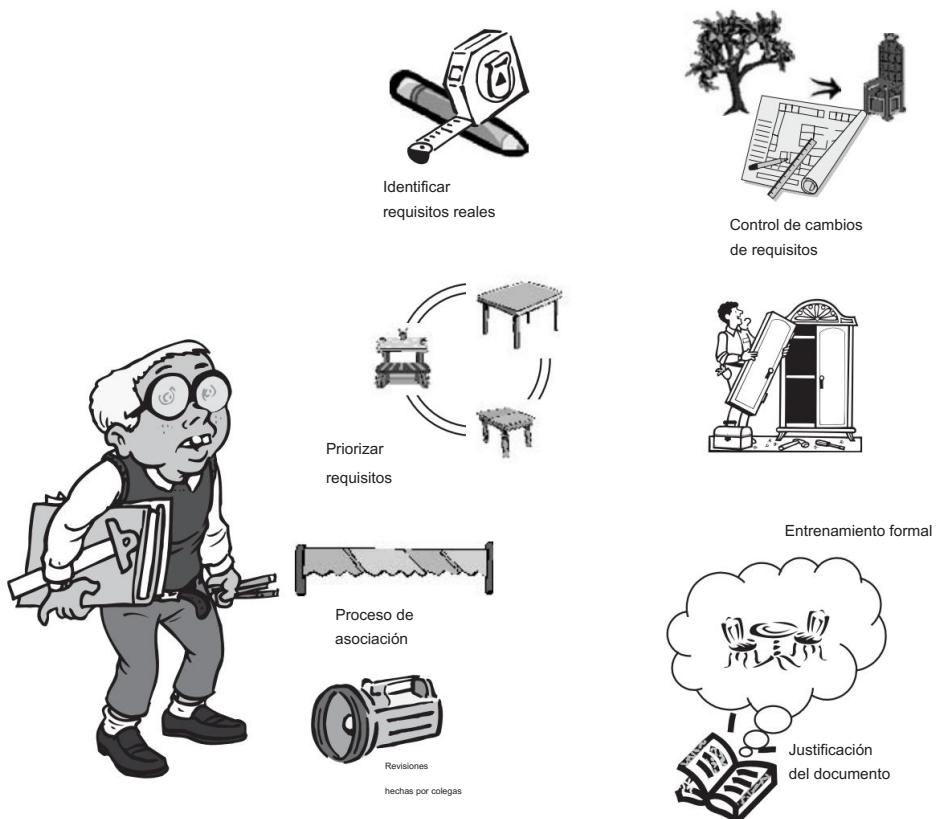


Figura 10.1 Hay buenas herramientas disponibles para ayudar. (Fuente: Richard Raphael. Usado con permiso.)

2. Reflexione sobre las habilidades que posee y las características que exhibe en el contexto del Capítulo 3. Quizás en colaboración con su gerente, desarrolle un plan de desarrollo profesional personal que le permita adquirir habilidades adicionales que considere deseables. Considere tomar medidas para fortalecer las características que le permitirán ser un mejor compañero de trabajo ante los ojos de sus compañeros.
3. En muchos proyectos existe confusión respecto a los tipos de requisitos. Si su proyecto presenta esta confusión, ¿qué pasos podría sugerir para aclararla?
4. ¿Utiliza un proceso de requisitos documentado en su tarea o proyecto? ¿Su grupo de trabajo ha aprovechado algún tiempo juntos para considerar posibles mejoras que se pueden hacer al proceso? ¿Podría ser esta una oportunidad para iniciar o reforzar un hábito de mejora continua?
5. ¿Son efectivos los métodos y técnicas de recopilación de requisitos que se utilizan? ¿Se podrían realizar algunas mejoras basadas en la experiencia de la industria con métodos y técnicas de recopilación de requisitos?

¿Sería útil una formación adicional sobre uno o más métodos o técnicas?

6. ¿Cuántas de las mejores prácticas para AR descritas en el Capítulo 6 se utilizan en su proyecto y en su organización? ¿Existen prácticas adicionales que tendría sentido considerar utilizar?
7. ¿Es aconsejable invertir algún esfuerzo en fortalecer alguna de las especialidades de las AR? habilidades especiales descritas en el Capítulo 7?
8. ¿Existe un enfoque de calidad integrado en su tarea o proyecto y en su organización? ¿Cuáles son algunos pasos que usted puede tomar para fortalecer y mejorar aún más el enfoque? ¿Podría ser útil brindar una sesión informativa a sus grupos de gestión para solicitar apoyo y patrocinio adicional?
9. ¿Cuál es su visión de la ingeniería de requisitos? ¿Cómo puedes trabajar para lograrlo?

Avanzando

Es posible que descubras (como yo) que la lectura estimula tus pensamientos sobre cómo seguir adelante. Leer no requiere necesariamente digerir cuidadosamente cada oración. Puede captar la esencia de un trabajo (libro, artículo, sitio web, presentación en una conferencia, etc.) mediante una revisión superficial, capturando algunas de las ideas principales y reflexionando sobre ellas a través del lente de su propia experiencia y creencias.

Recomiendo no intentar cambiar todo de una vez. Más bien, seleccione de tres a seis prácticas o áreas para su propia lista de compromiso personal. Estas son prácticas que estás dispuesto a aplicar en tus hábitos de trabajo diarios para cambiar de manera comprometida cómo haces las cosas en tu proyecto y entorno organizacional. Pueden referirse a áreas donde su proyecto u organización está recibiendo comentarios de un cliente sobre las necesidades de mejora, o áreas donde su proyecto u organización está teniendo algunos problemas (a veces se los denomina "puntos problemáticos").

A continuación se muestran algunos ejemplos de áreas de mejora candidatas que podrían ayudarle a crear su propia lista de compromiso:

1. Implementar un mecanismo para identificar requerimientos reales. Perseverar en hacer que el mecanismo sea exitoso y útil.
2. Documente la justificación de cada requisito (por qué es necesario).
3. Priorice los requisitos de su proyecto en colaboración con sus clientes y usuarios. Establezca aquellas que sean imprescindibles, prioridades altas, medias y bajas. Establecer una estrategia de entrega de productos que se base en las prioridades establecidas. Evaluar el riesgo de cada requerimiento, determinando su magnitud.
4. Implementar un mecanismo para controlar los cambios de requisitos y nuevos requisitos.

5. Proporcionar capacitación en requisitos formales para todas las personas involucradas en actividades relacionadas con los requisitos en su proyecto. Proporcione una sesión informativa sobre los requisitos para el equipo del proyecto y para sus clientes y usuarios. Hay una sesión informativa disponible en mi sitio web (www.ralphyoung.net) que podrías adaptarlo para este propósito.
 6. Iniciar revisiones por pares utilizando participantes capacitados en la revisión por pares y moderadores de revisión por pares. Considere realizar inspecciones de todos los documentos relacionados con los requisitos.
 7. Inicie un proceso de asociación en colaboración con su cliente para ganar compromiso con el éxito del proyecto.
 8. Diseñar o mejorar un proceso que parezca particularmente crucial para su objetivos y productos de trabajo en colaboración con los principales grupos de interés. Revisar por pares el borrador del proceso documentado e incorporar los comentarios de la revisión por pares. Entrene el proceso, luego implemente y Impleméntalo. Recoger datos e información sobre los resultados. Despues de haber estado en uso durante tres a seis meses, considere algunas ideas de mejora (PDCA).
 9. Cuantifique el costo estimado de retrabajo en su proyecto y comprométase a reducir el retrabajo en un porcentaje razonable (digamos 20%) mediante implementar un conjunto de mejoras en los procesos. Desarrollar y rastrear datos sobre el ROI de la iniciativa de mejora de procesos. Comprometer su gestión en la iniciativa.
 10. Defina y documente una política de requisitos para su proyecto o organización. Revisar por pares el borrador de la política documentada e incorporar los comentarios de la revisión por pares. Entrene la política, luego impleméntela y Impleméntalo. Recopilar datos e información relativa a los resultados. Per-severo en el uso continuo de la póliza.
 11. Considere seleccionar, capacitarse, implementar y utilizar un herramienta de requisitos automatizada. Proporcionar capacitación formal sobre nuevas herramientas automatizadas es una buena inversión. Además, considere hacer que el proveedor de la herramienta proporcione unos días de servicios de consultoría para explique su uso y adáptelo al entorno del proyecto específico y a las necesidades de su equipo.
 12. Considerar el uso (o el fortalecimiento del uso de) la trazabilidad bidireccional de requisitos utilizando su herramienta de requisitos automatizada.
 13. Proporcionar una formación de formación de equipos para su proyecto u organización. quizás facilitado por un facilitador capacitado externo a la organización. Durante la capacitación, mantenga una lista de las ideas aportadas por el participantes sobre cómo el proyecto u organización puede mejorar. Al finalizar la sesión de capacitación, lograr consenso²
2. Una técnica para lograr consenso es la votación múltiple. Vea la historia de QI de Six Sigma Qualtec: herramientas y técnicas, una guía a la resolución de problemas, [3, págs. 47–48].

entre los participantes sobre la prioridad de cada idea sugerida. Implementar de una a seis ideas principales (el número de ideas a implementar está determinado por el grado de compromiso de la gerencia con las ideas y los recursos disponibles).³

14. Lograr consenso sobre un conjunto de “reglas de conducta” que los miembros de su tarea o proyecto acepta apoyar. Documente y publique sus reglas de conducta. Proporcionar carteles en cada sala de reuniones. Hacernos responsables unos a otros ante ellos.
 15. Desarrolle un plan de requisitos para su proyecto. Revisión por pares del borrador del plan de requisitos. Entrenar el plan, estando abierto a sugerencias de mejora por parte de los participantes. Implementar e implementar el plan. Recopilar datos e información sobre los resultados observados durante su despliegue e implementación. Evaluar el impacto de crear e implementar un plan de requisitos y considerar iniciar una política o procedimiento organizacional que requiera o fomente el desarrollo de un plan de requisitos para todos los nuevos proyectos o esfuerzos mayores que un número determinado de meses-persona de esfuerzo.
 16. Forme un RWG en su organización. Compartir y evaluar los requisitos, políticas, procesos, mecanismos, prácticas, métodos, técnicas y herramientas que han tenido mayor éxito en cada proyecto. Documente los resultados. Comprometerse a ampliar el uso de aquellos que han tenido más éxito. Durante un período de tiempo (de uno a tres años), desarrolle un conjunto de mejores prácticas de requisitos que funcionen mejor en su organización. Considere tener un grupo de correo electrónico de los miembros del GTR para proporcionar un mecanismo para hacer preguntas y compartir información.
 17. Abordar el tema “Mejorar la comunicación dentro de mi proyecto” de manera proactiva. Piense en 4 sugerencias para mejorar las comunicaciones entre la gestión de proyectos y los miembros y equipos del proyecto, sus clientes y las principales partes interesadas. Considere establecer un glosario del proyecto y una lista de acrónimos.
 18. Adquiera el hábito de “hacer PDCA” al final de cada reunión y al alcanzar los hitos. Documentar las ideas y sugerencias ofrecidas. Dar seguimiento a las ideas y sugerencias seleccionadas.
 19. Fortalecer aún más el reconocimiento de las personas por los aportes realizados a la tarea, proyecto u organización. Establecer un mecanismo para celebrar las contribuciones del equipo o del proyecto. Formar el hábito de
3. Un enfoque de implementación sugerido es pedirle a un defensor de la idea que escriba un plan de acción y dirija un equipo de MC a través de la historia de la MC. Consulte la historia de QI de Six Sigma Qualtec : herramientas y técnicas, una guía para la resolución de problemas, [3, págs. 94–96 y 23–41].
4. La lluvia de ideas es una valiosa herramienta de mejora de la calidad para recopilar sugerencias. Vea la historia de QI de Six Sigma Qualtec : Herramientas y técnicas, una guía para la resolución de problemas, [3, págs. 43–46].

- reconocer a las personas por nuevas ideas y productos de trabajo en curso.
- Pilla a la gente haciendo las cosas bien.
20. Inicie o fortalezca el uso de herramientas de mejora de la calidad en su proyecto u organización, como lluvia de ideas, votación múltiple, diagramas de Pareto, diagramas de espina de pescado (o Ishikawa), PDCA, reglas de conducta, análisis de barreras y ayudas, contramedidas y diagramas de flujo de procesos. .
21. Revisar los motivos de los errores de requisitos. Determinar las causas fundamentales.
Haga una lluvia de ideas y sugerencias (“contramedidas”) para abordar las causas fundamentales. Seleccione algunas contramedidas para implementar y realice un seguimiento de los resultados.
22. Examine si cree que las reuniones de su proyecto son tan efectivas como pueden ser.
Considere ideas para aprovechar mejor el tiempo de las reuniones.
23. Revisar cada seis meses el estado de tu programa de estudio personal y del programa de mejora de tu proyecto u organización. Pregunte: ¿Cómo estoy (o cómo estamos)?
¿Necesitamos volver a comprometernos y comprometernos? ¿Necesitamos cambiar nuestro enfoque? ¿Necesitamos involucrar al proyecto o a la alta dirección y solicitar un patrocinio más fuerte u otras acciones? Realice PDCA sobre el estado y obtenga cierto consenso sobre hacia dónde ir a partir de ahí. Actúe según sus hallazgos.
- Establezca hitos a corto plazo y estadíquelos semanalmente.
24. Asegúrese de que los requisitos se utilicen y de que no se conviertan en “artículos de estantería”. Consiga el apoyo de los desarrolladores y, con la gerencia, articule la expectativa de que la capacitación sobre requisitos que se les brinde sea un elemento integral del desempeño esperado.
éxito.
25. Aumentar la visibilidad de la recopilación de requisitos: asegúrese de que todos los desarrolladores sepan cuándo ven requisitos nuevos o cambiantes y que deben comunicar esta información al CCB del proyecto; de lo contrario, el proyecto corre el riesgo de salirse de control.

Un mandala de requisitos

Un “mandala” es un diagrama: un círculo dividido en cuatro cuadrantes, cada uno con una simbología. Proporciona un mapa que resume dónde se encuentra un individuo u organización y hacia dónde se dirige.

La figura 10.2 proporciona un mandala relacionado con la ingeniería de requisitos.

El mandala de requisitos puede ayudarte a visualizar algunos de los próximos pasos que querrás dar. A propósito no he proporcionado listas de candidatos de cosas para incluir en cada cuadrante; esta debería ser su propia creación. En el cuadrante este, enumere algunas prácticas de requisitos que haya utilizado en el pasado o que se hayan aplicado en su proyecto o en su organización. En el cuadrante sur, enumere algunas fuentes de una situación mejorada: ideas sobre mejores



Figura 10.2 Un mandala de requisitos.

prácticas que podrían aplicarse. En el cuadrante oeste, enumere los objetivos que desea alcanzar mediante la aplicación de prácticas más efectivas. En el cuadrante norte, enumere las características del estado final deseado: metas personales, objetivos de proyecto, metas organizacionales u objetivos comerciales que desea lograr o apoyar.

Esta herramienta le brinda la oportunidad de reflexionar sobre todo lo que ha hecho en experiencias anteriores y actuales, así como lo que ha leído en este libro y lo que ha aprendido de otras maneras. Cada punto focal brinda una oportunidad para preguntas e ideas. Tómate unos momentos para enumerar algunas cosas en cada cuadrante y reflexiona sobre ellas. Además, mantenga algunas discusiones sobre esto entre los miembros de su tarea, proyecto o equipo organizacional.

Esto puede conducir a algunas direcciones útiles. Quizás cada uno debería tomarse un tiempo para dibujar su propio mandala.

Resumen Los RA

y los ingenieros se encuentran en una posición estratégica para mejorar los resultados del proyecto y las tasas de éxito. Los requisitos son la base de todo el trabajo posterior que se realiza en proyectos de ingeniería de software y sistemas. Mejorar las prácticas de requisitos que se utilizan puede tener enormes beneficios. Sea audaz al ofrecer su experiencia, energía y conocimientos.

Caso de estudio

Este estudio de caso es un ejemplo de cómo un proceso de RM eficaz y su cumplimiento pueden hacer que la resolución de problemas de interpretación de requisitos se realice sin problemas, incluso cuando los requisitos cambian durante el esfuerzo de desarrollo.

Un cliente cuestionó la implementación de un requisito de software por no cumplir con la intención del requisito del sistema de nivel superior durante una revisión del diseño de un proyecto. El cliente cuestionó el requisito de software por ser deficiente en comparación con el requisito del sistema principal, pero

reconoció que el diseño cumplía con los requisitos del software. Los requisitos de software para el sistema ya habían pasado por la revisión de requisitos de software y fueron aprobados por el cliente siguiendo el proceso de RM del proyecto. El cliente abrió un elemento de acción en la revisión del diseño como mecanismo para documentar formalmente el problema de interpretación de los requisitos del software. En esta etapa del proceso de diseño, sabíamos más sobre el entorno del cliente y acordamos (extraoficialmente) que tenía una preocupación válida. Estábamos demasiado avanzados en el proceso de desarrollo para realizar los cambios de diseño necesarios para abordar este elemento de acción en esta versión del producto, ya que los requisitos de software ya habían sido aprobados. Nuestro proceso de RM dictaminó que se presentara una mejora según el requisito de software aprobado que se abordará en una versión futura del producto. Esto permite que el proyecto continúe con el costo y el cronograma actuales.

Siguiendo nuestro proceso de ingeniería de sistemas, documentamos varias soluciones para abordar el problema. Las soluciones incluyeron modificar el diseño para abordar la nueva interpretación y proponer un proceso manual superpuesto con el diseño existente para abordar el problema abierto por el cliente. Se establecieron estimaciones de costos y cronogramas para cada una de las soluciones. Se informó a la alta dirección de ingeniería sobre el problema, los requisitos originales y las soluciones propuestas. La gerencia estuvo de acuerdo en que no podíamos afectar el cronograma y el presupuesto aprobados dado que el diseño cumplía con los requisitos de software aprobados. Nuestro equipo de gestión se había metido en problemas en un contrato anterior por intentar ser demasiado receptivos a la solicitud del cliente, lo que provocó retrasos en la entrega y sobrecostos. La dirección estaba decidida a no repetir ese error.

Llevamos la solución propuesta al elemento de acción al cliente en la reunión de cierre de la revisión del diseño. Señalamos mediante informes de seguimiento que habíamos descompuesto formalmente los requisitos del sistema en requisitos de software. Los requisitos descompuestos cumplieron con la intención según nuestro leal saber y entender en el momento de la aprobación. Los requisitos de software fueron aprobados como parte de la revisión de requisitos de software existentes. Presentamos la página de firma formal como prueba de esta aprobación.

Ahora habíamos establecido que el cliente era propietario de este problema. Cuando vieron las opciones propuestas, seleccionaron la que tenía la menor cantidad de cronograma e impacto en costos: el proceso manual superpuesto. La acción se cerró con una recomendación de proponer el cambio de diseño cuando se licitara la fase de mejora del producto planificada previamente del contrato. Reconocimos que el requisito no abordaba el problema del cliente planteado durante la revisión del diseño, pero el problema no se había planteado durante la revisión del requisito.

Al seguir nuestro proceso de RM pudimos evitar un gran impacto en los costos y el cronograma del programa. El cliente se hizo responsable del problema con nosotros. Desarrollamos un plan a corto plazo de bajo o ningún costo y acordamos incluir la solución permanente en la próxima versión del producto. Fue tentador decir: "Sí, señor cliente, tiene razón y modificaremos el diseño para satisfacer sus necesidades", pero esto se habría desviado de nuestro proceso y habría puesto en peligro la finalización del contrato dentro del plazo previsto.

cronograma y costo. Si hubiera sido un requisito crítico, habríamos pedido la aprobación del cliente para una nueva planificación de costos y cronograma, ya que se planteaba un nuevo requisito después de la aprobación de los requisitos.

don joven
ingeniero de requisitos
Corporación Telelógica

Referencias

- [1] Humphrey, WS, Introducción al proceso de software personal, Reading, MA:
Addison-Wesley, 1997.
- [2] Humphrey, WS, Introducción al proceso de software de equipo, Reading, MA:
Addison-Wesley, 2000.
- [3] Six Sigma Qualtec, QI Story: Tools and Techniques, A Guidebook to Problem Solving, 3.^a ed., 1999.
Se pueden solicitar copias llamando al (480) 586-2600.

Machine Translated by Google

Glosario

Asignación Asignar cada requisito a un componente del sistema donde se implementará.

Aplicación El uso de capacidades (servicios e instalaciones) proporcionadas por un sistema de información para satisfacer un conjunto de requisitos del usuario, como el procesamiento de textos.

Arquitectura La estructura subyacente de un sistema.

Artefacto Documento que representa el resultado del esfuerzo. A menudo se hace referencia a los artefactos como ejemplos de productos de trabajo necesarios para proporcionar evidencia que respalde las evaluaciones.

Atributo Característica de un requisito que es útil para ordenar, clasificar o gestionar requisitos.

Línea base Una especificación o producto que ha sido revisado y acordado formalmente y posteriormente sirve como base para un mayor desarrollo.

Se modifica únicamente mediante procedimientos formales de control de cambios.

Requisitos comerciales Las actividades esenciales de una empresa. Los requisitos comerciales se derivan de las metas comerciales (los objetivos de la empresa). Los escenarios comerciales se pueden utilizar como técnica para comprender los requisitos comerciales. Un factor clave en el éxito de un sistema es el grado en que el sistema respalda los requisitos del negocio y facilita que una organización los alcance.

Reglas comerciales Las políticas, condiciones y restricciones de las actividades comerciales respaldadas por un sistema; los procesos de decisión, directrices y controles detrás de los requisitos funcionales (por ejemplo, procedimientos, definiciones, relaciones y flujos de trabajo en el negocio, y conocimientos necesarios para realizar acciones).

Escenario empresarial Técnica que se puede utilizar para comprender una empresa u organización. Un escenario empresarial describe el proceso empresarial, la aplicación o el conjunto de aplicaciones, el entorno empresarial y tecnológico, las personas y los componentes informáticos que ejecutan el escenario y el resultado deseado de una ejecución adecuada.

Sistema empresarial Hardware, software, declaraciones de políticas, procedimientos y personas que en conjunto implementan una función empresarial.

Cliente Un componente de la aplicación que solicita servicios de un servidor.

Comercial listo para usar Un elemento de hardware o software que ha sido producido por un contratista y está disponible para compra general.

Complejidad El grado en que un sistema tiene requisitos, un diseño o una implementación que es difícil.

Gestión de la configuración Disciplina que aplica dirección y vigilancia técnica y administrativa para (a) identificar y documentar las características funcionales y físicas de un elemento de configuración; (b) controlar los cambios en esas características; y (c) registrar e informar cambios en el estado de procesamiento e implementación.

Restricción Un atributo necesario de un sistema que especifica limitaciones legislativas, legales, políticas, de políticas, procesales, morales, tecnológicas o de interfaz.

Cliente La(s) persona(s) con los fondos para pagar el proyecto o su producto final. El cliente no es necesariamente el usuario.

Necesidad del cliente Conjunto de requisitos deseados por un cliente.

Descomposición Separar los atributos de una necesidad del cliente (los requisitos de un sistema) para poder abordarlos.

Defecto Una variación de un atributo de producto deseado.

Prevención de defectos Tecnologías y técnicas (p. ej., SPC) que minimizan el riesgo de cometer errores en los entregables.

Eliminación de defectos Actividades que encuentran y corrigen defectos en los entregables.

Eficiencia de eliminación de defectos La relación entre defectos de desarrollo y defectos del cliente.

Requisito derivado Un requisito que se refina aún más a partir de un requisito de fuente primaria o de un requisito derivado de nivel superior, o un requisito que resulta de la elección de una implementación o elemento del sistema específico.

Diseño El proceso de definir la arquitectura, los componentes, las interfaces y otras características de un sistema.

Patrón de diseño Una descripción de un problema y su solución propuesta que indica cuándo aplicar la solución y las consecuencias. Véase Gamma et al., *Design Patterns* (Reading, MA: Addison-Wesley, 1995).

Desarrollo El proceso de transformar un diseño en componentes de hardware y software.

Experto en el dominio Una persona que ha estado trabajando en un campo particular durante un período extenso de tiempo y que está capacitado en esa área. A un experto en un dominio a menudo se le conoce como experto en la materia o PYME.

Empresa El nivel más alto de una organización.

Estudio de viabilidad Un análisis que proporciona una comprensión inicial del costo, la viabilidad, la arquitectura técnica de alto nivel y los requisitos de una capacidad o sistema.

Marco Estructura básica de ideas o marco de referencia.

Función Capacidad útil proporcionada por uno o más componentes de un sistema.

Punto de función Una medida de la complejidad del desarrollo de software.

Requisito funcional Un atributo necesario en un sistema que especifica lo que el sistema o uno de sus productos debe hacer.

Especificación o documento funcional Una colección completa de las características de un sistema y las capacidades que pondrá a disposición de los usuarios.

Proporciona un análisis detallado de los datos que se espera que manipule el sistema. Puede incluir una definición detallada de las interfaces de usuario del sistema.

Chapado en oro Agregar características y capacidades a los sistemas cuando no son requeridas por la especificación del sistema o los requisitos reales.

Sistema de información La parte informática de un sistema empresarial.

Tecnología de la información Una ciencia aplicada que utiliza hardware y software para apoyar la transferencia de ideas.

Institucionalización La construcción de infraestructura y cultura corporativa que respalden métodos, prácticas y procedimientos para que sean la forma continua de hacer negocios, incluso después de que quienes los definieron originalmente hayan desaparecido.

Interoperabilidad La capacidad de dos o más sistemas o componentes para intercambiar y utilizar información.

Iterar Repetir una secuencia de operaciones para producir resultados que se acercan sucesivamente al resultado deseado.

Interfaz La interacción o comunicación entre sistemas independientes o componentes de sistemas.

Equipo de producto integrado Un grupo que incluye clientes y desarrolladores y combina perspectivas en un todo funcional o unificado.

El equipo conjunto recomendado en este libro es un ejemplo de IPT.

Ciclo de vida El período de tiempo que comienza cuando se concibe un sistema y termina cuando el sistema ya no está disponible.

Modelo de ciclo de vida Marco de procesos y actividades relacionados con la evolución de un sistema que también actúa como referencia común para la comunicación y el entendimiento entre los participantes en el esfuerzo.

Defecto mayor Un problema que impide el uso efectivo de un producto de trabajo, como una deficiencia de diseño o el descubrimiento de requisitos conflictivos.

Mandala Un diagrama que proporciona un mapeo para un individuo o una organización que resume dónde está uno y hacia dónde va.

Medidas de efectividad Indicadores de alto nivel de qué tan bien el sistema realiza sus funciones, definidos en los términos y con la misma dimensión del documento de requisitos. Por ejemplo, si estamos tratando con el sistema de metro de una ciudad, podemos especificar que un usuario típico durante las horas pico hora no debe esperar más de un período de tiempo, en promedio, para el próximo tren.

Mecanismo Una forma de hacer algo o de lograr un resultado.

Método Una forma, técnica, proceso, plan, mecanismo, conjunto de habilidades o técnicas, disciplina, práctica, sistema, modelo, marco, capacidad o Procedimiento para hacer algo.

Metodología Conjunto de métodos, reglas y postulados empleados por un disciplina; un procedimiento o conjunto de procedimientos en particular.

Defecto menor Un problema que no impide el uso efectivo de una obra. producto, como un problema de formato, error ortográfico, problema de uso del idioma o acrónimo o definición no proporcionados o explicados.

Modelo Representación de la realidad que pretende facilitar comprensión.

Requisito no funcional Un atributo necesario en un sistema que Especifica cómo se deben realizar las funciones, a menudo mencionadas en los sistemas. ingeniería como las “ilidades” (por ejemplo, confiabilidad, reutilización, portabilidad, mantenibilidad, compatibilidad, verificabilidad, previsibilidad, seguridad, información). seguridad, eficiencia de recursos, exhaustividad y factores humanos).

Asociación Un proceso estructurado diseñado para crear una atmósfera de compromiso, cooperación y resolución colegiada de problemas entre organizaciones e individuos que trabajarán juntos.

Requisito de desempeño Un atributo necesario de un sistema que Especifica qué tan bien el sistema o uno de sus productos debe realizar una función, junto con las condiciones bajo las cuales se realiza la función.

Práctica La realización de actividades laborales repetidamente para volverse competente; la forma habitual de hacer algo para producir un buen resultado.

Requisitos priorizados Categorización de los requisitos reales en subconjuntos según la criticidad de la necesidad de un sistema o capacidad.

Marco del problema La definición de una clase de problema. Un marco problemático Consiste en un diagrama de marco, características de dominio y la preocupación del marco.

Véase Michael Jackson, Marcos de problemas: análisis y estructuración de problemas de desarrollo de software (Nueva York: Addison-Wesley, 2000).

Proceso Conjunto de actividades que resultan en la realización de una tarea o el logro de un resultado.

Capacidad del proceso La gama de resultados esperados que se pueden lograr siguiendo un proceso.

Descripción del proceso Un documento que describe un proceso, incluyendo, por ejemplo, su propósito, clientes, requisitos del cliente, criterios de entrada, entradas, salidas, criterios de salida, tareas involucradas y quién es responsable para cada uno, indicadores de medición, recursos necesarios y versión.

Diagrama de flujo del proceso Un diagrama que muestra una serie paso a paso de acciones a través de un procedimiento que utiliza líneas de conexión y un conjunto de estándares Símbolos adoptados por una organización.

Modelo de proceso Un marco para identificar, definir y organizar las estrategias funcionales, reglas y procesos necesarios para gestionar y respaldar la forma en que una organización hace o quiere hacer negocios. El proceso El modelo proporciona un marco gráfico y textual para organizar los datos y procesos en grupos manejables para facilitar su uso compartido y control en toda la organización.

Proyecto Un emprendimiento enfocado al desarrollo o mantenimiento de un producto. Normalmente un proyecto tiene su propia financiación, contabilidad y ejecución. cronograma.

Campeón del proyecto Un defensor que está muy familiarizado con el conjunto de Necesidades reales del cliente de un sistema que proporcione un papel activo en el esfuerzo de desarrollo, facilitando las tareas del equipo de desarrollo.

Gerente de proyecto o programa El rol con responsabilidad comercial total para un proyecto, responsable en última instancia ante un cliente.

Creación de prototipos Técnica para construir una versión rápida y aproximada de un sistema deseado o partes de ese sistema. El prototipo ilustra las capacidades del sistema a usuarios y diseñadores. Sirve como medio de comunicación. mecanismo para permitir a los revisores comprender las interacciones con el sistema. Les permite identificar problemas y considerar formas de mejorar una sistema. A veces da la impresión de que los desarrolladores están más más de lo que realmente es, lo que da a los usuarios una impresión demasiado optimista sobre las posibilidades de finalización.

Calidad Satisfacer las necesidades reales del cliente.

Cultura de calidad Presencia de una actitud de mejora continua y satisfacción del cliente en toda una organización.

Despliegue de la función de calidad Una metodología originalmente concebida en Japón en la década de 1970 que brinda una oportunidad para que el usuario y el desarrollador de un sistema comprendan los requisitos más completamente y prioricen a ellos.

Proceso Unificado de Rational Una metodología defendida por Rational Software, Inc. (ahora IBM, Inc.)

Requisitos reales Requisitos que reflejan las necesidades verificadas de un sistema o capacidad particular.

Requisito Un atributo necesario en un sistema; también una declaración que identifica una capacidad, característica o factor de calidad de un sistema para que tenga valor y utilidad para el usuario.

Asignación de requisitos Asignación de requisitos a los componentes arquitectónicos de un sistema (por ejemplo, una configuración de hardware o software). artículo, capacitación o documentación). A veces se lo denomina flujo descendente.

Análisis de requisitos Un método estructurado (organizado) para comprender los atributos que satisfarán la necesidad de un cliente.

Línea base de requisitos El conjunto de requisitos asociados con una lanzamiento particular de un producto o sistema.

Definición de requisitos Una descripción detallada en general, en lugar de más que funcionales, términos de los atributos necesarios en un sistema.

Derivación de requisitos Obtención de requisitos para un sistema a partir de fuentes proporcionadas por el cliente.

Documento de requisitos Un repositorio de los atributos de un sistema.

Obtención de requisitos El proceso de extraer y traer requisitos basados en la información proporcionada por el cliente.

Ingeniería de requisitos Un área dentro del campo más amplio de la ingeniería de sistemas y software que se centra en el proceso de ER.

Fuga de requisitos La adición o filtración de información no oficial requisitos a la especificación de requisitos, cuando los requisitos realmente no son necesarios.

Gestión de requisitos Seguimiento del estado de los requisitos y cambiar la actividad y rastrear los requisitos para las diversas fases y productos del esfuerzo de desarrollo.

Patrón de requisitos Un marco para el conjunto de requisitos que apoya las necesidades del producto o servicio, minimizando las lagunas en el conocimiento que podría provocar el fracaso del proyecto. El patrón de requisitos ayuda a capturar todos tipos de requisitos, independientemente del tipo de diseño, implementación o método utilizado para capturar e identificar los requisitos. Ver

P. Ferdinandi, Un patrón de requisitos: tener éxito en la economía de Internet (Boston: Addison-Wesley, 2002).

Proceso de requisitos Un conjunto de acciones del ciclo de vida completo del sistema relacionadas con los atributos necesarios de los sistemas. El proceso de requisitos implica Comprender las necesidades y expectativas del cliente (obtención de requisitos), análisis y especificación de requisitos, priorización de requisitos, derivación, partición y asignación de requisitos, requisitos.

seguimiento, gestión de requisitos, verificación de requisitos y validación de requisitos.

Trazabilidad de requisitos La capacidad de determinar la relación o conectividad entre las necesidades del cliente y los requisitos, o la relación de un requisito principal con un requisito secundario y viceversa. La capacidad de rastrear un requisito a lo largo del proceso de desarrollo del sistema, desde la especificación de requisitos hasta el diseño, pasando por los componentes del sistema, pasando por las pruebas y la documentación del sistema. Absolutamente crítico para todos los sistemas.

Verificación de requisitos Garantía independiente de que los requisitos se abordan y cumplen en un sistema.

Matriz de verificación de requisitos Un análisis que muestra el método de verificación para cada requisito.

Reutilizar Reutilizar tiene dos significados: (1) tomar el objeto X (por ejemplo, un objeto, subrutina o software COTS) que fue realizado por Y y usarlo directamente en otro proyecto; y (2) adaptar un producto de trabajo desarrollado (una especificación, un plan o proceso, por ejemplo).

Riesgo La posibilidad de sufrir una pérdida.

Arquitectura robusta Estructura subyacente de un sistema que puede satisfacer y adaptarse fácilmente a requisitos reales.

Rol Conjunto de responsabilidades definidas que pueden ser asumidas por uno o más individuos.

Escalabilidad La capacidad de crecer para adaptarse a mayores cargas de trabajo.

Escenario Técnica utilizada para comprender los requisitos.

Alta dirección Un rol suficientemente alto en la organización como para que su enfoque principal sea la vitalidad a largo plazo de la organización.

Calidad del software Software que combina las características de bajas tasas de defectos y alta satisfacción del usuario.

Especificación Documento que describe los requisitos técnicos y los procedimientos de verificación para artículos, materiales y servicios. Un resultado del proceso de análisis de requisitos.

Parte interesada Cualquiera que tenga interés en un sistema o en que posea cualidades que satisfagan necesidades particulares.

Requisitos establecidos Requisitos proporcionados por un cliente al comienzo de un esfuerzo de desarrollo de software o sistema. Para distinguirlo de las necesidades reales.

Experto en la materia Una persona que ha estado trabajando en un campo particular durante un período extenso de tiempo y que está capacitado en esa área. A un experto en la materia a menudo se le llama experto en el dominio.

Proveedor Una organización que contrata a un comprador para proporcionar un sistema.

Sistema Un conjunto integrado de personas, productos y procesos que proporcionan la capacidad de satisfacer una necesidad del cliente.

Ciclo de vida del sistema El conjunto de actividades involucradas en comprender la necesidad de un cliente, definir y analizar requisitos, preparar un diseño, desarrollar un sistema y probarlo, implementarlo, operarlo y mantenerlo, terminando con su retiro.

Ingeniería de sistemas Una disciplina técnica y de gestión que traduce una necesidad del cliente en un sistema que satisface la necesidad del cliente. Otra fuente afirma que la ingeniería de sistemas es el proceso iterativo, pero controlado, en el que se comprenden las necesidades del usuario y se convierten en un sistema operativo. El papel de la ingeniería de sistemas es: autoridad técnica en un proyecto; interfaz única para el cliente y el proyecto; diseño de arquitectura y sistemas; derivación, asignación e interpretación de requisitos; y otros.

Adaptación La actividad de modificar, elaborar o adaptar un proceso o documento para otro uso. La reutilización de artefactos personalizados ahorra tiempo y dinero y es una ventaja de un enfoque orientado a procesos.

Trabajo en equipo Apoyo proactivo unos a otros; necesario para el éxito de cualquier empresa importante. La colocación física facilita el trabajo en equipo y puede ser un requisito previo para el éxito.

TRABAJO EN EQUIPO Un entorno o entorno de trabajo donde se valora y aprecia el trabajo conjunto como un equipo eficaz.

Medidas de desempeño técnico Indicadores de qué tan bien funciona el sistema y qué tan bien se cumplen los requisitos; estimaciones o medidas de los valores de parámetros esenciales de rendimiento. Las medidas de desempeño técnico se utilizan para evaluar el impacto en el costo, el cronograma y el esfuerzo técnico.

Especificación técnica Una colección completa de los detalles de cómo se implementará un sistema, incluida la arquitectura técnica (hardware y software), la descomposición del sistema en subsistemas, la identificación de los módulos comunes que se desarrollarán y otros detalles que requieren definición para permitir el desarrollo del sistema.

Técnica Un conjunto de reglas a seguir para realizar una tarea, un tratamiento de detalles técnicos, un conjunto de métodos técnicos o un método para lograr un objetivo deseado.

Inserción de tecnología Agregar nueva tecnología a un sistema durante todo el ciclo de vida del sistema.

Herramienta Algo utilizado para facilitar la realización de una operación o la práctica de un proceso o actividad.

Estudio de comercio Análisis de cursos de acción alternativos en el que se realiza un equilibrio de factores, todos los cuales no se pueden obtener al mismo tiempo.

Lenguaje de modelado unificado (UML) Una notación de propósito general (una forma de documentar) que describe el comportamiento estático y dinámico de un sistema. No es un método de diseño ni un proceso de desarrollo.

Caso de uso Imagen de las acciones que realiza un sistema y que representa a los actores.

Basado en casos de uso Descripción el comportamiento de un sistema en función de cómo interactúan los usuarios con el sistema.

Modelo de casos de uso Una descripción del comportamiento funcional de un sistema que incluye todos los actores y todos los casos de uso a través de los cuales los actores interactúan con el sistema.

Usuario El individuo o grupo que utiliza un sistema en su entorno.

Fácil de usar Fácil de usar.

Perspectiva del usuario Mantener la visión de lo que el usuario quiere, necesita, prefiere, con lo que está contento y puede utilizar.

Satisfacción del usuario Calidad de los clientes satisfechos con los productos, niveles de calidad, facilidad de uso y soporte de un proveedor.

Validación Un proceso para confirmar que los requisitos reales se implementan en el sistema entregado.

Verificación Un proceso para asegurar que la solución de diseño satisface los requisitos.

Métodos de verificación Los enfoques utilizados para realizar la verificación: prueba, inspección, demostración y análisis.

Vista Una perspectiva de un sistema, como la vista funcional, de implementación o física.

Producto de trabajo Algo producido o creado como resultado de la actividad de desarrollo de sistemas o software.

Machine Translated by Google

Lista de acrónimos

AI	elemento de acción
	También conocido como
AP	plan de ACCION
APM	Asociación de directores de proyectos (en el Reino Unido)
BOE	base de estimación
BSI	Instituto Británico de Estándares
BT	Telecomunicaciones británicas
CUIDADO	Ingeniería de requisitos asistida por ordenador, herramienta de requisitos comercializada por Sophist Technologies (Alemania)
CCB	Tablero de control de configuración o tablero de control de cambio.
CM	gestión de configuración
CMM	Modelo de Capacidad de Madurez
CMMI	Integración del modelo de madurez de capacidad
CMP	plan de gestión de configuración
Grupo de trabajo de gestión de configuración CMWG	
Concepto de operaciones del CONOPS	
CENTRO	una herramienta de requisitos comercializada por Vitech Corporation
CUNAS	software comercial listo para usar
CR	solicitud de cambio
DACS	Centro de análisis y datos para software
DAR	resolución de análisis de decisión
DES	Soluciones empresariales de defensa
Departamento de Defensa de EE. UU.	
DOORS Sistema de requisitos dinámico orientado a objetos, una herramienta de requisitos comercializada por Telelogic Corporation	
DP	prevención de defectos

Departamento de Defensa

EIA	Asociación de Industrias Electrónicas (estándares de la industria grupo)
EPG	grupo de procesos de ingeniería
PAI	mejora de procesos de ingeniería
EPIP	plan de mejora de procesos de ingeniería
ESE	entorno de software de ingeniería
FD	documento funcional
FP	punto de función
FPA	análisis de puntos de función
GSA	Administración de Servicios Generales
GUI	interfaz gráfica del usuario
ICRE	Conferencia IEEE sobre ingeniería de requisitos
IDEF	definición integrada para el modelado funcional
<small>ES DECIR</small>	
	estimación de impacto
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
IFPUG	Grupo de usuarios de puntos de función internacionales
ILS	soporte logístico integrado
INCOSE Consejo Internacional de Ingeniería de Sistemas	
ÉL	tecnologías de la información
YO ASI	Organización de Estándares Internacionales
IV&V	verificación y validación independientes
JAD	desarrollo de aplicaciones conjuntas
jt	equipo conjunto
KPA	área de proceso clave
TDM	desarrollo impulsado por modelos
MQ	cuestionario de madurez
NGC	Corporación Northrop Grumman
O&M	operación y mantenimiento
OMA	Arquitectura de gestión de objetos
DIOS MÍO	grupo de administración de objetos
OOO	orientado a objetos
OOSE	ingeniería de software orientada a objetos
OSD/ AT&L	Oficina del Secretario de Defensa, Adquisiciones, Tecnología, y Logística
<small>Pensilvania</small>	
	área de proceso

Lista de acrónimos

CAMARADA	biblioteca de activos de proceso; Propósito, agenda y límite (en relación con una reunión).
PBS	estructura de desglose del producto
PD	descripción del proceso
PDCA	planificar-hacer-verificar-actuar
Pi	la mejora de procesos
PEPITA	plan de mejora de procesos
PM	director de programa, director de proyecto
PMP	plan de gestión del programa, plan de gestión del proyecto
PÁGINAS	planificación de proyecto o plan de programa
revisiones pares	revisión por pares
PRINCE es compatible con una metodología de gestión de proyectos registrada por la APM, estrechamente aliada de las normas británicas Instituto	
PSP	proceso de software personal
control de calidad	seguro de calidad
QAI	Instituto de Garantía de Calidad
QFD	Despliegue de la función de calidad
QI	mejora de calidad
QIDW	calidad en el trabajo diario (también conocido como proceso gestión)
control de calidad	gestión cuantitativa
QMB	junta de gestión de calidad
I+D	investigación y desarrollo
analista de requisitos	analista de requisitos
RAD	diseño rápido de aplicaciones, desarrollo rápido de aplicaciones
RD	documento de requisitos; desarrollo de requisitos (un área de proceso en CMMI)
RDM	gestión basada en requisitos
RE	requisitos
Gestión de requisitos REQM (acrónimo utilizado en CMMI para esta área de proceso)	
RESG	Grupo de especialistas en ingeniería de requisitos (en el Reino Unido)
RFC	solicitud de cambio
RFI	Solicitud de Información
RFP	solicitud de propuestas

Solicitud de cotización	solicitud de presupuesto
ReqPro	Requisite Pro (una herramienta de requisitos comercializada por IBM [anteriormente Rational] Corporation)
RM	gestión de requerimientos
RMA	Asistente del Gerente de Riesgos (una herramienta de riesgo)
retorno de la inversión	Retorno de la inversión
PR	plan de requisitos
RRB	junta de revisión de requisitos
RT	Requisitos Tracer, una herramienta de requisitos comercializada por Ingeniería Teledyne Brown (también conocida como Xtie-RT)
RTM	Requerimientos de trazabilidad matriz
RTM	una herramienta de requisitos comercializada por Integrated Chipware
Taller	
RUP	Proceso racional unificado
RWG	grupo de trabajo de requisitos
S&PE	Organización de Ingeniería de Sistemas y Procesos en DES, NGC
SA	arquitectura del sistema
Evaluación CMMI	estándar de SCAMPI para la gestión de procesos
	Mejora
SCE	evaluación de la capacidad del software
plan de desarrollo de software	
SE	Ingeniería de Sistemas
Modelo de madurez de capacidad de ingeniería de sistemas	SE-CMM (EPIC)
SEI	Instituto de Ingeniería de Software
SEMP	plan de gestión de ingeniería de sistemas
SEP	proceso de ingeniería de sistemas
SEPG	grupo de procesos de ingeniería de sistemas o software
SLA	acuerdo de nivel de servicio
PIZARRA	una herramienta de requisitos automatizada
SLC	ciclo de vida de los sistemas
PYME	experto en la materia
SEMBRAR	Declaración de trabajo
Ingeniería de productos de software SPE o PE	
Especificaciones	especificación
SPI	mejora de procesos de software
SPIP	plan de mejora de procesos de software

ESP	Software Productivity Research, una subsidiaria de propiedad total de los sistemas de gestión Artemis
SS	especificación del sistema
ETS	estándar
SRR	revisión de requisitos del sistema
SRS	especificación de requisitos del sistema o software
SSQ	Sociedad para la calidad del software
ESTQE	Pruebas de software e ingeniería de calidad (una publicación periódica; consulte www.stqemagazine.com).
SUNÁ	análisis de las necesidades del usuario basado en escenarios
SUDOESTE	software
Modelo de madurez de capacidad SW-CMM para software (desarrollado por SEI)	
SUE	Ingeniería de software
Sinergia RM	una herramienta de requisitos comercializada por CMD Corporation
TBQ	cuestionario basado en taxonomía
gestión total de la calidad	
TSP	proceso de software de equipo
ubl	Lenguaje empresarial universal
UML	Lenguaje de modelado unificado
URL	Localizador Uniforme de Recursos
V&V	verificación y validación
Vital Link una herramienta de requisitos comercializada por Compliance Automation	
EDT	estructura de desglose del trabajo
Xtie-RT	rastreador de requisitos, una herramienta de requisitos comercializada por Teledyne Brown Engineering
ZF	marco de zachman

Machine Translated by Google

Bibliografía

ABT Corporation, "Core Competencies for Project Managers", Libro Blanco, 2000. Véase www.tsepm.com/may00/art5.htm.

Adams, James L., *Blockbusting conceptual: una guía para mejores ideas* (3.^a ed.), Reading, MA: Perseus Books, 1986.

Adhikari, Richard, "El proceso de desarrollo es un esfuerzo mixto", *Client/Server Computing*, febrero de 1996, págs. 65–72.

Afors, Cristina y Marilyn Zuckermann Michaels, "A Quick Accurate Way to Determine Customer Needs", Sociedad Estadounidense para la Calidad: *Quality Progress*, julio de 2001, págs. 82–87.

Alexander, Ian, sitio web, easyweb.easynet.co.uk/~iany/index.htm.

Alexander, Ian F. y Richard Stevens, *Writing Better Terms*, Londres, Reino Unido: Addison-Wesley, 2002.

Alexander, Ian y Andrew Farncombe, John Boardman Associates (JBA), Plantilla de análisis de partes interesadas, Curso básico de ingeniería de sistemas, 2003.

Sitio web de la Sociedad Estadounidense para la Calidad (ASQ): www.asq.org/.

Andriole, Stephen J., *Gestión de requisitos del sistema: métodos, herramientas y casos*, Nueva York: McGraw Hill, 1996.

Bach, James, "James Bach on Risk-Based Testing: How to Conduct Heuristic Risk Analysis", Revista Software Testing and Quality Engineering (STQE), noviembre/diciembre de 1999, págs.

Bachmann, Felix, Len Bass, Gary Chastek, Patrick Donohoe y Fabio Peruzzi, *The Architecture Based Design Method*, Instituto de Ingeniería de Software, Informe técnico CMU/SEI-2000-TR-001, ESC-TR-2000-001, 2000.

Bass, Len, Paul Clements y Rick Kazman, *Arquitectura de software en la práctica*, Reading, MA: Addison-Wesley, 1998.

Bennis, Warren y Patricia Ward Biederman, *Genio: los secretos de la colaboración creativa*, Reading, MA: Perseus Books, 1997.

Bentley, Colin, PRINCE® 2: Un manual práctico, Woburn, MA: Butterworth-Heinemann, 1997.

Berezuk, Steven P., con Brad Appleton, Patrones de gestión de configuración de software: trabajo en equipo eficaz, integración práctica, Boston, MA: Addison-Wesley, 2003.

Bicknell, Barbara A y Kris D., Bicknell, La hoja de ruta hacia el éxito repetible: uso de QFD para implementar el cambio, Boca Raton, FL: CRC Press, 1995.

Boehm, Barry W., Economía de la ingeniería de software, Englewood Cliffs, Nueva Jersey: Prentice Hall, 1981.

Boehm, Barry, "Modelo en espiral de desarrollo y mejora de software", IEEE Computer, mayo de 1988 (también publicado en Barry Boehm, Software Risk Management, IEEE Computer Society Press, 1989, p. 26).

Boehm, Barry W., WinWin Spiral Model & Groupware Support System, Universidad del Sur de California, 1998. Disponible en sunset.usc.edu/research/WINWIN/index.html.

Boehm, Barry, Alexander Egyed, Julie Kwan, Dan Port, Archita Shah y Ray Madachy, "Using the WinWin Spiral Model: A Case Study", IEEE Computer, julio de 1998, págs. 33–44.

Boehm, BW y Hoh In, "Identificación de conflictos entre requisitos de calidad", Software IEEE, marzo de 1996, págs. 25–35.

Boehm, Barry W. y Kevin J. Sullivan, "Software Economics", CrossTalk, The Journal of Defense Software Engineering, diciembre de 1999. Véase www.stsc.hill.af.mil/.

Boehm, Barry y Richard Turner, "Observaciones sobre el equilibrio de la disciplina y la agilidad", extraído de Equilibrio de la agilidad y la disciplina: una guía para los perplejos, Boston, MA: Addison-Wesley, 2003.

Boehm, Barry y Wilfred J. Hansen, "The Spiral Model as a Tool for Evolutionary Acquisition", un esfuerzo conjunto del Centro de Ingeniería de Software de la Universidad del Sur de California y el Instituto de Ingeniería de Software (SEI), CrossTalk, mayo de 2001, págs . 4–11.

Brodmann, Judith G. y Donna L. Johnson, La CMM personalizada de LOGOS para pequeñas empresas, organizaciones pequeñas y proyectos pequeños, LOGOS International, Inc. www.tiac.net/users/johnson.

Brodmann, Judith G. y Donna L. Johnson, "Retorno de la inversión (ROI) de la mejora de procesos de software medido por la industria estadounidense", Práctica y mejora de procesos de software, Sussex, Inglaterra: John Wiley & Sons Ltd., 1995, págs. 35–47.

Buede, Dennis M., El diseño de ingeniería de sistemas: modelos y métodos, Nueva York: John Wiley & Sons, 2000.

Butler, K., "Los beneficios económicos de la mejora de procesos de software", Diafonía, 1995, págs. 28–35.

- Proyecto de Integración del Modelo de Madurez de Capacidades (CMMI). Ver www.sei.cmu.edu/cmmi/.
- Carr, Frank et al, Asociación en la construcción: una guía práctica para el éxito del proyecto, Chicago, IL: American Bar Association Publishing, 1999.
- Clark, Bradford K., "Efectos de la madurez del proceso en el esfuerzo de desarrollo", Disponible en www.ralphyoung.net/goodarticles/.
- Cockburn, Alistair, Redacción de casos de uso eficaces, Boston, MA: Addison-Wesley, 2001.
- Centro de análisis y datos de software (DACS), Informes técnicos de DACS. Consulte www.dacs.dtic.mil/techs/tr.shtml.
- Daughtry, Taz (ed.), Conceptos fundamentales para el ingeniero de calidad de software, Milwaukee, WI: ASQ Quality Press, 2002.
- Davis, Alan M., Gestión de requisitos suficientes (Redmond, WA: Microsoft Press, de próxima aparición).
- Davis, Alan M., "El arte de la clasificación de requisitos", IEEE Computer, IEEE Computer Society Press, vol. 36, núm. 3, marzo de 2003, págs. 42–49.
- Davis, Alan M., Requisitos de software: objetos, funciones y estados, Upper Saddle River, Nueva Jersey: Prentice Hall PTR, 1993.
- Deming, W. Edwards, Fuera de la crisis, Centro de Estudios de Ingeniería Avanzada del MIT, 1986.
- Dion, R., "Process Improvement and the Corporate Balance Sheet", IEEE Software, octubre de 1993, págs. 28–35.
- Eckes, George, Hacer que Six Sigma dure: gestionar el equilibrio entre el cambio cultural y técnico, Nueva York: John Wiley & Sons, Inc., 2001.
- Electronic Industries Alliance (EIA), Procesos estándar 632 de la EIA para diseñar un sistema, Arlington, VA, 1998.
- Asociación de Industrias Electrónicas (EIA), Estándar EIA 649, Estándar de consenso nacional para la gestión de configuración, Arlington, VA, 2001.
- Colaboración para la mejora de procesos de ingeniería (EPIC), un modelo de madurez de capacidad de ingeniería de sistemas, versión 1.1. Pittsburgh, PA: Instituto de Ingeniería de Software, Universidad Carnegie-Mellon, 1995, disponible en www.sei.cmu.edu/pub/documents/95.reports/pdf/mm003.95.pdf.
- Publicaciones de estándares federales de procesamiento de información (FIPS PUBS) 183, Definición de integración para modelado de funciones (IDEF0). Disponible en www.itl.nist.gov/fipspubs/idef02.doc.
- Feldmann, Clarence G., La guía práctica para la reingeniería de procesos empresariales utilizando IDEF0, Nueva York: Dorset House, 1998.
- Ferdinandi, Patricia L., Un patrón de requisitos: tener éxito en la economía de Internet, Boston, MA: Addison-Wesley, 2002.

Fowler, Martin, Patrones de análisis: modelos de objetos reutilizables, Reading, MA: Addison-Wesley, 1996.

Fowler, Martin, UML destilado: aplicación del lenguaje estándar de modelado de objetos, Reading, MA: Addison Wesley, 1997.

Gaffney, Steven, Sea honesto, Arlington, VA: JMG Publishing, 2002.

Gaffney, sitio web de Steven, www.StevenGaffney.com.

Gamma, Erich y otros, Patrones de diseño, Reading, MA: Addison-Wesley, 1995.

Garmus, David y David Herron, Análisis de puntos funcionales: prácticas de medición para proyectos de software exitosos, Reading, MA: Addison-Wesley, 2001.

Gilb, Tom, Impact Estimation Tables: Understanding Complex Technology Quantitatively, noviembre de 1997. Libro blanco disponible en el sitio web de Gilb: www.gilb.com.

Gilb, Tom, "Planificación para aprovechar al máximo la inspección", en Conceptos fundamentales para el ingeniero de calidad de software, Taz Daughtrey (ed.), p. 178.

Gilb, Tom, Principios de gestión de ingeniería de software, Harlow, Inglaterra: Addison-Wesley, 1988.

Gilb, Tom y Dorothy Graham, Inspección de software, Boston, MA: Addison-Wesley, 1993.

Materiales de META/QPC. Consulte www.goalqpc.com.

Gottesdiener, Ellen, Requisitos por colaboración, Reading, MA: Addison-Wesley, 2002.

Gottesdiener, Ellen, "Diez formas principales en que los equipos de proyectos hacen mal uso de los casos de uso y cómo corregirlos", disponible en www.therationaledge.com/content/jun_02/t_misuseUseCases_eg.jsp.

Gottesdiener, Ellen y Jim Bruce, "El valor de la estandarización de las reglas comerciales", disponible en www.ebgconsulting.com/BusRulesObjectMagsHTML.html.

Gottesdiener, Ellen, "Capturing Business Rules", Revista de desarrollo de software, vol. 7, N° 12, diciembre de 1999. Disponible en www.sdmagazine.com/.

Gottesdiener, Ellen, "Turning Rules Into Requests", Application Development Trends, julio de 1999. Disponible en www.adtmag.com/print.asp?id=3806.

Grady, Jeffrey O., Validación y verificación del sistema, Boca Raton, FL: CRC Press, 1997.

Grady, Jeffrey O., Análisis de requisitos de sistemas, Nueva York: McGraw-Hill, 1993.

Hadden, Rita, Liderando el cambio cultural en su organización de software, 2003.

Hadden, Rita, "¿Cuán escalables son las prácticas clave de CMM?" CROSSTALK, abril de 1998, págs. 18-23. Véase también www.ppc.com.

- Hall, David C., "Mejores prácticas: uso de un modelo de nivel de madurez de gestión de riesgos", Revista de calidad de software, vol. 2, No. 4, octubre de 2002, disponible en www.sqmmagazine.com/issues/2002-04/maturity.html.
- Harmon, Paul y Mark Watson, Comprendión de UML: la guía para desarrolladores, San Francisco, CA: Morgan Kaufman Publishers, Inc, 1997.
- Hay, David C., Patrones de modelos de datos: convenciones de pensamiento, Nueva York: Dorset House, 1996.
- Hay, David C., Análisis de requisitos: de la visión empresarial a la arquitectura, Upper Saddle River, Nueva Jersey: Prentice Hall PTR, 2003.
- Hay, John, Análisis de requisitos: desde la visión empresarial hasta la arquitectura, Englewood Cliffs, Nueva Jersey: Prentice Hall, 2002.
- Herbsleb, James, Anita Carlton, James Rozum, Jane Siegel y David Zubrow, Beneficios de la mejora de procesos de software basados en CMM: resultados iniciales, Informe técnico CMU/SEI-94-TR-013, Pittsburgh, PA: Instituto de Ingeniería de Software, agosto 1994.
- Higgins, Stewart A., et al, "Gestión de requisitos para productos de tecnología de información médica", IEEE Software 2003: 20(1), 26–33. Consulte www.computer.org/software.
- Hooks, Ivy F. y Kristin A. Farry, Productos centrados en el cliente: creación de productos exitosos a través de una gestión inteligente de requisitos, Nueva York: AMACOM (rama editorial de la American Management Association), 2001.
- Hooks, Ivy, "Escribir buenos requisitos: un tutorial de un día", patrocinado por el capítulo del Área Metropolitana de Washington (WMA) del Consejo Internacional de Ingeniería de Sistemas (INCOSE), McLean, VA: Compliance Automation, Inc., junio de 1997.
- Humphrey, Watts S., Introducción al proceso de software personal, Reading, MA: Addison-Wesley, 1997.
- Humphrey, Watts S., Introducción al proceso de software en equipo, Reading, MA: Addison-Wesley, 2000.
- Humphrey, WS, et al, "Mejora de procesos de software en Hughes Aircraft", Software IEEE, agosto de 1991, págs. 11-23.
- Inmon, WH, John A. Zachman y Jonathan C. Geiger, Almacenes de datos, almacenamiento de datos y el marco Zachman: gestión del conocimiento empresarial, Nueva York: McGraw Hill, 1997.
- Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), Conferencia IEEE sobre Ingeniería de Requisitos (ICRE) Sitio web: Conferences.computer.org/RE/.
- Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), IEEE 1220, Guía IEEE para Tecnología de la Información-Definición de Sistemas-Concepto de Operaciones (ConOps) Documento, IEEE, Nueva York, 1998.

Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), IEEE 1320.1, Estándar IEEE para lenguaje de modelado funcional: sintaxis y semántica para IDEF0, IEEE Computer Society, 1998.

Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), J-STD-016-1995, Estándar para procesos de ciclo de vida de software de tecnología de la información Acuerdo entre adquirente y proveedor de desarrollo de software (emitido para uso en senderos), Nueva York: Instituto de Ingenieros Eléctricos y Electrónicos, Inc., 1995.

Comité de Normas de Ingeniería de Software del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), IEEE STD 830-1998, Práctica recomendada por IEEE para especificaciones de requisitos de software, IEEE Computer Society, 25 de junio de 1998.

Comité de Normas de Ingeniería de Software del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), IEEE Std 1233a-1998, Guía IEEE para el desarrollo de especificaciones de requisitos de software, IEEE Computer Society, 8 de diciembre de 1998.

Estándar 12207 del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), Procesos del ciclo de vida del software, Nueva York: IEEE, 1998.

Sitio web de la Asociación Internacional de Facilitadores: www.iaf-world.org/.

Consejo Internacional de Ingeniería de Sistemas (INCOSE), INCOSE INSIGHT (Revista de Ingeniería de Sistemas) Sitio web: www.incose.org/insight.html.

Consejo Internacional de Ingeniería de Sistemas (INCOSE), sitio web de la organización nacional INCOSE: www.incose.org/se-int/.

Consejo Internacional de Ingeniería de Sistemas (INCOSE), Capítulo del Área Metropolitana de Washington (WMA) de INCOSE Sitio web: www.incose-wma.org/info/.

Sitio web del Grupo Internacional de Usuarios de Puntos de Función (IFPUG): www.ifpug.org.

Jackson, Michael, Marcos de problemas: análisis y estructuración de problemas de desarrollo de software, Londres, Reino Unido: Addison-Wesley, 2001.

Jones, Capers, Evaluación y control de riesgos de software, Englewood Cliffs, Nueva Jersey: Prentice Hall, 1994.

Jones, Capers, Estimación de costos de software, Nueva York: McGraw Hill, 1998.

Jones, Capers, "Factores positivos y negativos que influyen en la productividad del software", Burlington, MA: Software Productivity Research, Inc., versión 2.0, 15 de octubre de 1998.

Jones, Capers, Evaluaciones de software, puntos de referencia y mejores prácticas, Reading, MA: Addison-Wesley, 2000.

Jones, Capers, "Gestión de proyectos de software en el siglo XXI", American Programmer, vol. 11, No. 2, febrero de 1998. También disponible en spr.com/news/articles.htm.

Jones, Capers, Calidad del software: análisis y directrices para el éxito, Londres: International Thomson Computer Press, 1997.

Jones, Capers, Calidad del software en 2000: qué funciona y qué no, 18 de enero de 2000.

Jones Capers, "Qué significa ser 'el mejor en su clase' en software", Burlington, MA: Software Productivity Research (SPR), Inc., versión 5, 10 de febrero de 1998.

Korson, Timothy, "El uso indebido de los casos de uso: gestión de requisitos", Disponible en www.korson-mcgregor.com/publications/korson/Korson9803om.htm.

Kotonya, Gerald e Ian Sommerville, Ingeniería de requisitos: procesos y técnicas, Chichester, Reino Unido: John Wiley & Sons, 1998.

Kulak, Daryl y Eamonn Guiney, Casos de uso: requisitos en contexto, Nueva York: ACM Press, 2000.

Leffingwell, Dean y Don Widrig, Gestión de requisitos de software: un enfoque unificado, Reading, MA: Addison-Wesley, 2000.

Markert, Charles, Partnering: Unleashing the Power of Teamwork, 2002, informe informativo disponible en Markert@facilitationcenter.com.

McConnell, Steve, Guía de supervivencia de proyectos de software, Redmond, Washington: Microsoft Press, 1998.

McGibbon, T., "Un caso de negocio para la mejora de procesos de software revisado: medición del retorno de la inversión a partir de la ingeniería y la gestión de software", número de contrato SP0700-98-4000, Centro de análisis y datos para software (DACS), ITT Industries, ingeniería avanzada y División de Ciencias, Roma, NY, 30 de septiembre de 1999. Disponible en www.dacs.dtic.mil/techs/roispi2/.

McKinney, Dorothy, "Seis traducciones entre el lenguaje del software y el lenguaje de la gestión", IEEE Software 2002: 19(6) 50–52. Ver www.computer.org/software.

Comunicado de prensa de Northrop Grumman Information Technology Defense Enterprise Solutions sobre CMMI Nivel 5. Consulte www.irconnect.com/noc/pages/news_releases.mhtml?d=35405.

Northrop Grumman Information Technology Defense Enterprise Solutions, "The Road to CMM(I) Level 3", documento técnico disponible en Ralph.joven@ngc.com.

Grupo de gestión de objetos (OMG), Introducción a UML. Ver www.omg.org/gettingstarted/what_is.uml.htm.

Object Management Group (OMG), Especificación del lenguaje de modelado unificado OMG, versión 1.4, septiembre de 2001 (566 páginas). Disponible en www.omg.org/technology/documents/formal/mof.htm.

Palmer, James D., "Trazabilidad", Ingeniería de requisitos de software, RH Thayer y M. Dorfman (eds.), 1997, págs. 364–374.

Paulk, MC, "Uso del software CMM con buen criterio", ASQ Software Quality Professional vol. 1, núm. 3, junio de 1999, págs. 19–29. Disponible en www.sei.cmu.edu/publications/articles/paulk/judgment.html.

Paulk, Mark C., "Using the Software CMM with Good Judgment: Small Projects & Small Organizations", Presentación en el Capítulo de Washington, DC, Mesa redonda de la Sociedad para la Calidad del Software (SSQ) 1998, 26 de enero de 1998.

Paulk, Mark C., Bill Curtis, Mary Beth Chrissis y Charles V. Weber, Capability Maturity Model for Software, versión 1.1, febrero de 1993, Software Engineering Institute (SEI), Carnegie-Mellon University, Pittsburgh, PA, 1993. Véase www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.HTML.

Porter-Roth, Bud, Solicitud de propuesta: una guía para el desarrollo eficaz de RFP, Boston, MA: Addison-Wesley, 2002.

Grupo de especialistas en ingeniería de requisitos (RESG) (en el Reino Unido) Sitio web: www.resg.org.uk/.

Robertson, Suzanne y James Robertson, Dominar el proceso de requisitos, Harlow, Inglaterra: Addison-Wesley, 1999.

Rogers, Everett M., Difusión de innovaciones (4^a ed.), Nueva York: The Free Press, 1995.

Ross, Jeanne W. y Peter Weill, "Seis decisiones de TI que su personal de TI no debería tomar", Harvard Business Review, noviembre de 2002, págs. 85–91.

Sabourin, sitio web de Rob, www.amibug.com/index.shtml.

Scholtes, P., B. Joiner y B. Streibel, The Team Handbook (2.^a ed.), Madison, WI: Oriel Inc., 2001.

Scott y Jaffee, Empoderamiento: una guía práctica para el éxito.

Sharp, Helen y otros, "Identificación de partes interesadas en el proceso de ingeniería de requisitos", IEEE, 1999, págs. 387–391.

Six Sigma Qualtec, Historia de QI: Herramientas y técnicas, Guía para la resolución de problemas, tercera edición, 1999. Llame al 480-586-2600 para obtener información. Véase también www.ssqi.es/homepage.asp.

Smith, Preston G. y Donald G. Reinertsen, Desarrollo de productos en la mitad del tiempo (2.^a ed.), Nueva York: John Wiley & Sons, Inc., 1998.

Sitio web de la Sociedad para la Calidad del Software: www.ssq.org/.

Revista de desarrollo de software, sitio web: www.sdmagazine.com/.

Instituto de Ingeniería de Software (SEI), Identificación de riesgos basada en taxonomía, Informe técnico CMU/SEI-93-TR-6, Pittsburgh, PA: SEI, junio de 1993. Véase www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf.

Sitio web de Software Productivity Research, Inc., www.spr.com.

Sommerville, Ian, Ingeniería de software (6^a ed.), Reading, MA: Addison-Wesley, 2001.

Sommerville, Ian y Pete Sawyer, Ingeniería de requisitos: una guía de buenas prácticas, Nueva York: John Wiley & Sons, 1997.

Sommerville, I., P. Sawyer y S. Viller, "Viewpoints for Requisitos Elicitation: A Practical Approach", Actas de la Conferencia Internacional sobre Ingeniería de Requisitos de 1998 (ICRE '98), 6 al 10 de abril de 1998, Colorado Springs, CO, Nueva York: IEEE Computer Society, 1998, págs. 74–81. Consulte computer.org/proceedings/icre/8356/8356toc.htm.

Sorensen, Reed, Comparación de metodologías de desarrollo de software, Centro de soporte de tecnología de software, enero de 1995. Disponible en www.stsc.hill.af.mil/crosstalk/1995/jan/comparis.asp.

Thayer, Richard H. y Merlin Dorfman (eds.), Ingeniería de requisitos de software (2.^a ed. revisada), Los Alamitos, CA: IEEE Computer Society Press, 2000.

Thayer, Richard H. y Mildred C. Thayer, "Glosario de ingeniería de requisitos de software", Ingeniería de requisitos de software (2^a ed.), Richard H.

Thayer y Merlin Dorfman (eds.), Los Alamitos, CA: IEEE Computer Society Press, 1997.

The Standish Group International, Inc., CHAOS Chronicles 2003Report, West Yarmouth, MA: The Standish Group International, Inc., 2002. Véase www.standishgroup.com.

The Standish Group International, Inc, ¿cuáles son sus requisitos? 2003, West Yarmouth, MA: The Standish Group International, Inc., 2002.

Página web del equipo de productos integrados del Cuerpo de Ingenieros del Ejército de EE. UU. Véase www.usace.army.mil/ci/lcmis/lcmipt.html.

Walton, Mary, El método de gestión de Deming, Nueva York: The Putnam Publishing Group, 1986.

Watts, FB, Manual de control de documentos de ingeniería: gestión de la configuración en la industria (2^a ed.), Park Ridge, Nueva Jersey: Noyes Publications, 2000.

Waugh, Penny, materiales de capacitación para participantes en la revisión por pares y moderadores de la revisión por pares, Northrop Grumman Information Technology Defense Enterprise Solutions, 2002. Contáctela en PWaugh@ngc.com.

Webster Bruce F., Errores del desarrollo orientado a objetos, M&T Books, 1995.

Weinberg, Gerald M., "¡Simplemente diga no! Mejorando el proceso de requisitos", Programador estadounidense (10) 1995:19–23.

Whitten, Neal, "Cumplir con los requisitos mínimos: cualquier cosa más es demasiado", PM Network, septiembre de 1998.

Wiegert, Karl E., sitio web, www.processimpact.com/.

Wiegert, Karl E., Requisitos de software (2^a ed.), Redmond, WA: Microsoft Press, 2003.

Wiegers, Karl E., "¿Funcionan sus inspecciones?" StickyMinds.com, 24 de junio de 2002. Véase www.stickyminds.com.

Wiegers, Karl E., Revisiones por pares en software: una guía práctica, Boston, MA: Addison-Wesley, 2002.

Wiegers, Karl E., "Requisitos de inspección", StickyMinds.com, 30 de julio de 2001. Consulte www.processimpact.com/.

Wiegers, Karl E., "Hábitos de los analistas eficaces", Revista de desarrollo de software, vol. 8, núm. 10 (octubre de 2000), págs.

Wiegers, Karl, "10 Requisitos Trampas a Evitar", Revista de Ingeniería de Calidad y Pruebas de Software, enero/febrero de 2000. Consulte www.stqemagazine.com/featured.asp?id=8.

Wiegers, Karl E., "Lo primero es lo primero: priorizar los requisitos", Revista de desarrollo de software, vol. 7, núm. 9, septiembre de 1999, págs. 24–30.

Wiegers, Karl E., "Automatización de la gestión de requisitos", Desarrollo de software, julio de 1999.

Wiegers, Karl E., Creación de una cultura de ingeniería de software, Nueva York: Dorset House Publishing, 1996.

Wiley, Bill, Requisitos esenciales del sistema: una guía práctica para métodos basados en eventos, Reading, MA: Addison-Wesley, 2000.

Wood, Jane y Denise Silver, Desarrollo conjunto de aplicaciones, Nueva York: John Wiley & Sons, 1995.

Young Ralph R., "Resumen de requisitos iniciales del proyecto", sitio web: www.ralphyoung.net.

Young, Ralph R., Prácticas de requisitos eficaces, Boston, MA: Addison-Wesley, 2001.

Young, Ralph R., "Prácticas recomendadas de recopilación de requisitos", CrossTalk, 15(4), abril de 2002, págs. 9-12.

Young, Ralph R., Plantilla de plan de requisitos y plan de requisitos de muestra. Véase www.ralphyoung.net.

Young, Ralph R., "Estudio sobre el comercio de herramientas y requisitos", disponible en www.ralphyoung.net/publications/Requirements_Tools_Trade_Study1.doc.

Young, Ralph R, "La importancia y el valor de la mejora de procesos". Disponible en www.ralphyoung.net.

Young, Ralph, "El valor de un grupo de trabajo organizacional". Disponible en www.ralphyoung.net.

Sitios web de Zachman Framework, por ejemplo, consulte www.zifa.com/.

Sobre el Autor

El Dr. Ralph R. Young es un líder activo y colaborador en sistemas, software e ingeniería de procesos. Su principal interés es aportar un sólido conocimiento práctico de las mejores prácticas a una amplia comunidad profesional y académica. En esta búsqueda, imparte cursos y talleres de ingeniería de procesos y requisitos, es un orador frecuente en reuniones y conferencias y mantiene contacto regular con expertos de la industria. Ha recibido premios por trabajo en equipo, liderazgo, mejora continua y publicaciones y, a menudo, se le reconoce por sus contribuciones en la gestión y mejora de procesos. Ha creado una serie de siete textos que cubre todo el espectro de cómo lograr su visión de la ingeniería de requisitos que se describe en el Capítulo 9. Este es el segundo libro de la serie; su primer libro, *Prácticas efectivas de requisitos* (Addison-Wesley, 2001), encontró una audiencia receptiva con RA en informática e ingeniería, desarrolladores y gerentes, y generó una serie de solicitudes de presentaciones de charlas y talleres y otras oportunidades de escritura.

El Dr. Young es director de mejora de procesos de ingeniería, sistemas e ingeniería de procesos, Defense Enterprise Solutions (DES), en Northrop Grumman Information Technology, un proveedor líder de soluciones basadas en sistemas. El Dr. Young ayudó a llevar su antigua unidad de negocios (Litton PRC) al Nivel 5 de CMMI y su unidad de negocios actual (DES) al Nivel 5 de CMMI. Apoya proyectos internos y externos para mejorar sus capacidades para utilizar técnicas de mejora de procesos, implementar requisitos prácticos y desarrollar innovaciones para facilitar la gestión de proyectos. Dirige un grupo de trabajo de requisitos que involucra a más de 50 ingenieros de requisitos de proyectos en su unidad de negocios.

El Dr. Young se graduó de la Universidad de New Hampshire y obtuvo una maestría en economía y un doctorado en administración de empresas en la Universidad George Washington en Washington, DC. Ha estado involucrado en actividades de desarrollo de sistemas y software durante más de 35 años. En 1972, fue nombrado director de la División de Desarrollo de Sistemas del condado de Fairfax, Virginia, donde un grupo de 45 desarrolladores altamente calificados proporcionaron sistemas de última generación para funciones del gobierno local. Posteriormente, participó y gestionó diversos sistemas y actividades de soft

Martin Marietta Corporation, TRW, PRC, Inc., Litton PRC y Northrop Grumman Information Technology.

Él y su esposa, Judy, han estado casados durante 37 años. Judy es ejecutiva de una asociación y líder en deportes y actividad física, por lo que Ralph sale a caminar temprano todos los días. Ralph disfruta de las actividades familiares con hijos y nietos, la música, el canto, la naturaleza, el aire libre y la naturaleza. Una prioridad en su vida es la participación activa en las comunidades de fe de las iglesias locales. Después de jubilarse, Judy y Ralph sueñan con vivir a bordo de un barco pesquero y viajar mucho.

Índice

A

Afors, Cristina, 98, 106
 Después de la fiebre del oro: creación de una verdadera profesión de ingeniería de software, 199 Metodologías de desarrollo ágiles, 154, 155 Aldridge, EC Jr., 73 Alexander, Ian, 35, 44, 56, 60, 65, 73 , 95, 96, 104, 105, 166 Sociedad Estadounidense para la Calidad (ASQ), 37 Patrones de análisis, 20 Analizar soluciones candidatas (ACS), 88 Enfoques, 121 Herramientas de requisitos automatizadas, 210 adquirir, 91–92 experiencias, 87 fuerza industrial, usar, 118–19 lista de, 87 preguntas para, 87–88 seleccionar, 86–91

B

Bases de estimaciones (BOE), 150 Requisitos de comportamiento. Consulte Requisitos funcionales Mejores prácticas, 103, 109–26, 207 objetivo, propósito y misión acordados, 121 uso de herramientas de requisitos automatizados, 118–19 categorías, 110 control de cambios, 117– 18 participación del cliente/usuario, 115 inspecciones de documentos, 117 utilización de expertos en el dominio/ PYMES, 116 directrices efectivas para reuniones/envío de correo electrónico, 122 técnicas efectivas de recopilación de requisitos, 115 chapado en oro y, 115 clima de mejora, establecimiento, 123 lista, para el desarrollo y gestión de requisitos, 110, 111–12

desarrollo, implementación y cumplimiento de reglas de reuniones, 121–22 identificación de requisitos mínimos, 116–17 identificación de objetivos, documentación, acuerdo, 113 políticas de requisitos de organización/proyecto, 119– 20 priorización de requisitos, 117 uso del glosario/lista de acrónimos del proyecto, 115 mecanismos, enfoques, métodos, técnicas y herramientas probados, 120– 21 identificación de requisitos reales, 114 documentación de la justificación de los requisitos, 114 requisitos, 109 decisiones de requisitos, 115 iteración de requisitos, 115–16 requisitos desarrollo del plan, 112 capacitación de requisitos, 114 talleres de requisitos, 113 evaluación de riesgos, 122 cuantificación del ROI, 116 identificación/ participación de las partes interesadas, 113 resumen, 123 gestión de equipos, 122–23 versiones/ lanzamientos uso de productos de trabajo, 118 redacción de requisitos, 112–13 Véase también Analistas de requisitos (RA) Bicknell, BA, 153, 166 Bicknell, KD, 153, 166 Trazabilidad bidireccional, 210 Blitz QFD, 153 Boehm, Barry, 23, 73, 105, 154, 159, 166, 167 BPwin, 97 Brodman, JG, 140, 156, 166 Buede, Dennis, 56, 60, 96, 97, 106 Requisitos de construcción, 48 Impulsores comerciales, 170 Requisitos comerciales, 49–50

- Reglas comerciales, 50–51
- líneas de base,
 - 51 documentación, 51
 - como base de requisitos funcionales, 50 lista de, 50–51 Butler, K., 156, 166
- C Caliber RM, 88, 186
- Áreas de mejora candidatas, 209–12 Modelo de madurez de capacidad (CMMI), 6, 139, 174, 193 como ortogonal al éxito del proyecto, 156 medición de madurez de proceso, 156 Integración del modelo de madurez de capacidad (CMMI), 6, 139, 193
 - Análisis y resolución casual (CAR), 145 cliente, producto, requisitos de componentes del producto, 48 análisis de requisitos, 48 sitio web, 13, 60 Carr, Frank, 96, 196 Carroll, Pete, 93, 95 Estudios de casos del cliente priorización de requisitos, 104–5 proceso de RM efectivo, 213–
 - 15 contratista de la junta federal, 163–64 discusiones de PM, 12 fracaso del proyecto, 24–25 mejora de la calidad desviada, 189–91 gerente senior, 43 caso legal de integración de sistemas, 123–26 Web finalización del sitio, 200–202 Rendimiento del sitio web, 57–60 Control de cambios, 18–19 tablero (CCB), 67 mecanismo, establecimiento, 83–85, 209 ilustración de procedimientos, 133 calidad del producto y, 85 Solicitudes de cambio (CR), 132 Cambio(s) después de la línea base de requisitos, 118 compromiso, 209 gestión, 198 operativo, gestión, 134 costos del proyecto y, 118 evaluación de riesgos, 122 supresión, 85 Véase también Control de cambios Informe CHAOS, 12, 55 Clark, BK, 7, 156, 166, 191 Cockburn, Alistair, 96, 136, 165 Inspecciones de código, 103 Colaboración, 15–18, 114 Mejora de las comunicaciones, 211
- Requisitos de los componentes, 55 Concepto de operaciones (CONOPS), 65 Auditorías de configuración, 134 Control de configuración, 132–33 Tablero de control de configuración (CCB), 115, 117, 133
- definido, 133 para revisión, 134
- Gestión de configuración (CM), 129–35
- herramientas automatizadas, 89 definidas, 129 métricas, 134–35 plan, 3, 70, 131 políticas, 131 RA y, 129–35 Coordinación RM, 131 Pymes, 89 técnicas, 83 herramientas, 131
- Contabilidad del estado de configuración, 134
- NÚCLEO, 186
- Contramedidas, 149
- Diáfonía, 34, 96
- Productos centrados en el cliente: crear productos exitosos
- Productos a través de requisitos inteligentes Gestión, 97, 127
- Colaboración
- con clientes, 114 estrategia de participación, 67–69 composición, 67 preferencias, 89 requisitos, 55 encuestas de satisfacción, 175–76 soporte, 198 hablar con, 90
- D**
- Patrones de modelos de datos: convenciones de pensamiento, 20 Daughtrey, T., 151, 152, 166 Davis, Alan, 5, 54, 60, 198, 203
- Resolución de análisis de decisiones (DAR), 19, 88
- Prevención de defectos (DP), 145–49
- análisis, 146
 - definido, 145
 - proceso, 145–46
 - ilustración de proceso, 146
 - talleres, 146
- Detección
- de defectos, 150
 - principales, 150 eliminación, 84 Deming, W. Edwards, 39, 43, 177, 194, 202 Requisitos derivados, 52 Restricciones de diseño , 52

- inspecciones, 103
 procesos, 77, 176, 180–87, 210
 requisitos, 48, 52 riesgos, 161
 Diseñadores, que involucran, 98 Patrones de diseño, 20 Expectativas de los desarrolladores, 199 soporte, 198–99 Desarrollo de productos en la mitad del tiempo, 153 Riesgos de desarrollo , 162–63 Difusión de innovaciones, 19 Dion, R., 156, 167 Doing PDCA, 39, 211 Expertos en el dominio, 116 Sistema dinámico de requisitos orientado a objetos (DOORS), 55, 88, 119, 186 atributos, 92, 94 módulos , 95 atributos del sistema utilizan conocimientos, 94–95 Consulte también Herramientas de requisitos automatizados
- mi
 Eckes, G., 202 Prácticas de requisitos eficaces, 3, 17, 20, 21, 67, 70, 71, 78, 86, 97, 98, 103, 116, 120, 122, 183 Asociación de Industrias Electrónicas (EIA) Estándar, 649, 165 Norma 632, 48 Encuestas de satisfacción de empleados, 176 Líneas base de ingeniería, 131 Propuestas de cambios de ingeniería (ECP), 132 Manual de Control de Documentación de Ingeniería: Gestión de configuración para la industria, 134 Mejora de procesos de ingeniería Colaboración (EPIC), 60 Entorno de software de ingeniería (ESE), 92 Riesgos de especialidades de ingeniería, 162 El diseño de ingeniería de sistemas: modelos y Métodos, 96, 97 RA de nivel de entrada, 30 Requisitos medioambientales, 55 Requisitos esenciales del sistema: una guía práctica para Métodos basados en eventos, 97 Modelo evolutivo comparación, 74–75 ilustrada, 76
- F Facilitación, 144–45 mediación, 22–23 diseño de procesos, 183 habilidades, 37–38 Farncombe, A., 105 Farry, KA, 13, 85, 97, 106, 164, 203 Feldmann, CG, 139, 165 FIPS PUB 183, 139, 165 Enfoque, 40–41, 42 Fowler, Martín, 20, 26, 101, 135, 165 Documento funcional (FD), 51–52 Requisitos funcionales, 51–52 base de reglas comerciales, 50 definido, 51 Análisis de puntos de función (FPA), 140 Puntos de función (FP), 151 Conceptos Fundamentales para la Calidad del Software Ingeniero, 151, 152
- G Gaffney, Steven, 37, 44 Gamma, Eric, 20, 26, 165 Garmus, David, 151, 166 Gartner Group, 194 Geiger, JC, 60 Gilb, Tom, 104, 142, 143, 150, 151, 166 Chapado en oro , 41, 115, 199 Gottesdiener, Ellen, 37, 44, 46, 51, 67, 70, 79, 96, 98, 105, 113, 135, 165 Grady, Jeffrey O., 46, 53, 60, 85, 97 , 104, 106, 153, 166 Ruta de crecimiento, 21 Una guía para la gestión de la configuración de software, 134–35
- Principios rectores, 171–72 Guiney, E., 136, 165
- h Hadden, R., 139, 166 Requisitos de hardware, 46 restricciones, 46 rendimiento, 46 ver ejemplos, 58 Consulte también Requisitos de software Harmon, Paul, 97 Harroff, Noel, 167 Hay, David, 18, 20, 25, 26 Hay, John , 48, 60 Herbsleb, J., 156, 167 Herron, David, 151, 166 Requisitos de alto nivel, 50 centrándose en, 82 reescritura, 81–82
- Revisión de información histórica, 64 Hooks, IF, 13, 26, 79, 85, 97, 104, 106, 114, 127, 164, 198, 203 Humphrey, Watts, 39, 85, 97, 150, 152, 166, 167, 199, 205, 206, 215

I

Conferencia IEEE sobre ingeniería de requisitos, 35, 44
Software

IEEE, 34 IIties, 53–
54 Estimación

de impacto (IE), 142–43 definido, 142
posibilidades

de error, 143 usos, 142–43

Áreas de mejora,

209–12 Comparación de modelos

de desarrollo incremental, 74–75 ilustrados,
76 Inmon, WH, 60

INSIGHT, 34

Código de inspecciones,

103 diseño, 103

en desarrollo,
150

documentos

relacionados con requisitos,

117–18, 150–52

Instituto de Ingenieros Eléctricos y Electrónicos
(IEEE)

Conferencia, 35

Estándar 830, 165

Estándar 1233, 136, 165

Norma 1471, 137

Norma 12207, 48

Ver también Normas

Requisitos de apoyo logístico integrado (ILS), 54

Enfoque de equipo de producto integrado (IPT), 198

Enfoque de calidad integrado, 169–91, 207

componentes, 172–73

ilustración de componentes, 173

QI y, 173–79

resumen, 189

trabajo en equipo y, 187–89

Ver también Calidad

Definición de integración para modelado de funciones
(IDEF), 139, 140

Riesgos de integración, 161–62

Requisitos de interfaz, 53

Asociación Internacional de Facilitadores, 37, 44

Consejo Internacional de Ingeniería de Sistemas
(INCOSO), 37, 44

Organización Internacional de Normalización (ISO), 169

Introducción al proceso de software personal, 97, 150,
205

Introducción al proceso de software de equipo, 150

j

Jackson, Michael, 20, 26 John

Boardman Associates (JBA), 65 Johnson, DL,
140, 156, 166

Desarrollo conjunto de aplicaciones, 97

Equipos conjuntos, 8, 78, 114, 198

Jones, Alcaparras, 84, 85, 103, 106, 107, 150, 151, 153, 166

RA junior, 30

k

Requisitos clave, 56 PLAN

de conocimiento, 150 Korson,

T., 82, 83, 106, 136, 165 Kotonya, Gerald,
97 Kulak, D., 136, 165

l

Leffingwell, decano, 18, 25, 79, 97, 101, 106

León, Alexis, 134, 165

Niveles de abstracción, 82

Actividades

del ciclo de

vida, 16 decisión de enfoque,
73–77 comparación de modelos, 74–75

Necesidades de apoyo logístico, 54

LOGOS CMM a medida para pequeñas empresas, pequeñas
Organizaciones y pequeños proyectos, 140

METRO

Gestión de requisitos de software: un enfoque
unificado, 18, 97

Markert, C., 13, 69, 104

Dominio del proceso de requisitos, 7

McConnell, Steve, 97, 199

McGibbon, T., 156, 167, 203

McKinney, Dorothy, 38, 44

Mecanismos, 120–21

Facilitación de la mediación, 22–23

Direcrices

para reuniones, desarrollo/aplicación, 122

requisitos mínimos, 116 reglas, 121–

22 Métodos, 121

Métricas

CM, 134–35

monitoreo del desempeño a través de, 177 rol de

RA, 22 uso,

22 Michaels,

MZ, 106 RA de nivel

medio, 30 El uso

indebidamente de los casos de uso, 136

norte

Necesidades

definidas, 45 cumplimiento de requisitos mínimos, 116

Requisitos negociables, 55

- Requisitos no funcionales, 52
 Requisitos no negociables, 55
- oh
- Arquitectura de gestión de objetos (OMA), 136
 Grupo de gestión de objetos (OMG), 135, 165
 Desarrollo orientado a objetos (OO), 22
 Gestión del cambio operativo, 134
 Líneas base de operaciones, 131
 Políticas organizativas, 64–65, 119
 Requisitos de origen, 56
- PAG
- Palmer, James D., 98, 106 Análisis de Pareto, 148 Diagramas de Pareto, 148
- Asociaciones, 68–69
 costos, 68
 definidos, 68
 iniciación, 210
 elementos de éxito, 69
- Colaboración en la construcción: una guía práctica para Éxito del proyecto, 96, 196
- Paulk, MC, 13, 139, 153, 165, 202 PDCA
- ciclo, 179
 haciendo, 39, 211
 realizando, 212
- Revisiones por pares,
 150 iniciando, 210
 proceso, 71
- Revisiones por pares en software: una guía práctica, 71, 97
- Requisitos de desempeño, 46, 53
- Proceso de software personal (PSP), 206
- Políticas, 119–20
 CM, 131
 definición/documentación, 210
 organizacional, 64–65, 119
 requisitos, 210
- Porter-Roth, B., 105 Guía práctica para la reingeniería de procesos de negocio utilizando IDEFO, 139
- Conocimientos prácticos, 154
- Metodología PRINCE2, 158
- Priorización de requisitos, 4–5, 104–5, 117, 198, 209
- Gestión de prioridades, 172
- Proceso
 enfoque, 6–7
 ingeniería, 183
 indicadores, 185
 gestión, 176–77 propietario, 183 requisitos, 54
 Descripción del proceso (PD), 6
- plantilla, 183, 184–85
 escritura, 183
- Diseño de procesos, 176–77, 180–87, 210
 actividades, 180
 diagrama de flujo completo, 183
 ejercicio, 182–83
 facilitación, 183
 símbolos de diagrama de flujo, 182 plantilla de diagrama de flujo, 181 proceso, 180–82 taller, 183
- Mejora de procesos, 176–77, 210 clima, 123
 plan (PIP), 190
- PM y, 155 apoyo, 154–56
- Reutilización de procesos, 19–20
 definido, 19
 ejemplo, 20
 recursos, 20
- Estructura de desglose del producto (PBS), 157–59 ventaja, 158 como
 herramienta de planificación de proyectos, 159
- Requisitos del producto, 54
- Plan de gestión del proyecto (PMP), 70
- Gerentes de proyecto (PM), 2
 desafíos, 196
 enfoque, 195, 196
 reuniones con, 10–11
 mejora de procesos y, 155 calidad y 195
 soporte, 197–98
- Actividades de
 gestión de riesgos del proyecto, 84
 procesos, 42
- Lista de
 acrónimos de proyectos, 72–73
 115 glosario, 72–73, 115
 objetivos, 113
 O&M, plan
 131, 3
 I+D, 86
 pequeños,
 11 “pequeños”,
 206 documentos de visión y alcance, 69–70
- Creación de prototipos, 102–3
- q
- Requisitos de calificación, 53
- Aplicación
 de la calidad, 152
 atributos, 53–54
 impulsores de negocio para, 170
 compromiso con, 170
 principios rectores para, 171–72

- Indicadores de calidad
 - (continuación), 185
 - enfoque integrado, 169–91 rol de
 - gestión en, 170–71
 - PM y, 195
 - Ver también Enfoque de calidad integrado
- Aseguramiento de la calidad (QA),
 - 176 plan, 3,
 - 70 proactivo, 152
 - revisiones, 176
 - especialistas, 153
- Despliegue de la función de calidad (QFD), 153
- Encuesta de satisfacción
 - del cliente de mejora de la calidad, ciclo 175–76, 174
 - encuestas de satisfacción de los empleados, 176
- Ciclo PDCA, 179
 - seguimiento del desempeño, 177
 - diseño, gestión, mejora de procesos, 176–77
 - modelos de mejora de procesos y, 174–75
 - Control de calidad, 176
 - historias, 177–79 equipos, 175 técnicas, 169, 173–79, 177 capacitación, 175
- Junta de gestión de calidad (QMB), 174
- Gestión cuantitativa (QM), 22

- R**
- Documentación fundamentada, 83, 114, 198, 209.
- Rosa racional, 97
- Conjunto de herramientas racionales, 119
- Proceso Unificado Racional (RUP), 119
- Lectura, 209
- Requisitos reales, 56
 - definidos, 2
 - en desarrollo, 83
 - en evolución, a partir de los requisitos establecidos, 78–80, 84
 - mecanismo de identificación, 209
 - identificación, 2, 114
 - carga, 92–95
 - requisitos establecidos vs., 50
 - tutoriales, 46–47 Ver también Requisitos Trabajo
 - real, 2 Estrategia
- recomendada, 9–10 Referencias, 9
- Reinertsen, DG, 153, 166 Requisitos actividades, 3–5 asignación, 5
 - análisis, 4
 - identificación
 - de atributos, 88
- matriz de atributos, 93
- mejores prácticas, 103, 109–26
- construcción, 48 negocios, 49–50
- clarificación, 4
- colaborativo, 17
- componente, 55
- cliente, 55
- decisiones, no tomar, 115 definido, 1–2, 45 definición, 4
- definición del proceso de requisitos, 9 derivados, 5, 52
- diseño, 48, 52
 - ejemplos de contexto detallados, 58 técnicas de obtención, 17–18, 96 ambientales, 55 evolución de, 7–8
 - funcionales, 50, 51–52 buenos, criterios de, 8
 - hardware, 46 ejemplos de vista de hardware/software, 58 de alto nivel, 50, 81–82
 - identificación, 4, 101–2
- ILS, 54
- importancia de, 1–12
- interfaz, 53
- iteración, 115–16
- clave, 56
- conocible, 205–13
- etiquetado, 92
- fuga, 70 apoyo
- logístico, 54 gestión, 5
- mandala, 212–13 mínimo, 116–17
- negociable, 55 no
- funcional, 52 no
- negociable, 55
- originario, 56
- particionamiento, 5 desempeño, 46, 53 políticas, 210
- priorización, 4–5, 104–5, 117, 198, 209 proceso, 54 producto, 54 calificación, 53
- documentación justificativa, 83, 114, 198, 209
- Ejemplos de visualización de RA, 59 reales, 2
- reformulados, 4
- riesgos, 160–61 roles y responsabilidades, 8 software, 46, 81–82 fuente, 55

- ingeniería especializada, 53–54
especificación, 4, 142
declarado,
2 subsistema, 55
sistema, 50, 55
taxonomía, 47
ejemplos de taxonomía, 58–59
terminologías a evitar, 55–56 pruebas,
5
seguimiento,
5 capacitación,
114 tipos, RA vista, 49
tipos de ejemplos, 56–57
tipos de, 45–60
incognoscible, 2, 54
usuario,
50 validado, 5, 53
verificado, 5, 53
vistas de tipos, 45–48
volatilidad, 84
talleres, 67, 113
escritura, 112–13
Ver también Ingeniería de requisitos
Análisis de requisitos: desde vistas comerciales hasta
Arquitectura, 18, 48
Analistas de requisitos (RA), 2 objetivos
alcanzables, 42 actitud
de mejora continua, 39 mejores prácticas, 109–
26, 207 desafíos, 24 rol de control
de cambios, 18–
19 características, 34–42
características como
contramedidas, 35 lista de características, 36–
37
CM y, 129–35 rol
colaborativo, 15–18
comunicación con la gerencia, 38 educación
continua, 34–36 efectivo, 34–42
aplicación de
prácticas efectivas, 38–39 conocimiento
tecnológico en evolución, 41–42 rol de facilitación/
mediación, 22–23 habilidades de
facilitación/negociación, 37–38 como
facilitadores, 144–45
enfoque, 40–
41 rol en la ruta de
crecimiento, 21 junior (nivel
inicial), 30 liderazgo,
143–44 aprendizaje,
38–39 niveles
de, 29 actividades del ciclo
de vida, 16 como oyente, comunicador,
escritor, 37 marcando la
diferencia, 42
métricas del rol, 22 nivel medio, 30
rol de nueva tecnología, 19
persistencia/perseverancia, 38
proactivo, 38 rol
de reutilización de procesos,
19–20 aplicación de principios de calidad,
152 en recopilación de requisitos, 62
vista de tipos de requisitos, 49
estimación de recursos/tiempo, 39–40
responsabilidad de puntos de vista, actitudes,
relaciones, 39
contribución al proceso de riesgo,
42 roles, 15–23, 16, 206
nivel superior, 30–31
habilidades, 27–34,
206 matriz de habilidades,
28–29 habilidades especializadas,
127–64, 207 actividades de
fortalecimiento,
36–37 función de estudio, 23 mejora
de la tasa de éxito, 195 función de métodos/
técnicas de apoyo, 21–22
pensamiento innovador, 41 capacitación, 81
Descripción del puesto de analista de requisitos (RA),
32–34
Descripción, 32
Conocimiento necesario, 32
Medidas de desempeño, 33
Referencias, 34
Responsabilidades, 32–33
Habilidades necesarias, 32
Requisitos por Colaboración: Talleres para
Definición de necesidades, 96
Desafíos de ingeniería de
requisitos, 195–97
dificultad, 1, 194
objetivos,
193 visión, 193–202
Ingeniería de requisitos: una guía de buenas prácticas,
97
Ingeniería de Requisitos: Procesos y Técnicas,
97
Grupo de especialistas en ingeniería de requisitos
(RESG), 37, 44
Errores de requisitos, 127–29
definidos, 127
gastos, 127
frecuencia, 129
experiencia en la industria, 129
razones para, 212
reducción, 128–29
acciones de reducción, 130
tipos de, 129
Recopilación de requisitos, 61–105
solicitud, 114

- Recopilación de requisitos (continuación)
- adquisición de herramientas de requisitos automatizadas, 91–92
 - selección automatizada de herramientas de requisitos, 86–91
 - establecimiento del mecanismo de control de cambios, 83–85
 - lista de verificación, 63, 207 completar, 103–4
 - estrategia de participación del cliente/usuario, 67–69
 - técnicas efectivas, 115 uso efectivo de, 208–9 revisión de información histórica, 64 niveles de abstracción, 82 decisión de enfoque de ciclo de vida, 73–77 organización revisión de políticas, 64–65 desempeño, 96–98 prácticas, métodos, selección de técnicas, 86 principios, 82–83 glosario/lista de acrónimos del proyecto, 72–73 documento de visión/alcance del proyecto, 69–70 enfoque de prueba de concepto, 102–3 documentación fundamentada, 83 desarrollo de requisitos reales, 83 evolución de requisitos reales, 78–80 carga de requisitos reales, 92–95 incorporación de mejores prácticas de requisitos, 103 identificación de requisitos, plan de requisitos 101–2, 70–72 sesiones de capacitación relacionadas con los requisitos, 80–81 revisiones de requisitos, 98 escenarios, 99–101 reescritura de requisitos de software, 81–83 identificación de partes interesadas, 65–67 resumen, 204 adaptación, 77–78 desarrollo de estrategia de trazabilidad, 98–101 visibilidad, 212 Planificación de V&V, 85–86 Gestión de requisitos (RM), 40 mejores prácticas, 110, 111–12 Coordinación CM, 131 Planes de requisitos (RP), 7–10 antecedentes, 7 resumen del contrato/proyecto, 7 definido, 3 en desarrollo, 70–72, 112, 211 evolución de los requisitos, 7–8 propósito, 7 muestra de índice, 71 redacción, 3 Apéndices del proceso de requisitos, 10

en desarrollo, 120 ideas de mejora, 147–48 interactivas, 2 inversión en, 5–6 iterativas, 2 mecanismos, métodos, técnicas, herramientas, 9 integración de prácticas, 9 estrategia recomendada, 9–10 referencias, 9 requisitos, 3 adaptación, 77–78, 120

Composición del repositorio de requisitos de, 86 elementos, 91

Especificaciones de requisitos, 141–42 Matriz de trazabilidad de requisitos (RTM), 30 Grupo de trabajo de requisitos (RWG), 185–87 ventajas, 185–86 información, 187 definido, 185 evolución, 186 formación, 211 iniciación, 185

Requisito Pro, 88, 119, 186 Retorno de la inversión (ROI) inspección en desarrollo, 150 cuantificando, 116

Asistente de Gerente de Riesgos (RMA), 40 Análisis de riesgos, 160–63 evaluación, 122 diseño, 161 desarrollo, 162–63 especialidades de ingeniería, 162 integración, 161–62 manejable, 205–13 proceso, 42 requisitos, 160–61

La hoja de ruta hacia el éxito repetible: utilizar QFD para Implementar el cambio, 153

Robertson, J., 13, 165 Robertson, S., 13, 165 Rogers, Everett M., 19, 26 Ross, JW, 44 RTM Workshop, 88, 186 Reglas de conducta, 211

S Sabourin, Rob, 151–52, 166 sistema SATURNO, 57 Sawyer, Pete, 97 Análisis de necesidades de los usuarios basado en escenarios (SUNA), 97, 99, 101

Escenarios definidos, 99

- en recopilación de requisitos, esquema de uso 99–101, 100–101
- Scholtes, PR, 203 RA
- de nivel superior, 30–31
- acuerdos de nivel de servicio (SLA), 134 Sharp, Helen, 66, 105 Shewhart, Walter, 39 Silver, Denise , 97, 106 Six Sigma, 193 SLIM, 150 Proyectos pequeños, 11 Smith, PG, 153, 166 Society for Software Quality (SSQ), 37 Plan de desarrollo de software (SDP), 3 tendencias globales, 138–39 QFD, 153 Desarrollo de software Revista, 34 Software Productivity Research (SPR), 149, 150 Guía de supervivencia de proyectos de software, 97 Calidad del software: análisis y pautas para el éxito, 85, 153 Requisitos de software, 46 reescritura, 81–82 ver ejemplos, 58 Consulte también Requisitos de hardware Requisitos de software : Objetos, funciones y estados, 54 Sommerville, Ian, 97, 159, 167 Sorensen, R., 105 Requisitos de origen, 55 Requisitos de ingeniería especializada, 53–54 Modelo en espiral, 7, 73–75 comparación, 74–75 ilustrado, 77 Identificación de partes interesadas, 4, 65–67, 113 participación, 113 objetivos del proyecto y, 113 comunicación de requisitos, 80 roles, 66 procedimientos operativos estándar (SOP), 131 estándares EIA 632, 48 EIA 649, 165 FIPS PUB 183, 139, 165 IEEE 830, 165 IEEE 1233, 136, 165 IEEE 1471, 137 IEEE 12207, 48 Grupo Standish Internacional, 194 Colaboración de requisitos declarados relacionados con, 114 definidos, 2 requisitos reales en evolución de, 78–80 requisitos reales vs., 50 Ver también Declaración de requisitos de trabajo (SOW), 2, 119 Control estadístico de procesos (SPC), 152 Stevens, R., 56, 60, 96 Expertos en la materia (PYME), 72, 116 Requisitos del subsistema, 55 Encuestas de satisfacción del cliente, 175–76 satisfacción de los empleados, 176 Arquitectos de sistemas, que involucran, 98 Requisitos del sistema, 50, 55 Análisis de requisitos del sistema, 97 Modelo de madurez de capacidad de ingeniería de sistemas (SE-CMM), 45, 88 Plan de gestión de ingeniería de sistemas (SEMP), 3, 70 Análisis de requisitos de sistemas, 53 Validación y verificación de sistemas, 97
- t**
- Sastrería, 19, 20 criterios, 89 como habilidad crítica, 78 definidos, 19 proceso de requisitos, 77–78, 120 Cuestionario basado en taxonomía (TBQ), 159 Capacitación para la formación de equipos, 210–11 equipos alto rendimiento, 187, 188 conjunto, 8, 78, 114, 198 gestión y, 188–89 gestión, 122–23 miembros, 187, 197 Proceso de software de equipo (TSP), 206 Trabajo en equipo, 187– 89 efectivo, 188 factores de evolución, 187–88 Entorno de TRABAJO EN EQUIPO, 72, 79, 84 Técnicas, 121 Tecnologías en evolución, conocimiento de, 41–42 nuevas, 19 Plan de prueba, 3 Herramientas, 121 requisitos automatizados, 86–92, 87, 118–19, 210 proceso de requisitos, 9 Gestión de Calidad Total (TQM), 169 Estrategia de trazabilidad, 98–101 Lecciones de estudios comerciales, 89–91 Capacitación, 199 plan, 70 provisión, 114, 210 como técnica de mejora de la calidad, 175

- Capacitación (continuación)
RA, 81
sesiones relacionadas con requisitos, 80–81
resistencia, 80
formación de equipos, 210–11
Turner, Ricardo, 154, 166
- Ud.
- UML destilado: aplicación del objeto estándar
Lenguaje de modelado, 135
- Comprensión de UML: la guía para desarrolladores, 97
- Aplicación Unified Modeling
Language, 136
definido, 135
diagramas, 136
experiencia en el uso, 137
Dios mío, 135
Conocimiento de RA, 135–39 en
trabajos relacionados con requisitos, 136–37
- Requisitos incognoscibles, 2, 54
- Casos de uso: requisitos en contexto, 136
- Requisitos del usuario, 50
- V
- Requisitos validados, 53
Validación, 154
Verificación, 153
Requisitos verificados, 53
Enlace vital, 186
Planificación de V&V, 85–86
- W.
- Walton, María, 43, 166, 191
Modelo de cascada
- comparación, 74–75
ilustrado, 76
- Watson, Mark, 97 Watts,
Frank B., 134, 165 Waugh, Penny,
71, 105 Webster, BF, 26 Weill,
P., 44 Weinberg, GM,
105, 203
- Whitten, Neal , 40, 44, 55, 60, 79,
117 Widerman, Max, 167 Widrig, Don, 18, 25, 79,
97, 101, 106 Wiegers, Karl,
31, 35, 44, 70, 71, 78, 88, 97,
104, 105, 106, 107, 135, 191, 195
- Wiley, Bill, 97
- Madera, Jane, 97, 106
- Aplicación de estructura de desglose del
trabajo (WBS), 156–59
desafío, 157
definido, 156
como “perro de un trabajo para
deshacer”, 158 organización, 157
- Productos de trabajo, 157
Redactar mejores requisitos, 56, 96
Redacción de casos de uso eficaces, 96, 136
- Y
- Joven, Don, 215
Joven, RR, 13, 25, 44, 55, 60, 97, 105, 106, 191
- Z
- Zachman, JA, 60 Marco
Zachman, 48, 60