



Proyectos de calidad comienzan con requisitos de calidad

Guilherme Siqueira Simões

17 - Julio - 2015

Agenda

- ¿Por qué preocuparse por la calidad en requisitos?
- ¿Qué es calidad?
- ¿Qué es requisito de software?
- La especificación de requisitos
- El rol de la especificación de requisitos
- El nivel de detalle para la especificación
- Criterios de calidad en requisitos
- Actividades que aseguran la calidad de los requisitos

¿Por qué calidad en requisitos?



- **47% de los fracasos** en proyectos se deben a la mala gestión
requisitos
*PMI's Pulse of the Profession: Requirements Management
A Core Competency for Project and Program Success - 2014

- **20% de los defectos** tienen su origen en requisitos
**Software Defects Origins and Removal Methods
Capers Jones - 2014

- Encontrar y corregir defectos en el software después de entregado es **>100 x más costoso** que hacerlo en la fase de requisitos
***Software Defect Reduction – Top 10 List
Barry Boehm y Victor Basili - 2001

¿Qué es calidad?



- Calidad es el grado en que un conjunto de características inherentes cumple con los requisitos - ISO 9000
- Pero, ¿Cómo asegurar calidad si hay errores en los requisitos?
(20% de los defectos tienen su origen en requisitos)

¿Qué es requisito de software?

- (1) Condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo
- (2) Condición o capacidad que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, estándar, especificación u otra documentación formalmente impuesta
- (3) Representación documentada de una condición o capacidad como en (1) o (2)



deseo (proyecto)



producto



documentación

especificación

IEEE Standard Glossary of Software
Engineering Terminology (IEEE 610)

Especificación de Requisitos



- Es un conjunto de requisitos que:
 - Ayuda a los clientes a describir con precisión lo que desean obtener (de un software)
 - Ayuda los desarrolladores a entender exactamente lo que quiere el cliente

Rol de la Especificación



- Ser un contrato entre cliente y desarrolladores
 - No se debe enfocar en aspectos de diseño o implementación
 - Además de ser detallado, debe promover la comunicación entre las dos partes
- El nivel de confianza entre las dos partes determina el nivel de detalle

Nivel de Detalle

Menos detalle	Más detalle
Clientes altamente involucrados	Desarrollo externo
Desarrolladores con experiencia considerable en el asunto del problema	Miembros del equipo del proyecto geográficamente dispersos
Existen antecedentes disponibles (por ejemplo: Reingeniería)	Pruebas basadas en los requisitos
Una solución de paquete será utilizada	Se requiere estimaciones con más precisión
Contexto más orientado a los cambios	Se requiere trazabilidad de los requisitos

Criterios de Calidad*



- **Correcta:** Cada requisito satisface la necesidad o la demanda legítima del negocio. Es decir, debe trazar alguna necesidad (requisito) de negocio
 - No contiene requisitos superfluos
 - Menos riesgos de *scope creep* y *gold plating*

• **Clara (sin ambigüedad):** Tiene una interpretación única para todo el público. El lenguaje natural, casi siempre usado para describir requisitos, es inherentemente ambiguo. Tips:

- Utilice las mejores prácticas para la escritura de texto
- Use un glosario para definir términos del contexto
- No use términos subjetivos o indeterminados: “etc”, “entre otros”, “bueno”, “amigable”, “flexible”, “portable”, “razonable”, “intuitivo”
- Use ejemplos

Criterios de Calidad



- **Completa:** Todos los elementos relevantes del contexto de interés (el dominio del problema) están descritos. Ejemplos:
 - Funcionalidades, aspectos de calidad, restricciones de diseño, interfaces externas
 - Definición de todas las respuestas para cada tipo posible de entrada al software
 - Rótulos y referencias a todas las figuras, tablas y diagramas en la especificación
- La trazabilidad ayuda a garantizar completitud
- Especificación con partes "para definir" es incompleta

- **Consistente:** No existen contradicciones entre los documentos de requisitos, sea en un mismo nivel o en niveles diferentes.

Ejemplos de contradicciones:

- Temporalidad: REQ-03 indica que el evento A precede al evento B; REQ-12 indica que los eventos A y B son simultáneos

- Dos requisitos utilizan diferentes nombres para el mismo objeto del mundo real

- A menudo, la inconsistencia surge de solicitudes de cambios mal asimilados en la especificación



•**Modificable:** Pueden realizarse cambios de una manera fácil, completa y consistente, sin comprometer la estructura y estilo de la especificación. Directrices:

- Tener una organización coherente y fácil de usar que incluye, por ejemplo, una tabla de contenido, un índice, glosario y control de cambios
- No ser redundante
- Expresar cada requisito por separado en lugar de combinarlo con otros requisitos

- **Priorizada:** Cada requisito de la especificación tiene atribuido un valor de importancia relativa basado en uno o más criterios, por ej.: riesgo, costo
- Una buena priorización asegura que el esfuerzo se centrará en las necesidades más críticas, lo que reduce los riesgos del proyecto
- La priorización permite:
 - Determinar los requisitos que se deben analizar primero
 - Planificar que requisitos implementar primero
 - Cuánto tiempo o atención se destinará a los requisitos

• **Verificable:** Algún método (costo-beneficio aceptable) puede ser establecido para demostrar objetivamente que el software cumple con el requisito. De lo contrario, éste debe ser eliminado o revisado. Ejemplos:

- El requisito no funcional "la interfaz de usuario debe ser fácil de usar" no es verificable debido a su subjetividad
- El requisito no funcional "95% de las operaciones se debe procesar en menos de 1 segundo cada uno" es verificable porque se expresa en términos mensurables

- **Trazable:** Establece la relación entre los requisitos, sus fuentes y sus productos derivados
- Hace la especificación más modificable, facilita el análisis del impacto de los cambios, y verificación de su corrección y completitud
- Ayuda a garantizar el cumplimiento del software con sus requisitos, identificando los requisitos que faltan o sobran y verificar si todo los objetivos de negocio están cubiertos por los requisitos y productos
- Ayuda en la Gestión de Riesgos: requisitos con muchas relaciones tienen mayores riesgos.

Cómo asegurar calidad: Verificación



- **Verificación de Requisitos:** Asegura que las especificaciones y modelos cumplen con los estándares. Ej.: errores de notación, datos incompletos o inconsistentes, malas prácticas
- Técnicas principales: Listas de Verificación y Revisiones
- Constituye una comprobación final para asegurarse de que los requisitos están listos para la validación por los clientes

Cómo asegurar calidad: Validación



- **Validación de Requisitos:** Asegura que todos los requisitos estén alineados con los requisitos del negocio. Es decir, tratan de garantizar que se cumplen todas las necesidades de negocio del cliente
- Técnicas principales: Revisiones y Prototipos

Conclusión



- Requisitos son claves para que los proyectos logren éxito
- Requisitos no equivalen a documentación
- Especificación detallada no siempre significa mejor calidad
- Verificación y validación deben siempre estar presentes
- Proyectos con calidad comienzan con requisitos de calidad

Para saber más...



- PMI's Pulse of the Profession: Requirements Management - A Core Competency for Project and Program Success – 2014
- A Guide to the Business Analysis Body of Knowledge – BABOK 3.0 (IIBA)
- Business Analysis for Practitioners: A Practice Guide (PMI)
- More About Software Requirements: Thorny Issues and Practical Advice (Karl E. Wiegers)
- IEEE Std 830 Recommended Practice for Software Requirements Specifications

¡Gracias por su atención!

¿Preguntas?

Guilherme Siqueira Simões

guilherme.simoes@fattocs.com

www.linkedin.com/in/guilhermesimoes

Skype: [guilherme.s.simoes](https://www.skype.com/people/guilherme.s.simoes)