

¿Qué es Software Quality Assurance? ¿Porqué facilita el ahorro de dinero ?

Introducción

El objetivo de este documento es explicar qué es Software Quality Assurance, cuáles son los beneficios económicos de su implementación y cómo interviene en las etapas del proceso de desarrollo.

Objetivos

La implementación de una disciplina de SQA tiene como principal objetivo aumentar la calidad de los entregables durante todo el proceso de desarrollo. Muchos requerimientos de calidad, sobre todo aquellos que tienen que ver con la performance, la usabilidad, la carga, la disponibilidad, etc. pueden ser tratados como riesgos. Es decir que, el hecho de que uno de ellos no se cumpla, implica un riesgo.

Entonces, al asegurar la calidad del software durante su proceso, se disminuyen los riesgos asociados, aumentando la predictibilidad del desarrollo de software. Esto trae aparejado una serie de beneficios de variada visibilidad.

Entre los que más se destacan, podemos nombrar:

- > Reducción de los **tiempos de desarrollo**, principalmente el tiempo de retrabajo generado en la fase de testing.
- > Optimización del **uso de los recursos**, que disminuye el costo de la infraestructura necesaria para soportar la aplicación.
- > Disminución del **costo de mantenimiento**, ya que se generan aplicaciones más seguras y estables.
- > Aumento de la **permeabilidad al cambio** y facilidad para medir el impacto del mismo
- > Asegura el cumplimiento de los **requerimientos**, tanto los funcionales como los de calidad.
- > Promueve el seguimiento de los **estándares** definidos
- > Provee **información** sobre la calidad del proyecto a los stake holders con menor conocimiento técnico.
- > Los desarrollos se vuelven más **predecibles**, facilitando las estimaciones

Actividades de SQA en el Proceso de desarrollo

SQA es una disciplina que está compuesta por una serie de actividades que acompañan al proceso de desarrollo. El objetivo de estas tareas es aumentar, administrar y monitorear la **calidad** de los entregables producidos. Para poder identificar estas actividades y el momento oportuno para realizarlas es necesario revisar el ciclo de vida de un proyecto. Para esto nos basamos en el análisis de fases/disciplinas/esfuerzo realizado en RUP por ser un proceso muy difundido en el mercado, aunque el mismo análisis puede aplicarse a otros procesos de desarrollo.

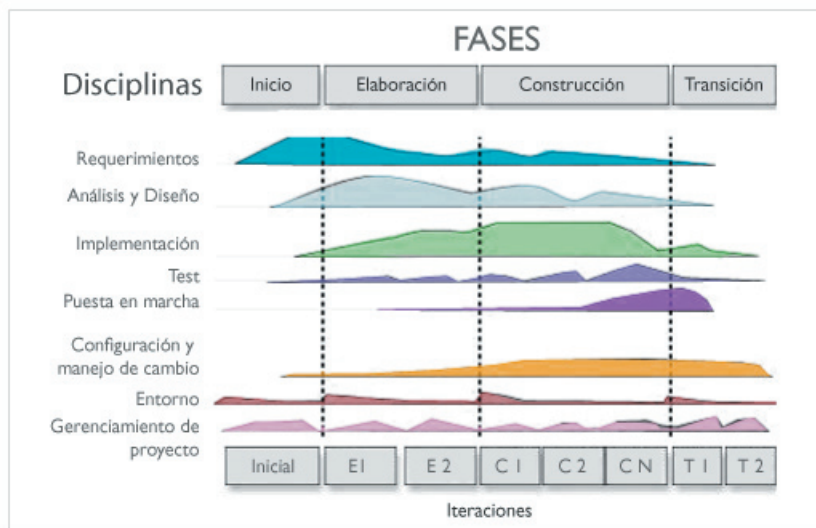


Figura 1: fases y disciplinas de RUP

Analizando el diagrama, notamos que el esfuerzo de cada disciplina varía según la fase del proyecto. De aquí se deriva que el momento para controlar la calidad de cada disciplina es cuando mayor esfuerzo se le dedica.

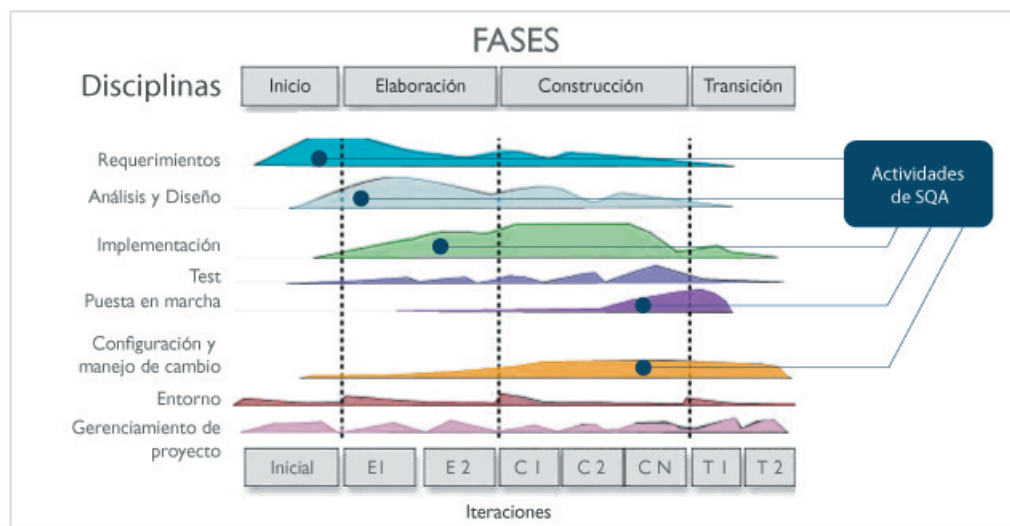


Figura 2: fases y disciplinas de RUP: relación con las actividades de SQA

En iniciativas de SQA saludables el impacto de la misma debería verse reflajado en todos los artefactos generados en el proceso de desarrollo. No limitándose solamente al testing o verificación funcional, la cuál es una acepción bastante difundida de la función de SQA, pero diametralmente desacertada. En una perspectiva más amplia del rol sinérgico de SQA en un proyecto de desarrollo, podemos encontrar, entre otras, las siguientes las actividades:

- > **Verificación de requerimientos:** esta actividad se concentra en validar la completitud, correctitud, claridad y no ambigüedad de los requerimientos de un sistema.
- > **Validación y verificación de documentación:** esta actividad se encarga de controlar la correctitud, completitud y no ambigüedad de la documentación. La documentación en UML es muy útil para esta práctica por el poder semántico que tiene y por la posibilidad de validar sintácticamente la documentación.
- > **Validación de arquitectura:** esta actividad es muy importante para evaluar la factibilidad de cumplir con los requerimientos no funcionales y detectar de forma temprana los principales riesgos asociados al proyecto.
- > **Control de diseño:** esta actividad se enfoca en validar el diseño lógico de los componentes, su distribución, interacción e identificar componentes que puedan ser reutilizables y el alcance de los requerimientos de calidad definidos por parte de cada uno de ellos.
- > **Control de código:** se subdivide en 2 actividades:
- > **Control estático del código:** es la validación del código contra un conjunto de reglas, *best practices* y estándares predefinidos.
- > **Control dinámico del código:** el control se focaliza en el uso de los recursos que hace la aplicación y la cobertura del código que hacen los test unitarios

De todas estas tareas las más comunes son las de testeado. y, aunque tienen un alto impacto ya que previenen que el grueso de los defectos sean percibidos por el usuario final, lamentablemente, como veremos luego, estas actividades son las que traen el menor costo beneficio por estar en etapas tardías del proceso de desarrollo.

Tareas

Es necesario tener en cuenta que para realizar algunas de estas actividades primero es necesario realizar otras actividades como ser:

- > Definición de estándares y *best practices* de desarrollo
- > Elección de herramientas para documentar y desarrollar.
- > Etc.

Estas tareas tienen que ver con el hecho que para poder validar la calidad de algo, es necesario contar, previamente, con la definición, requerimiento o estándar contra el cual validar.

Análisis de costos

Implementar alguna o todas de las actividades anteriormente descritas implica tiempo, esfuerzo y obviamente, dinero. Por esto es necesario analizar el ROI de esta implementación. Para esto nos basaremos en un estudio que analiza las etapas de un proyecto tradicional y los defectos introducidos, los encontrados y el costo de su reparación. Como se puede apreciar, la introducción del 85% de los defectos de la aplicación se produce al inicio de la etapa de construcción. Pero estos defectos se van encontrando paulatinamente, en etapas posteriores.

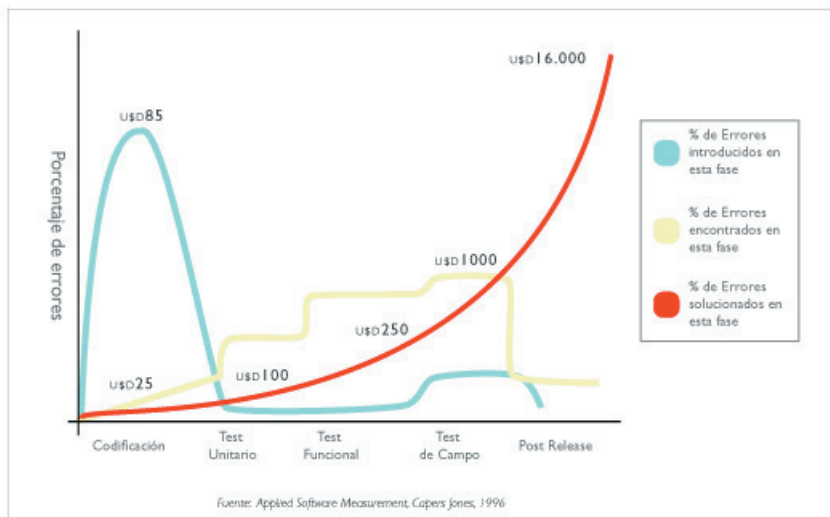


Figura 3: distribución de los errores a lo largo de las etapas de desarrollo y su costo asociado

Mientras más se demoren en encontrar un error, más costoso será repararlo. Por esto se proponen establecer **actividades que acompañen al proceso de desarrollo** (en lugar de puntos de control) para maximizar los defectos encontrados en cada etapa y minimizar la cantidad e impacto de los defectos que se encuentran en las etapas finales. Tratemos de calcular el costo de un defecto encontrado durante el testing o luego de la puesta en producción, donde el costo asociado no es sólo el de generar una nueva versión, sino que se le suma el costo de del test de regresión, la nueva puesta en producción y el tiempo que este bajo el sistema.

¿Por dónde empezamos?

Implementar todas las actividades de SQA al mismo tiempo es costoso e impráctico. Es mejor empezar por las actividades en donde se vean resultados de forma más rápida y efectiva. Además, una implementación paulatina y retroalimentada ayudará a que el proceso sea más armonioso y fácilmente aceptado. Si se está en las etapas iniciales del proyecto conviene empezar por la verificación de requerimientos y la validación de arquitectura, mientras que si ya se está avanzado, es mejor revisar el diseño o directamente el código.

Métricas de calidad

La única manera de poder mostrar cuales son los beneficios que aportan las actividades de SQA es a través de métricas que reflejen la evolución del proceso de desarrollo. Si no podemos medir las mejoras no podrán evaluarse la efectividad del proceso de SQA y tampoco podrá mejorarse. Ejemplos de este tipo de métricas podrían ser:

- > Número de incidencias reportadas en producción
- > Número de incidencias reportadas en testing
- > Cantidad de ciclos de testing/retrabajo necesarios.
- > Tiempo necesario para realizar un cambio (categorizado por impacto)

Conclusiones

SQA es un conjunto de actividades que tienen como objetivo bajar el **costo de desarrollo** alcanzando los parámetros de calidad establecidos. Es factible alcanzar calidades y costos establecidos pero es necesario saber, que no es posible injertar calidad deseada al final del proceso de desarrollo, sino que se debe crear a lo largo de todo el proceso. Mientras **antes** se comiencen con las actividades de SQA, mayores serán los **beneficios**.

Autores

Valerio Adrian Anacleto
adrian@epidataconsulting.com

Diego H. Mornacco
diegom@epidataconsulting.com

URL al artículo original

http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=57

Contacto Institucional

Para ampliar la información no dude en comunicarse con nosotros:

Tel: (+5411) 5031-0060/61
E-mail: info@epidataconsulting.com
Maipú 521 Piso 1, Oficina A
Ciudad Autónoma de Buenos Aires
(C1006ACE)

Sitio Web:

www.epidataconsulting.com