Software quality management solutions
Thought leadership paper

Rational. software

IBM

Optimización de la calidad del software: *balanceando la transformación y el riesgo del negocio*

Michael Lundblad
Gerente de programa de software Rational de IBM Software Group

Moshe Cohen
Gerente de ofertas de software Rational de IBM Software Group

Contenido

Introducción

Un estudio reciente de IBM de CEOs muestra que el 66 por ciento de los CEOs espera que sus organizaciones se inunden con cambios, en gran parte conducidos por la innovación y la transformación. Para ir al ritmo, las organizaciones de desarrollo de software necesitan lanzar software crítico en menos tiempo, pero esto incrementa el riesgo y usualmente pone en riesgo a la calidad. Así que la pregunta es, ¿Cómo puede usted ahorrar tiempo y reducir costos sin sacrificar la calidad?

Aunque la aviación tiene más de 100 años, a la mayoría de nosotros todavía nos sorprende ver a un avión despegar y aterrizar. Lo que es aún más increíble es cómo el software de piloto automático controla y realiza todo con muy poca intervención del piloto. Considere la destreza del Joint Strike Fighter (JSF), también conocido como el F-35. De punta hacia arriba, puede parar a la mitad del aire, mantener su posición y luego reanudar su vuelo supersónico. Esta maniobra también es controlada con software.

¿Y los carros? Los vehículos de hoy en día contienen de 100 a 150 computadoras. Son esencialmente redes de computadoras sobre ruedas, comunicándose continuamente una con otra así como con tecnología global que proporciona servicios como telemetría, posicionamiento global, seguridad dentro del vehículo y diagnósticos de sistemas.

Cada día interactuamos directamente e indirectamente con sistemas de software masivos. El inventario en nuestros supermercados favoritos está controlado mediante software; nuestras cuentas de cheques y de ahorros están controladas mediante software; nuestros préstamos, cuentas de servicios públicos, historial médico y elegibilidad para medicamentos controlados, todo se maneja con software.

Estos avances traen conveniencia y progreso, pero también traen riesgo y gastos de encargados. Paul Ehrlich dijo una vez, “Errar es humano, pero para realmente estropear las cosas se necesita una computadora”, Considere estos problemas relacionados con software, que fueron reportados por devtopics.com como de estar entre los 20 peores desastres del tipo.1

Durante el “Lunes Negro” (19 de octubre de 1987), el Dow Jones Industrial Average se desplomó 508 puntos, perdiendo 22,6 por ciento de su valor total. El S&P 500 se cayó 20,4 por ciento. La mayor pérdida en un día que Wall Street haya jamás sufrido fue causada por un defecto de software. Conforme los inversionistas se deshicieron de acciones en un éxodo masivo, los programas de cómputo de compra/venta de acciones generaron una inundación de órdenes de ventas, haciendo que los sistemas se colgaran y con esto dejaron a los inversionistas sin visión y sin capacidad de comerciar acciones.
En diciembre del 2008, un importante periódico israelí (Yediot Acharonot) imprimió una edición con el encabezado, “El bicho que dirige los hospitales está demente”, El artículo no se refería a una bacteria o a un virus que estaba enfermando a las personas; se referían a un error de software que causó que los resultados de análisis de sangre se asociaran a los pacientes equivocados. El ministerio de la salud de Israel dijo que el defecto apareció cuando los resultados de los análisis fueron transferidos electrónicamente de los laboratorios hacia los hospitales.
La entrega de software de calidad nunca ha sido más crítica para los negocios; de hecho, es crítica para la sobrevivencia del negocio. En la economía de hoy en día, los CIOs necesitan balancear delicadamente la conducción hacia la transformación del negocio con la gestión de los riesgos para el negocio. Los requerimientos cambian frecuentemente, los proyectos se retrasan, y los costos son examinados a conciencia —pero la calidad debe mejorar o el negocio quiebra. Estas tareas que al parecer son imposibles están obligando a los equipos de entrega de software a pensar de maneras nuevas e innovadoras, a examinar abordajes de desarrollo ágiles y a desafiar a los métodos tradicionalmente dominantes. Están cambiando su visión de la gestión de calidad, y la están observando como un esfuerzo continuo que demanda formación de equipos que colaboren entre ellos, técnicas de automatización, y reportes precisos en tiempo real, o uso de paneles de instrumentos de métricas para facilitar el análisis de discusión gobernado.
Este trabajo examina el balance entre la transformación y el riesgo. Revisa los motivos del cambiante escenario de desarrollo de software, y discute qué soluciones pueden ayudar a su negocio a mejorar la gestión de calidad para superar el tiempo de implementación, las reducciones de costos y la calidad mejorada del producto.

En un entorno de negocios donde debe generarse una alta calidad con poco financiamiento, la gestión de calidad de software es crítica.

La calidad de software requiere que los equipos encuentren un balance entre tres dimensiones: alcance (requerimientos), costo y tiempo (figura 1). Dependiendo de qué encuesta lea usted, aproximadamente el 80 por ciento del presupuesto de TI de una organización se gasta en operaciones. Del restante 20 por ciento, 70 a 80 por ciento es gastado encontrando y arreglando defectos en aplicaciones legadas. Consecuentemente, el financiamiento de recursos para la entrega de calidad de software es fijo, en el mejor de los casos.
La experiencia directa de IBM con organizaciones en todo el mundo y los datos recopilados por investigadores que trabajan con cientos de compañías muestran que la mayoría de las firmas invierten 25 por ciento o más de su tiempo y costo de ciclo de vida de desarrollo en aseguramiento de la calidad. Además, 30 por ciento de los costos de proyectos de desarrollo de software están asociados a re-trabajo, y 70 por ciento de esa cantidad está relacionada a errores de requerimientos.2 Con los procesos de desarrollo de hoy en día de herramientas y cascada, la calidad de software requiere más tiempo y dinero que nunca. Para permanecer competitivos, los negocios necesitan encontrar maneras de mejorar la calidad mientras acortan el tiempo de implementación del software crítico para el negocio.

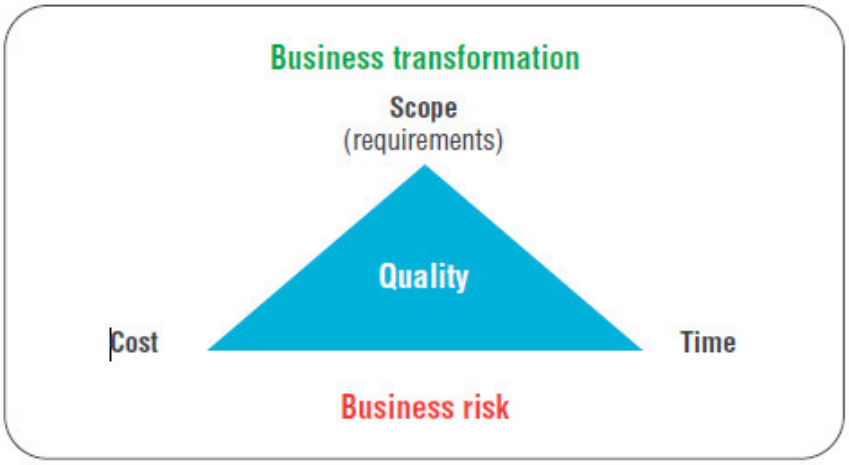

Figure 1: "The iron triangle"—managing the balance between business transformation and risk.

Del 20 por ciento restante de un presupuesto de TI que queda después de los gastos operativos, tres cuartos se gastan en encontrar y arreglar defectos.

El mantener un balance en el "triángulo de hierro" es crítico para las compañías que desea ir detrás de la innovación sin incrementar los riesgos.

Los requerimientos nunca están congelados … siempre permanecen cambiantes
Muchos de los defectos más graves llegan hasta el software a través de requerimientos: definición de requerimientos vaga, mala gestión de requerimientos o diseminación

incompleta de requerimientos hacia todos los participantes. Y estos problemas no sólo surgen al inicio de un proyecto; a parecen a lo largo de todo el ciclo de vida de aplicación. Los ejecutivos de alto nivel desearían que se formaran los requerimientos y luego fueran congelados para que no se les pudieran hacer cambios en el futuro. Esto en efecto ayudaría a incrementar la calidad del software, pero la realidad de hoy en día es que los requerimientos cambian constantemente. Reflejan la velocidad de los negocios de hoy, la urgencia por permanecer competitivos y la necesidad de estar en conformidad con las normatividades impuestas en muchos de nuestros procesos.

Consideremos el caso donde un CIO se da cuenta de que las nuevas normatividades requerirán el cambio de varias aplicaciones. Primero el CIO y su equipo necesitan descubrir cómo afectarán estos cambios al software de la compañía, y luego necesitan ajustar sus actividades de prueba para abordar estos cambios. Mientras tanto, la extensión de los cambios, el tiempo que toma el probar manualmente los cambios, y el costo asociado a la realización de estos cambios convencen a las compañías de que el incremento de automatización, las pruebas de regresión y los procesos iterativos y ágiles representan la mejor manera de gestionar estos inevitables efectos colaterales de actualizaciones de software.

Analizando el impacto del cambio de requerimientos en un mundo geográficamente distribuido

Continuemos con nuestro ejemplo de cambio regulatorio. La mayoría de las compañías lidiarían con esos cambios de la manera más natural y directa: Harían que cada uno de los equipos analicen los impactos a sus aplicaciones y luego actuarían de acuerdo a ello. En este escenario, probablemente acabaríamos con diferentes equipos efectuando simultáneamente tareas similares en los mismos productos. Pero es común para los negocios el tener múltiples proyectos al mismo tiempo en diferentes ubicaciones, ya sea internos o externalizados, y todos necesitan ser actualizados para reflejar los cambios regulatorios.

Las actualizaciones de software proporcionan otra oportunidad fértil para que los defectos invadan al código.

Cuando los requerimientos son malamente definidos o gestionados —o cuando cambian sin un seguimiento preciso — se pueden introducir defectos graves.

Uno no puede más que asumir que este abordaje crea una gran cantidad de desperdicio así como un incremento en el riesgo. ¿Y si los diferentes equipos interpretan a los nuevos requerimientos de manera diferente? ¿Y si acaban con casos de prueba que reflejan

diferentes comportamientos cuando se supone que deben ser idénticos? El lidiar con múltiples proyectos sin incrementar significativamente el riesgo requiere de un proceso de prueba donde un conjunto único de requerimientos pueda ser correlacionado a múltiples proyectos. Requiere de la capacidad de definir casos de prueba de donde se puedan generar y/o derivar scripts de prueba. Requiere de la capacidad de compartir y reutilizar casos de prueba entre proyectos. Requiere de un plan de prueba que, como los requerimientos, no sea un documento congelado o una página Web jerárquica. Es un activo dinámico siempre actualizado, el que permite la definición y compartir los procesos de prueba y estrategias, los reportes a nivel del negocio contra objetivos de calidad, y—igual de importante— el soporte para actividades de colaboración de equipo como revisiones y aprobaciones.

Riesgos asociados con los entornos de ejecución

El software de hoy en día es tan complejo que en muchos casos sencillamente es imposible garantizar aplicaciones libres de errores. En otros casos, incluso si es posible, no es práctico financieramente. Tome, por ejemplo, un caso donde un portal de contacto con el cliente necesita ser probado. El primer paso consiste en que el equipo de prueba desarrolle los casos de prueba, donde algunos son manuales mientras que otros son automatizados. Dependiendo de la complejidad de la aplicación y del entorno que se pretende para ella, un ciclo de pruebas puede tomar desde días hasta semanas o incluso meses. De hecho, asumamos que el portal da soporte a tan sólo dos idiomas y cuatro diferentes navegadores Web o versiones, y se ejecuta sobre tres bases de datos y un servidor de aplicaciones. Tan sólo estos pocos componentes requieren que el ciclo de prueba se repita 24 veces (2 x 4 x 3 x 1). Así que a lo que le toma una hora para ser probado, ahora le toma 24. Y lo que lleva una semana de 40 horas, ahora lleva 24 semanas, casi medio año. Conforme el número de configuraciones que necesitan ser probadas crece, es tentador el eliminar las pruebas de algunas configuraciones, y con esto se incrementa el riesgo de entregar software defectuoso.

Cuando una aplicación tiene varias configuraciones, existe la tentación de omitir algunas, pero al hacerlo se incrementa el riesgo de entregar software contaminado.

La colaboración es esencial para la defensa contra posibles defectos debido a que permite a todos los miembros del equipo comunicarse durante la definición de requerimientos y actualizaciones de software.

IBM investigó una manera promisoria para reducir significativamente el riesgo asociado con un gran número de entornos de ejecución. Los resultados mostraron que existe un pequeño subconjunto de combinaciones que puede detectar un porcentaje muy alto de defectos muy fácilmente. Esencialmente, acelera la detección de defectos. Como se muestra en la figura 2, las pruebas bajo múltiples entornos de ejecución siguen la típica

curva “lenta”, donde los defectos son detectados sin ninguna prioridad especial. La detección de defectos a lo largo del tiempo es en su mayoría en la zona aleatoria, y la planificación de pone en riesgo. Sin embargo, al aplicar optimizaciones inteligentes se lleva a la detección de defectos a lo largo del tiempo, hacia la zona optimizada, donde los defectos se detectan mucho más temprano en el ciclo de pruebas, ahorrando tiempo y reduciendo significativamente el riesgo de entregar software contaminado.

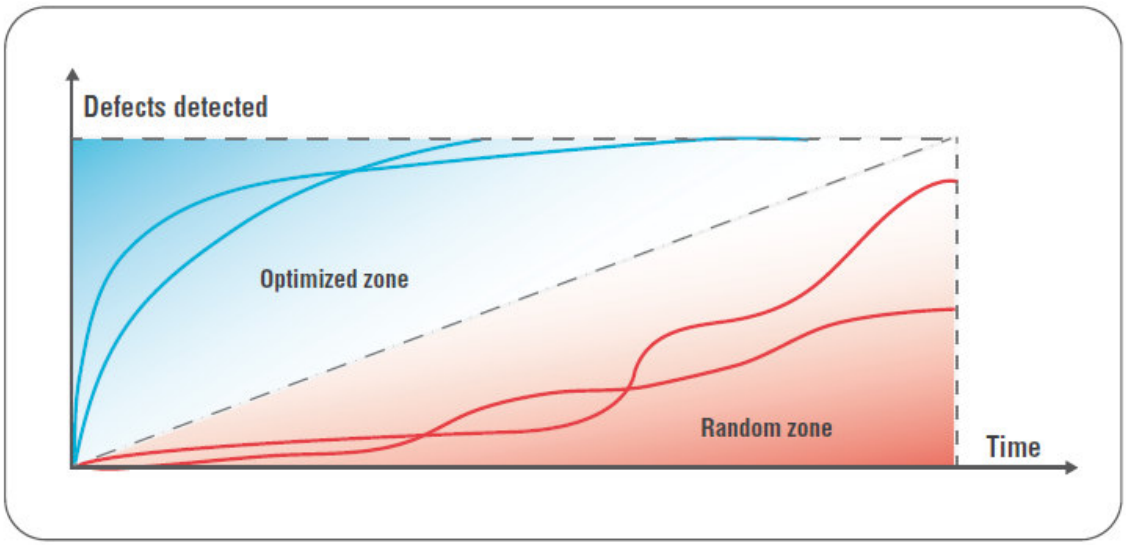

Figure 2: Speeding up the discovery of defects through smart optimizations

IBM ha descubierto que probar un pequeño subconjunto de combinaciones de configuración puede ayudar a los negocios a detectar un alto porcentaje de defectos muy rápido.

Cambio emergente de paradigma en la gestión de calidad

Para estar a la altura de este desafío de entrega de software de calidad, las organizaciones deben entablar procesos de colaboración que estén basados en automatización y estén acompañados de gobernanza de proyecto accionable a lo largo del ciclo de vida de entrega. Las pruebas tradicionales de aseguramiento de calidad (QA) simplemente validan que el software en desarrollo satisfaga las expectativas del usuario final en relación a la funcionalidad, disponibilidad y rendimiento previamente al despliegue. El encontrar defectos en la fase de pruebas de QA es mucho más caro y consume mucho más tiempo que encontrar los defectos más temprano en el ciclo de vida de desarrollo de la aplicación (figura 3).

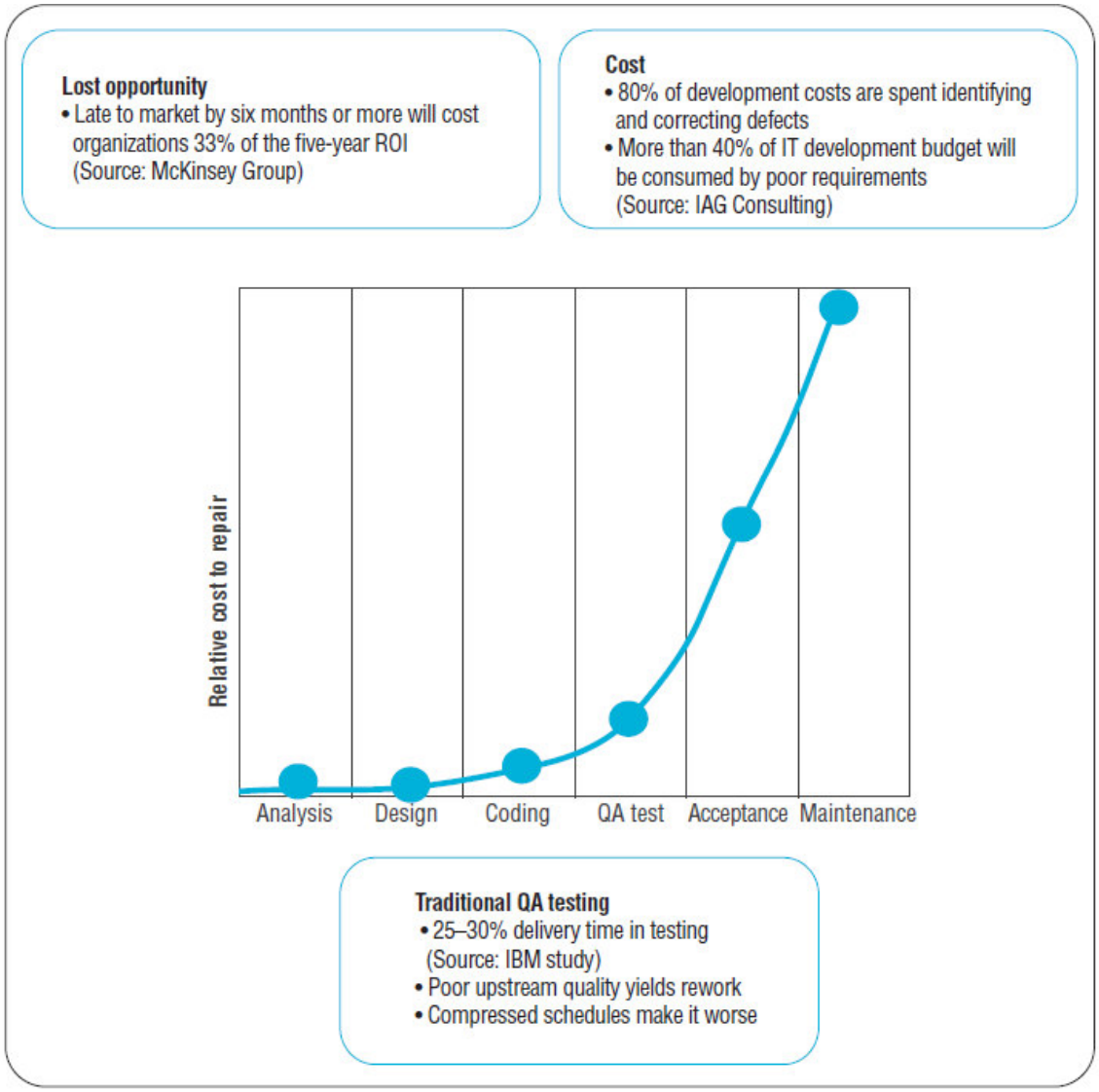

Figure 3: Opportunities to optimize quality and time to market exist throughout the application development lifecycle.

La incorporación de gestión de calidad en una etapa más temprana en el ciclo de vida de desarrollo de software puede ayudar a las organizaciones a evitar el alto costo de arreglar errores cuando sea más tarde dentro del proceso

La gestión de calidad se ha convertido en un deporte en equipo, pero la colaboración enfrenta muchas barreras. Las barreras geográficas bloquean la comunicación de los equipos entre múltiples husos horarios, y crean brechas en procesos que provocan re-trabajo. Las barreras de infraestructura crean problemas de integración de procesos, herramientas y datos que evitan que los equipos reciban información oportuna, lo que provoca demoras en la planificación. Finalmente, las barreras organizacionales presentan problemas con habilidades de dominio, débil gobernanza de proyecto y dilemas de externalización como la seguridad de propiedad intelectual.

¿Cómo mejoran las mejores firmas la calidad del software? Una revisión de IBM Global Services3 de 846 proyectos entre varios clientes reveló un valor diferenciado significativo con los abordajes colaborativos y automatizados. Los procesos de pruebas completos, las tecnologías integradas de ciclo de vida de principio a fin, el caso de prueba basado en la industria y la reutilización de script, y el análisis avanzado de defectos y los procesos de gestión de calidad han mostrado los siguientes rangos de mejora:

- Mejoras en la calidad del 30 al 70 por ciento
- Mejoras en el tiempo del ciclo del 20 al 50 por ciento
- Reducciones de costos del 15 al 60 por ciento

La colaboración enfrenta tres barreras principales: dispersión de equipos a lo largo de geografías, infraestructuras incompatibles incompatible y desalineación de las capacidades del personal.

	Developing repeatable industry test solutions			Advanced defect analysis	Developing repeatable test procedures applicable to future projects		Integrating end-to-end processes		
Asset	Test cases copied	Manual scripts copied	Manual scripts reuse	Prevent and block duplicate defects	Baseline and migrate documentation	Baseline artifacts	Leveraging component reuse	Dynamic updates of test assets	Total
Quantity	343	350	1,393	905	1,365	2,023	1,029	2,227	9,635
Hours saved	167	175	696	1,755*	683	1,011	515	557	5,558
Value	$16,690	$17,514	$69,633	$175,452	$68,254	$101,125	$51,459	$55,673	
Total			$103,387	$175,452		$169,379		$107,132	$555,799

* Hours saved assumes an average of four hours to detect the duplication. In reality, it often takes much longer.

Table 1: Differentiated business value results from an automated, process-led, collaborative approach to quality management.

Tabla 1 resumen de algunos de los descubrimientos de este estudio, que muestra cómo 846 proyectos de estos clientes ahorraron en promedio más de medio millón de dólares por proyecto.
¿Dónde encontraron ahorros estas organizaciones? En sus procesos de gestión de calidad. Desarrollar soluciones de prueba para la industria repetibles. • Parece razonable que dentro de cualquier industria dada, los casos de prueba y los scripts manuales se vuelvan razonablemente similares. Esto significa que copiar y reutilizar esos casos de prueba y scripts comunes podría brindar ahorros sustanciales. En este estudio, los ahorros promedio por proyecto estuvieron arriba de los US$100.000 (16.690 + 17.514 + 69.633). Realizar análisis de defecto avanzado.• No es fácil detectar y prevenir defectos duplicados, pero es muy importante detectarlos de manera temprana. Si usted no lo hace, corre el riesgo de tener a múltiples equipos trabajando en el mismo defecto sin saberlo, usualmente produciendo incluso más defectos. La detección automatizada de defectos duplicados no sólo mejora la calidad, sino que también reduce el riesgo y el costo. En este estudio, asumiendo un promedio de cuatro horas para detectar la duplicación, los ahorros promedio por proyecto fueron de US$175.452.

Las organizaciones pueden ahorrar tiempo y dinero al prestar más atención a la gestión de calidad a lo largo de todo el ciclo de vida de desarrollo y entrega de software.

Desarrollar procedimientos de prueba repetibles que sean aplicables a proyectos futuros. El desarrollar nuevos artefactos de prueba para cada proyecto, en vez de reutilizar artefactos disponibles, lleva tiempo y es innecesario. El estandarizar sobre procedimientos y técnicas de prueba repetibles puede ahorrar una cantidad considerable de tiempo. Los ahorros promedio por proyecto fueron de cerca de US$170.000 (68.254 + 101.125).

Integrando procesos de principio a fin. • El integrar gestión de requerimientos con técnicas de gestión de calidad proporciona rastreabilidad bidireccional completa de requerimientos a casos de prueba y resultados de prueba. También permite varios análisis de impacto, especialmente conforme los requerimientos cambian a lo largo del desarrollo. Estos son claves para aprovechar las ventajas competitivas al incrementar la calidad y disminuir el costo. Los ahorros promedio por proyecto en este estudio fueron e más de US$100.000 (51.459 + 55.673).

En resumen, un abordaje guiado por procesos, colaborativo y automatizado, hacia la gestión de calidad no sólo habilitó estrategias de mitigación de riesgo y calidad mejorada, sino que también ahorró, en promedio, ¡más de US$500.000 por proyecto!

Cumpliendo la misión

Para cumplir la promesa de innovación y crecimiento —por ejemplo, agilidad de negocios de arquitectura orientada hacia servicios (SOA) y desarrollo de software basado en componentes a lo largo de equipos distribuidos geográficamente — las organizaciones deben establecer una estrategia de gestión de calidad unificada y lista para responder. Los equipos de entrega de software y sistemas pueden ayudar en este esfuerzo al examinar tres áreas claves.

Una robusta estrategia de gestión de calidad puede ayudar a una organización a posicionarse para la innovación y el crecimiento.

Colaboración

Los equipos deben mantenerse en sincronía con procesos eficientes y dinámicos y flujos de trabajo basados en actividades. Los siempre cambiantes requerimientos del negocio deben extenderse a través del proceso de calidad para asegurar que los casos de prueba sean actualizados y que los desarrolladores comprendan los últimos requerimientos. QA y los gerentes de proyecto necesitan asegurar que sus equipos estén trabajando en las tareas de más alta prioridad. La planeación de pruebas debe ser continua e involucrar un abordaje orientado hacia metas con criterios de entrada/salida y configuraciones de entorno priorizadas.

Automatización

El desarrollo de software ha observado que las actividades de profesionales individuales, como pruebas funcionales y de rendimiento, se han vuelto automatizadas. Y los ingenieros de construcción de hoy en día usan scripts y herramientas de construcción que ahorran tiempo. Pero se necesita hacer más para automatizar el proceso y los pasos entre papeles para mejorar las eficiencias organizacionales, ahorrar dinero y acelerar la implementación (ej., reutilización de caso de prueba y script de prueba entre las líneas del negocio, suministro/análisis automatizado de laboratorio de pruebas, y traducción automatizada de modelos de casos de uso en casos de prueba).

Gobernanza accionable

Los resultados de todos los tipos de pruebas—incluyendo pruebas de unidad, funcionales, de integración y de escalabilidad—deben estar disponibles inmediatamente para reporte y análisis de tendencias. La integración de información técnica del proyecto con los análisis del negocio apoya la influencia de los tomadores de decisiones de alto nivel con respecto a la asignación y uso de recursos para alinear TI y los negocios. La gestión de pruebas y las posibilidades de planeación deben estar estrechamente integradas en el análisis de requerimientos y los procesos de definición. Los equipos de prueba comprometidos en la planeación temprana de caso de prueba deben

Las organizaciones deben establecer una estrategia de gestión de calidad que involucre colaboración, automatización y gobernanza accionable.

La plataforma IBM Jazz™ incluye posibilidades de ALM para dar soporte a la gestión de requerimientos, a las actividades de desarrollo y a la gestión de calidad, mientras que al mismo tiempo ayuda a los equipos a superar las barreras geográficas, de infraestructura y organizacionales para la colaboración. Adicionalmente, los hubs basados en Web permiten a las personas trabajar juntas para entregar calidad perdurable de software como un activo de negocios estratégico. Las nuevas soluciones permiten a los usuarios el:
Colaborar• a lo largo del negocio, con equipos de desarrollo y prueba con procesos dinámicos- y flujos de trabajo basados en actividades para planeación y ejecución de pruebas.
Automatizar• procesos de ciclo de vida de trabajo intenso y detectar problemas de calidad más temprano, reduciendo el tiempo de implementación, recortando costos y mitigando los riesgos para el negocio.
Reportar• métricas priorizadas hechas a la medida para individuos y para equipos, facilitando una mayor visibilidad y permitiendo a los tomadores de decisiones actuar con confianza.
Entregar• mayor predictibilidad al correlacionar patrones de despliegue exitosos a indicadores operacionales de rendimiento clave (KPIs).
Las compañías que tomen ventaja de las nuevas herramientas de definición/gestión de requerimientos serán capaces de:
Alinear• proyectos de desarrollo con objetivos del negocio para reducir el riesgo de fallas en el proyecto que cuestan a las organizaciones miles de millones de dólares anualmente.
Conducir• mejor la obtención y validación de requerimientos entre expertos del negocio y técnicos usando técnicas visuales y de colaboración comprobadas (ej., bosquejos de procesos de negocios, bosquejos de interfaz de usuario y guiones gráficos, y casos de uso).
Gestionar• el cambio a requerimientos y alcance del proyecto de manera más efectiva (ej. Análisis de impacto).
Mejorar• el tiempo de implementación del proyecto y el retorno de la inversión al reducir el re-trabajo debido a requerimientos malos o faltantes.

Las herramientas de gestión de calidad destacan posibilidades de colaboración, automatización y reporte para ayudar a las organizaciones a entregar productos de mayor calidad —una y otra vez.

Como parte de una estrategia de calidad general ALM, las herramientas de entrega de software como las descritas en la tabla 2 también pueden ayudar a asegurar la calidad y el rendimiento del software crítico para el negocio.

Tools	Description
Architecture modeling	• Validates user-defined rules that represent architectural constraints • Automatically detects design patterns and important object-oriented structures • Detects structural antipatterns (such as tangles, hubs and butterflies) that degrade performance • Automatically refactors tangles through quick fixes
Development	• Helps developers detect memory corruption, leak detection, performance profiling and code coverage
Automated build validation testing	• Prevents faulty builds from being deployed into the test lab or system test environment • Enables developers to take advantage of off-hours cycles to test an application's stability and functionality
Manual testing	• Promotes best practices such as test modularity and reusability to transition teams from manual to automated testing
Functional and system testing	• Enables teams to build tests that manually or automatically check for regression and functional errors • Shortens automation test cycles to improve quality through broader and deeper test coverage • Encourages more accurate, reliable and reproducible tests • Extend functional testing to Oracle applications environment, including OracleForms*
Load and scalability testing	• Determines load and scalability thresholds for technologies and applications such as Java* Platform, Enterprise Edition (Java EE); Web (particularly portals); SOA; Siebel; Oracle and SAP • Ensures that a software application can scale and perform to meet SLAs and user expectations • Allows testers to pinpoint the source of performance bottlenecks, enabling them to navigate to source code without wasting time wading through multiple levels of code • Enables a better return on hardware investments by executing pre-deployment capacity planning tests that size the server resources needed to achieve the desired performance and throughput • Extend performance testing to enable testing of Oracle NCA solutions such as Oracle Business Suite *
Production application support solutions	• Provides collaboration between operations and development teams for closed-loop application support, problem isolation and repair • Captures transaction log and trace information, as well as extended system resource data for more granular problem determination, thus reducing response time to application support diagnoses • Enables faster restoration of service levels to the business

Table 2: ALM tools contribute to quality lifecycle delivery.

Las herramientas ALM ayudan a construir y a asegurar la calidad a través de la elaboración de modelos de arquitectura, desarrollo, pruebas funcionales y de sistemas, pruebas de validación de construcción automática, pruebas manuales, pruebas de carga y de escalabilidad, y soporte de aplicación de producción.

Resumen
Durante esta década hemos observado una colisión de tres vías entre la búsqueda por transformación del negocio a través de innovación, la necesidad de mejora en la calidad del software para gestionar el riesgo para el negocio, y la demanda por reducir costos para la sobrevivencia económica.
El cambio de la industria hacia trabajo en equipo colaborativo entre equipos distribuidos geográficamente, junto con procesos automatizados y reportes para dar soporte a la gobernanza accionable, está cambiando el rostro de la entrega de software de calidad.
La gestión de calidad ya no sólo tiene que ver con equipos de QA con herramientas automatizadas de prueba dando soporte a la entrega tradicional de software; ni es acerca de procesos de control de gestión de calidad que hacen lenta la entrega o simplemente prueban y reparan defectos. La gestión de calidad tiene que ver con insertar la calidad en un ciclo de desarrollo interactivo y un programa de aptitud de software de ciclo cerrado, apoyado por herramientas, datos y métricas rastreables integradas. La gestión de calidad efectiva torna al proceso de entrega más manejable y menos doloroso, y ayuda a construir confianza entre los equipos de operaciones. Esencialmente, rompe el triángulo de hierro mencionado antes en este trabajo debido a que permite a los gerentes el optimizar alcance, costo y tiempo mientras que mejora la calidad.
La habilitación de la gestión de calidad a lo largo del ciclo de vida y la detección de defectos más temprano en el proceso, reducen el costo y mejoran la credibilidad. La comunicación constante vía un conjunto común y bien entendido de requerimientos —y respuestas rápidas a cambios en esos requerimientos —puede inyectar calidad en el desarrollo de software desde el inicio. Adicionalmente, la gobernanza diligente de los procesos de construcción y prueba para corregir cursos y asignar recursos, puede ayudar a las organizaciones a volverse más flexibles, a abordar las normatividades de conformidad, a reaccionar más rápido a las condiciones cambiantes del mercado y, a final de cuentas, a conducir el crecimiento del negocio. Dentro de esta economía que no perdona, el éxito o el fracaso usualmente dependen de una cosa: ¿de quién es el producto de mejor calidad?
Para obtener más información
Para saber más acerca de las soluciones de gestión de calidad de IBM, llame a su representante de IBM, o visite:
ibm.com/software/rational/offerings/quality
Un agradecimiento especial al Ron French de IBM Global Business Services por compartir descubrimientos claves de su trabajo y del trabajo de su equipo.

IBM CorporationSoftware GroupRoute 100Somers, NY 10589U.S.A.

Producido en los Estados Unidos de América Marzo del 2009

1 http://www.devtopics.com/20-famous-software-disasters

2 Walker Royce, *Software Project Management: A Unified Framework*, Addison-Wesley Professional, Indianapolis, 1998.

3 IBM Global Services, SEANT (Systems Engineering Architecture & Test) ReUse Program, 2007.