

## **Optimización de calidad del software: equilibrar la transformación y riesgo de los negocios**

*Michael Lundblad, administración de programas, software  
Rational, IBM Software Group*

*Michael Lundblad, administración de ofertas, software  
Rational, IBM Software Group*

---

## Contenidos

---

- 2 Introducción**
- 4 Equilibrar la transformación y riesgo de los negocios**
- 8 Cambio emergente de paradigma en la administración de calidad**
- 11 Implementar la visión**
- 13 Soluciones de administración de calidad**
- 16 Resumen**
- 16 Para obtener más informaciones**

***Lo percibamos o no, interactuamos con sistemas masivos de software todos los días.***

### Introducción

Un estudio reciente de IBM sobre directores ejecutivos muestra que el 66% de los CIOs esperan que sus organizaciones sean continuamente alcanzadas por el cambio, impulsado en su mayor parte por la innovación y la transformación. Para mantener el paso, las organizaciones de desarrollo de software necesitan liberar nuevas versiones de software crítico de negocios en menos tiempo, aunque esto aumenta el riesgo y a menudo produce un compromiso de la calidad. Y así se presenta la siguiente pregunta, *¿Cómo se puede ahorrar tiempo y reducir costos sin sacrificar la calidad?*

Aunque la aviación ya tiene más de 100 años de historia, la mayoría de nosotros todavía se sorprende ante la vista de un avión al despegar y aterrizar. Lo que es más increíble aún, es cómo el software de piloto automático controla y ejecuta la totalidad de la acción con poca intervención del piloto. Considere la agilidad y destreza del “Joint Strike Fighter” (JSF), también conocido como F-35. Apuntando hacia arriba, puede detenerse en el aire, mantener su posición y luego retomar su vuelo supersónico. Esta maniobra también es controlada a través del software.

¿Y qué podemos decir acerca de los automóviles? Los vehículos de hoy en día contienen entre 100 y 150 computadoras. Ellos son esencialmente redes de computadoras sobre ruedas, comunicándose continuamente unas con otras, así como también tecnología global, que proporcionan servicios tales como telemetría, posicionamiento global, seguridad vehicular y diagnósticos de sistema.

Todos los días interactuamos directa e indirectamente con sistemas masivos de software. El inventario en nuestros supermercados favoritos es controlado a través del software; nuestras cuentas corrientes y de ahorro también; nuestras hipotecas; facturas de servicios públicos, historia clínica y elegibilidad para drogas de prescripción, todas son controladas por el software.

Estos avances traen comodidad y progreso, aunque también generan riesgos y gastos relacionados. Paul Ehrlich dijo una vez, “Errar es humano, aunque para realmente arruinar las cosas, se necesita una computadora”. Considere los siguientes problemas relacionados al software, reportados por devtopics.com, como los 20 peores desastres de este tipo.<sup>1</sup>

---

## Destaque

---

En el “lunes negro” (19 octubre de 1987), el índice industrial Dow Jones se derrumbó 508 puntos, perdiendo el 22,6% de su valor total. El S&P 500 cayó 20,4%. La mayor pérdida que Wall Street sufrió en un único día, en toda su historia fue causada por un defecto de software. A medida que los inversores huían de las acciones en un éxodo masivo, programas informáticos de negociación generaron masivas órdenes de venta, el colapso de sistemas, y dejaron a los inversores efectivamente ciegos e incapaces de negociar.

En el mes de diciembre de 2008, un periódico israelí líder (*Yediot Acharonot*) imprimió una edición con el encabezado, “La contaminación que enloquece a los hospitales”. El artículo no se refería a una bacteria o un virus que enfermaba a las personas; sino de un defecto de software que produjo que los resultados de pruebas de sangre fueran asociados a los pacientes equivocados. La oficina del ministro de salud de Israel dijo que dicho defecto apareció, cuando los resultados de las pruebas fueron transferidos electrónicamente de los laboratorios a los hospitales.

***En un entorno de negocios donde la alta calidad debe ser alcanzada con poca financiación, la administración de la calidad del software es crítica.***

La entrega de software de calidad nunca fue más crítica para los negocios; de hecho, es crítica para que dichos negocios sobrevivan. En la economía de hoy en día, los CIOs necesitan equilibrar con delicadeza el impulso hacia la transformación de los negocios, con la administración de riesgo empresarial. Los requerimientos cambian a menudo, los proyectos se atrasan, los costos se investigan - pero la calidad debe mejorar o los negocios fracasan. Esta tarea aparentemente imposible, está forzando a los equipos de entrega de software, a pensar desde perspectivas nuevas, examinar abordajes ágiles de desarrollo, y desafiar métodos dominantes históricamente. Están cambiando su punto de vista sobre la administración de calidad, viéndola como un esfuerzo continuo que exige el trabajo colaborativo en equipo, técnicas de automatización, y reporte preciso y en tiempo real, o intercambio de conocimientos que faciliten el análisis racional de la discusión.

Este documento examina el equilibrio entre la transformación y el riesgo. Observa los motivos de la cambiante situación del desarrollo del software, y discute cuáles soluciones pueden ayudar a su empresa a mejorar la administración de la calidad para cumplir con excelencia y a tiempo con el mercado, reducciones de costos y acentuada calidad del producto.

---

**Destaque**

---

***Del 20% de un presupuesto de TI, residual luego de los gastos de operación, tres cuartas partes se destinan al hallazgo y reparación de defectos***

***Mantener un equilibrio en “el triángulo de hierro” es crítico para la compañía que desea alcanzar la in aumentar el riesgo.***

**Equilibrar la transformación y riesgo de los negocios**

La calidad del software requiere que los equipos encuentren un equilibrio entre tres situaciones: alcance (requerimientos), costo y tiempos (figura 1). Dependiendo de la encuesta estudiada, aproximadamente el 80% del presupuesto de TI de una organización, se destina a operaciones. Del 20% restante, entre el 70% y el 80% se destina a encontrar y corregir defectos en las aplicaciones heredadas. Consecuentemente, la financiación de recursos para la entrega de calidad de software es fija en el mejor de los casos.

La experiencia directa de IBM con organizaciones en todo el mundo, y datos reunidos por investigadores que trabajan con cientos de compañías, muestran que la mayoría de las firmas invierten 25% o más de su ciclo de vida de desarrollo y costos, en el control y garantía de la calidad. Además, el 30% de los costos del proyecto de desarrollo de software, está asociado con el re-trabajo, y el 70% de dicha suma está relacionado a los errores de requerimientos.<sup>2</sup> Con los procesos de desarrollo en cascada y herramientas actuales, la calidad del software requiere más tiempo y costos que nunca. Para permanecer competitivos, las empresas necesitan encontrar formas de mejorar la calidad, al mismo tiempo de acortar los tiempos de entrega al mercado del software crítico de negocios.



Figura 1: “El triángulo de hierro” – administrar el equilibrio entre la transformación de los negocios y el riesgo.

---

### **Destaque**

---

***Cuando los requerimientos están pobremente definidos o administrados – o cuando cambian sin capacidad de seguimiento precisa – pueden ser introducidos serios defectos.***

***Las actualizaciones de software proveen una oportunidad fértil para que los defectos invadan el código.***

Los requerimientos nunca son los mismos ... simplemente continúan cambiando. Muchos de los más serios defectos se generan en el software debido a los requerimientos: definición vaga de dichos requerimientos, administración pobre de los mismos, o diseminación incompleta de los requerimientos a todos los participantes. Y estas cuestiones no solamente aparecen al comienzo de un proyecto, sino que se presentan a lo largo de todo el ciclo de vida de la aplicación. El autoengaño entre los ejecutivos de nivel superior hace que los requerimientos se formen y se congelen, y de esa forma que no pueden ser realizados cambios en el futuro. Ciertamente esto ayuda a aumentar la calidad del software, aunque la realidad actual es tal que los requerimientos continúan cambiando. Ellos reflejan la velocidad actual de los negocios, la urgencia por permanecer competitivos, y la necesidad de cumplir con las regulaciones impuestas en muchos de nuestros procesos.

Consideremos un caso donde un CIO percibe que las nuevas regulaciones requerirán el cambio de una serie de aplicaciones. Primero el CIO y su equipo necesitan imaginar cómo dichos cambios afectarán el software de la compañía, y luego necesitan ajustar sus actividades de pruebas para resolver dichos cambios. Mientras tanto, el alcance de los cambios, el tiempo que lleva para probarlos manualmente, y el costo asociado con la realización de dichos cambios, convencen a las compañías de que una mayor automatización, prueba de regresión, y ágiles procesos iterativos, son la mejor forma de administrar estos inevitables efectos colaterales de las actualizaciones de software.

Analizando el impacto de requerimientos cambiantes en un mundo geográficamente distribuido

Continuemos con nuestro ejemplo de cambio regulatorio. La mayoría de las compañías deberán enfrentarse con dichos cambios en la forma más directa y natural: Deberán conseguir que los equipos analicen los impactos en sus aplicaciones y actuar de la forma correspondiente. En este escenario, probablemente terminemos con diferentes equipos realizando actividades similares en forma simultánea sobre los mismos productos. Aunque es común que las empresas tengan múltiples proyectos al mismo tiempo en diferentes publicaciones, dentro de la empresa o tercerizadas, y todos ellos necesitan ser actualizados para reflejar los cambios regulatorios.

---

**Destaque**

---

***La colaboración es esencial para defenderse contra posibles defectos, porque permite a los miembros del equipo comunicarse durante la definición de los requerimientos y actualizaciones de software.***

***Cuando una aplicación posee numerosas configuraciones, es tentador omitir la prueba de algunas de ellas, pero ello aumenta el riesgo de entregar software contaminado.***

Uno no puede ayudar sino asumir que esta aproximación crea un grado significativo de pérdida, así como también riesgos aumentados. ¿Qué sucede si equipos diferentes interpretan los nuevos requerimientos de forma diferente? ¿Y qué sucede si acaban con casos de prueba que reflejan diferentes comportamientos, cuando se supone que sean idénticos? Enfrentarse a muchos proyectos sin aumentar significativamente el riesgo, requiere un proceso de pruebas donde un único conjunto de requerimientos pueda ser referido a múltiples proyectos. Se requiere la habilidad de definir entornos de prueba, desde los cuales scripts de pruebas puedan ser generados y/o derivados. Se requiere la capacidad de compartir y rehusar entornos de prueba entre proyectos. Se requiere un plan de prueba que, al igual que los requerimientos, no es un documento congelado o una página web jerárquica. Es un activo vivo, dinámico, siempre actualizado, que permite la definición y el intercambio de procesos de prueba, estrategias, e informes a nivel de negocios, comparando objetivos de calidad, e igualmente importante, dar soporte a actividades colaborativas en equipo tales como revisiones y aprobaciones.

Riesgos asociados con ambientes de ejecución

El software de hoy en día es tan complejo que en muchos casos es simplemente imposible garantizar aplicaciones libres de error. En otros casos, incluso si es posible, no es financieramente práctico. Por ejemplo considere un caso donde el portal de contacto con el cliente necesita ser probado. El primer paso necesario es que el equipo de pruebas desarrolle entornos de prueba, algunos de los cuales son manuales mientras que otros son automatizados. Dependiendo de la complejidad de la aplicación, y el ambiente para el cual fue pensada, un ciclo de pruebas puede demorar desde algunos días hasta semanas o incluso meses. De hecho, asumamos que el portal da soporte a solamente dos idiomas, y cuatro diferentes navegadores de web, o versiones, y que se ejecuta sobre tres bases de datos y un servidor de aplicaciones. Estos pocos componentes solamente, aún requieren que el sitio de pruebas se repita 24 veces (2 x 4 x 3 x 1). Por lo tanto lo que lleva una hora de pruebas, ahora representa 24. Lo que lleva una semana de 40 horas, ahora representa 24 semanas, casi la mitad de un año. A medida que el número de configuraciones que necesitan ser probadas crece, es tentador eliminar la prueba de algunas configuraciones, y por lo tanto aumenta el riesgo de entrega de software con defectos.

---

**Destaque**

---

***IBM descubrió que probando un pequeño subconjunto de combinaciones de configuración, puede ayudar a los negocios a detectar un muy alto porcentaje de defectos, muy rápidamente.***

IBM investigó una forma prometedora para reducir significativamente el riesgo asociado con un gran número de ambientes de ejecución. Los resultados mostraron que existe un pequeño subconjunto de combinaciones que pueden detectar un muy alto porcentaje de defectos, muy rápidamente. Esencialmente, aumenta la velocidad de detección de defectos. Como se muestra en la figura 2, realizar pruebas bajo múltiples ambientes de ejecución sigue la típica curva “lenta”, donde los defectos son detectados sin prioridad especial. La detección de defectos a lo largo del tiempo se realiza principalmente en la zona aleatoria, y los cronogramas son comprometedores. Sin embargo, aplicar optimizaciones inteligentes, lleva a la detección de defectos a la zona optimizada a lo largo del tiempo, donde los defectos son detectados de forma mucho más temprana en el ciclo de pruebas, ahorrando tiempo y reduciendo significativamente el riesgo de entrega de software contaminado.

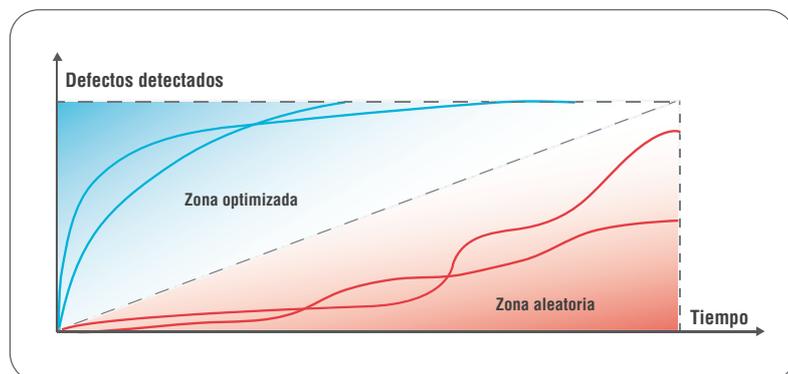


Figura 2: Acelerar el descubrimiento de defectos a través de optimización es inteligentes.

**Destaque**

*Incorporar administración de calidad de forma más temprana en el ciclo de vida de desarrollo de software, puede ayudar a las organizaciones a evitar el alto costo de reparar defectos cuando ya es tarde.*

**Cambio emergente de paradigma en la administración de calidad**

Para resolver el desafío de entrega de software de calidad, las organizaciones deben comprometerse en procesos colaborativos, impulsados por la automatización, y acompañados por un gobierno accionable del proyecto, a lo largo del ciclo de vida de la entrega. La tradicional prueba de garantía de calidad (QA), simplemente valida el hecho de que el software en desarrollo satisface las expectativas del usuario final de funcionalidad, disponibilidad y desempeño, con prioridad a la implementación. Descubrir defectos en la fase de pruebas de QA es mucho más caro y consume más tiempo, que encontrar defectos más tempranamente en el ciclo de vida del desarrollo de la aplicación (figura 3).

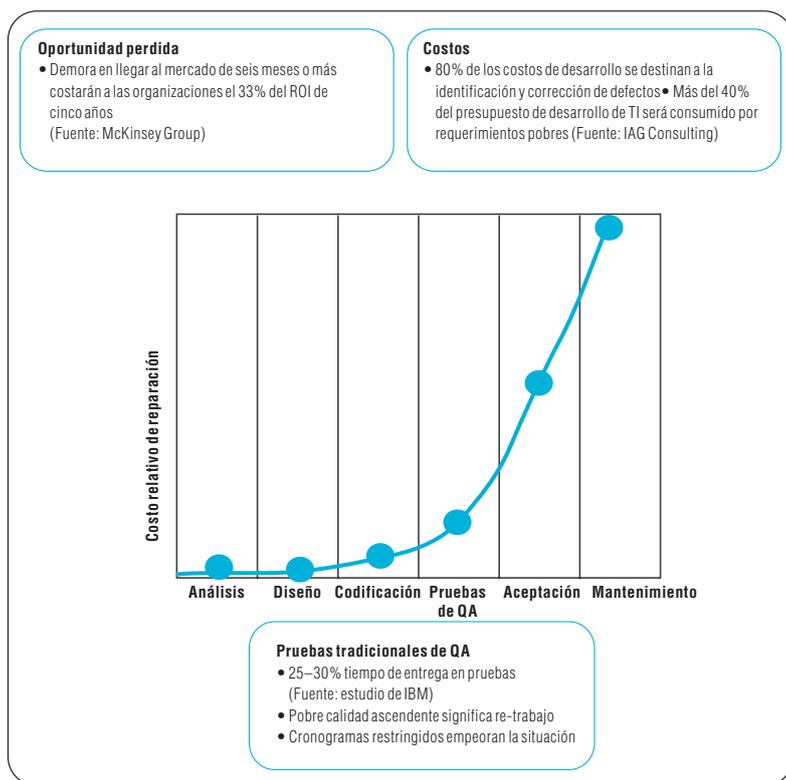


Figura 3: Oportunidades para optimizar la calidad y el tiempo de llegada al mercado existen a través del ciclo de vida del desarrollo de la aplicación.

---

## Destaque

---

***La colaboración enfrenta tres barreras principales: la dispersión del equipo a lo largo de geografías, infraestructuras incompatibles y falta de alineación de la capacidad del personal.***

La administración de calidad se ha convertido en un deporte de equipo, aunque la colaboración enfrenta muchas barreras. *Barreras geográficas* bloquean la comunicación del equipo entre múltiples zonas horarias, y crean intervalos de proceso que producen trabajo redundante. *Barreras de infraestructura* crean problemas de procesos, herramientas e integración de datos, que evitan que los equipos reciban la información en forma oportuna, lo que produce cronogramas atrasados. Finalmente, barreras de organización introducen problemas con la experiencia y conocimiento del dominio, gobierno de proyecto débil, y dilemas de tercerización, como la seguridad de la propiedad intelectual.

¿Cómo mejoran la calidad del software las firmas más renombradas en su categoría? Una revisión de IBM Global Services<sup>3</sup> de 846 proyectos entre múltiples clientes, revelaron un valor diferenciado significativo con aproximaciones colaborativas y automatizadas. Extensos procesos de prueba, tecnologías integradas de ciclo de vida punta a punta, entornos de pruebas basados en industrias y reutilización del código, análisis avanzados de defectos, y procesos de administración de calidad, han demostrado los siguientes alcances de las mejoras:

- *Mejora de la calidad entre el 30% y el 70%*
- *Mejoras de la duración del ciclo entre el 20% y el 50%*
- *Reducciones de costos entre el 15% y el 60%.*

	Desarrollar soluciones de prueba de industrias repetibles.			Análisis avanzado de defectos	Desarrollar procedimientos de prueba repetibles aplicables a proyectos futuros.		Integrar procesos punta a punta		
Activos	Entornos de pruebas copiados	Scripts manuales copiados	Reutilización de scripts manuales	Evitar y bloquear defectos duplicados	Documentación de línea de base y migración	Artefactos de línea de base	Impulsar la reutilización de componentes	Actualizaciones dinámicas de activos de prueba	Total
Cantidad	343	350	1,393	905	1,365	2,023	1,029	2,227	9,635
Obras ahorradas	167	175	696	1,755*	683	1,011	515	557	5,558
Valor	\$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

\* Horas ahorradas asume un promedio de cuatro horas para detectar la duplicación. En realidad, a menudo lleva mucho más tiempo.

La tabla 1 resume algunos de los hallazgos provenientes de este estudio, que muestran cómo 846 de estos proyectos de clientes, ahorraron un promedio de más de medio millón de dólares por proyecto.

¿Dónde encontraron ahorros estas organizaciones? En sus procesos de administración de calidad.

**Las organizaciones pueden ahorrar tiempo y dinero prestando más atención a la administración de calidad a lo largo de la totalidad del ciclo de vida de entrega y desarrollo del software.**

- **Desarrollar soluciones de prueba de industrias repetibles.** Parece razonable que dentro de cualquier industria dada, los entornos de pruebas y scripts manuales se hagan en bastante similares. Esto significa que copiar y reusar dichos entornos de prueba comunes y scripts, pueden producir ahorros sustanciales. En este estudio, el ahorro promedio por proyecto fue de más de \$100.000 (16.690 + 17.514 + 69.633).
- **Realizar análisis avanzados de defectos.** No es fácil detectar y evitar efectos duplicados, aunque es muy importante detectarlos en forma temprana. En caso contrario, se corre el riesgo de tener múltiples equipos trabajando sobre el mismo defecto, sin saberlo, a menudo produciendo incluso más defectos. La detección automatizada de defectos duplicados, no solamente mejora la calidad, sino que también reduce riesgos y costos. En este estudio, asumiendo un promedio de cuatro horas para detectar la duplicación, los ahorros promedios por proyecto fueron de \$175.452.

---

**Destaque**

---

- **Desarrollar procedimientos de prueba repetibles que son aplicables a proyectos futuros.** Desarrollar nuevos artefactos de prueba para cada proyecto, en vez de reutilizar artefactos disponibles, consume tiempo y es innecesario. Estandarizar procedimientos y técnicas de prueba repetibles puede ahorrar una cantidad considerable de tiempo. Los ahorros promedio por proyecto fueron de aproximadamente \$170.000 (68.254 + 101.125).
- **Integrar procesos punta a punta.** Integrar la administración de requerimientos con técnicas de administración de calidad, proporciona rastreabilidad total bidireccional desde requerimientos a entornos de pruebas y resultados de pruebas. También permite diversos análisis de impacto, especialmente a medida que los requerimientos cambian a través del desarrollo. Éstos son clave para impulsar las ventajas competitivas al aumentar la calidad y disminuir los costos. Los ahorros promedio por proyecto en este estudio fueron mayores a \$100.000 (51.459 + 55.673).

En resumen, una aproximación automatizada colaborativa y gobernada del proceso a la administración de calidad, no solamente permitió estrategias de disminución de riesgo y mejor calidad, sino que también ahorró en promedio, más de \$500.000 por proyecto!

#### **Implementar la visión**

Para implementar la promesa de innovación y crecimiento – por ejemplo, la agilidad de negocios desde una arquitectura orientada a servicios (SOA), y desarrollo de software basado en componentes, entre equipos distribuidos geográficamente – las organizaciones deben establecer una estrategia de administración de calidad ágil en la respuesta y unificada. Los equipos de entrega de sistemas y software pueden ayudar en este esfuerzo, examinando tres áreas clave.

***Una fuerte estrategia de administración de calidad puede ayudar a posicionar a una organización en el camino de la innovación y crecimiento.***

---

## Destaque

---

***Las organizaciones deben establecer una estrategia de administración de calidad que involucre la colaboración, automatización y control accionable.***

### Colaboración

Los equipos deben mantenerse en sincronía con procesos dinámicos, simplificados y flujos de trabajo basados en actividad. Los requerimientos siempre cambiantes de negocios, deben propagarse a través del proceso de calidad, para garantizar que los entornos de prueba sean actualizados y que los desarrolladores entiendan los últimos requerimientos. Los líderes de proyecto y QA necesitan garantizar que sus equipos están trabajando en las tareas de más alta prioridad. La planificación de pruebas debe ser continua e involucrar una aproximación orientada a objetivos, con criterios de entrada/salida y configuraciones priorizadas de los entornos.

### Automatización

El desarrollo de software ha visto como actividades de prácticas individuales, tales como pruebas funcionales y de desempeño, se automatizan. Y hoy en día los ingenieros de desarrollo utilizan código y herramientas de construcción que ahorran tiempo. Pero más necesidades a ser cubiertas para automatizar procesos y etapas, entre funciones para mejorar deficiencias de la organización, ahorran dinero y aceleran la velocidad de entrega al mercado (por ejemplo, entornos de prueba y reutilización del script de prueba entre líneas de negocios, suministro/análisis automatizado de laboratorio de pruebas, y traducción automatizada de modelos de caso de uso en entornos de pruebas).

### Control accionable

Los resultados de todo tipo de pruebas - incluyendo pruebas de unidad, funcionales, integración y escalabilidad - deben estar disponibles inmediatamente para el informe y análisis de tendencia. La integración de la información técnica del proyecto con análisis de negocios, da soporte a la influencia de los tomadores de decisión de alto nivel, en la distribución y uso de recursos para alinear TI y los negocios. La administración de pruebas y capacidades de planificación deben estar ajustadamente integradas en los análisis de requerimientos y procesos de definición. Los equipos de prueba comprometidos en la planificación temprana de entornos de pruebas, deben tener acceso en tiempo real a los negocios del proyecto y requerimientos funcionales, casos de uso, y acuerdos de nivel de servicio (SLAs). La integración de los datos del estado del proyecto con los análisis del negocio permite a los analistas de negocio comparar los cambios de proyecto con objetivos y restricciones de negocios. Evaluar dichos datos durante cada fase permite a los equipos de proyecto realizar ajustes y ponderar riesgos de prioridad. El resultado general proporciona un mejor control del proyecto y, cuando aparecen problemas, una más precisa evaluación del riesgo de negocio. La información correcta en el momento correcto, concentrada y filtrada para el análisis de decisión, produce una entrega de software controlada, y accionada.

---

**Destaque**

---

***Las líneas guía del proceso, mejores prácticas y herramientas integradas de equipo, son elementos esenciales en las soluciones de administración de calidad de hoy en día.***

**Soluciones de administración de calidad**

Las soluciones de administración de calidad de hoy en día proporcionan líneas guía del proceso de entrega de software, mejores prácticas y herramientas integradas de equipo, para ayudar a las organizaciones a alcanzar las mejoras en la eficiencia de costo, dimensiones de calidad y tiempo, para valorar los proyectos críticos de negocios.

Como se demuestra más arriba, la colaboración es clave. Las personas pueden colaborar utilizando el correo electrónico o formas acentuadas de compartir archivos, aunque estos son métodos muy inconvenientes, que ni siquiera son parte del diseño central de la entrega de software. Diversas organizaciones han desarrollado herramientas de administración colaborativa del ciclo de vida de la aplicación (ALM), que impulsan la transparencia del progreso del equipo, y gobiernan la forma en que dichos equipos trabajan. La plataforma Jazz™ incluye capacidades de ALM para dar soporte a la administración de requerimientos, actividades de desarrollo y administración de calidad, al mismo tiempo que ayuda a los equipos a sobreponerse a la infraestructura geográfica, y barreras de la organización para la colaboración. Además, centros coordinadores basados en la web, permiten a las personas trabajar juntas para entregar calidad duradera de software, como activo estratégico de negocios. Nuevas soluciones permiten a los usuarios:

---

### Destaque

---

**Las herramientas de administración de calidad presentan capacidades de colaboración, automatización e informe, ayudando a las organizaciones a entregar productos de más alta calidad – una y otra vez.**

- **Colaborar** entre negocios, equipos de desarrollo y pruebas con procesos dinámicos – y flujos de trabajo basados en actividad para la planificación y ejecución de pruebas.
- **Automatizar** procesos del ciclo de vida de mano de obra intensiva y captura temprana de cuestiones de calidad, reduciendo el tiempo de llegada al mercado, cortando costos, y mitigando los riesgos de negocios.
- **Reportar** métricas priorizadas adaptadas para individuos y equipos, facilitando una mayor visibilidad y permitiendo que los tomadores de decisión actúen con confianza.
- **Entregar** una mayor predecibilidad gracias al mapeo de patrones de implementación exitosos, a indicadores de desempeño operativos clave (KPIs).

Las compañías que aprovechan las nuevas herramientas de administración/definición de requerimientos serán capaces de:

- **Alinear** proyectos de desarrollo con objetivos de negocios, para reducir el riesgo de fallas del proyecto que cuestan a las organizaciones millones de dólares anualmente.
- **Impulsar** mejores respuestas y validación de requerimientos entre los negocios y expertos técnicos, utilizando técnicas de colaboración probadas y visuales (por ejemplo, bosquejos de procesos de negocio, borradores y paneles de discusión de interface del usuario, y utilizar entornos de prueba).
- **Administrar** cambio de los requerimientos y alcance del proyecto de forma más efectiva (por ejemplo, análisis de impacto).
- **Mejorar** el tiempo de llegada al mercado del proyecto, y retorno de la inversión, reduciendo el re-trabajo debido a requerimientos pobres y faltantes.

**Destaque**

*Las herramientas de ALM ayudan a construir y garantizar la calidad mediante el modelado de arquitectura, desarrollo, pruebas funcionales y del sistema, pruebas de validación automatizadas de la construcción, pruebas manuales, pruebas de carga y escalabilidad, y soporte de la producción de la aplicación.*

Como parte de una estrategia general ALM de calidad, las herramientas de entrega de software tales como las descritas en la tabla 2, también pueden ayudar a garantizar la calidad y desempeño de software crítico de negocios.

Herramientas	Descripción
<b>Modelo de arquitectura</b>	Valida las reglas definidas por el usuario que representan restricciones de arquitectura Automáticamente detecta patrones de diseño e importantes estructuras orientadas a objetos Detecta anti patrones estructurales (tales como nudos, centro concentradores y mariposas) que degradan el desempeño. Automáticamente refactoriza nudos mediante reparaciones rápidas
<b>Desarrollo</b>	Ayuda a los desarrolladores a detectar corrupción de memoria, detección de pérdidas, perfilar desempeño, y cobertura del código
<b>Prueba de validación de construcción automatizada</b>	Evita que construcciones con efectos sean implementadas en el laboratorio de pruebas o el entorno de pruebas del sistema Permite que los desarrolladores aprovechen los ciclos fuera de hora, para probar la estabilidad funcionalidad de una aplicación
<b>Prueba manual</b>	Promueve las mejores prácticas tales como modularidad y reutilización de pruebas, a equipos de transición, de pruebas manuales a automatizadas.
<b>Pruebas de sistema y funcionales</b>	Permite a los equipos construir pruebas que de forma manual o automática verifican la regresión y errores funcionales Acorta los ciclos de automatización de pruebas, mejorando la calidad mediante una cobertura de prueba más profunda y amplia Estimula pruebas más precisas, confiables y reproducibles
<b>Pruebas de carga y escalabilidad</b>	Determina umbrales de carga y escalabilidad de tecnologías y aplicaciones tales como Java™ Platform, Enterprise Edition (Java EE); Web (particularmente portales); SOA; Siebel; Oracle y SAP Garantiza que una aplicación de software pueda escalarse y ejecutarse, alcanzando las expectativas de usuario y SLA Permite a los realizadores de pruebas señalar el origen de cuellos de botella del desempeño, permitiéndoles navegar hasta el código fuente sin perder tiempo en atravesar múltiples niveles de código Permite un mejor retorno en inversiones de hardware, al ejecutar pruebas de planificación de capacidad de pre implementación, que miden los recursos de servidor necesario para alcanzar el desempeño y resultados deseados
<b>Soluciones de producción de soporte de aplicación</b>	Provee colaboración entre equipos de operaciones y desarrollo, para obtener un soporte de aplicación de loop cerrado, aislamiento y reparación de problemas Captura registro de transacciones e información de rastreo, así como también datos extendidos de recursos del sistema, para una determinación más granular del problema, y así reducir tiempo de respuesta de los diagnósticos de soporte de la aplicación Permite una restauración más rápida de los niveles de servicio a los negocios

Tabla 2: Las herramientas ALM contribuyen a la entrega de ciclo de vida de calidad.



## Resumen

En esta década hemos visto una colisión de tres vías, entre la exigencia de la transformación de negocios a través de la innovación, la necesidad de calidad mejorada del software para administrar riesgos de negocios, y la demanda en reducir costos para sobrevivir económicamente.

El cambio de la industria hacia el trabajo colaborativo en equipos entre equipos distribuidos geográficamente, acoplado con procesos y reportes automatizados en apoyo del control accionable, está cambiando el rostro de la entrega de software de calidad.

La administración de calidad ya no se refiere solamente a equipos de QA con herramientas automatizadas de prueba que dan soporte a la entrega de software tradicional; ni se trata acerca de procesos de control de administración de la calidad que demoran la entrega, o simplemente probar y reparar defectos. La administración de calidad se trata acerca de insertar calidad en un ciclo de desarrollo iterativo, y un programa de ejercicio productivo de software de loop cerrado, con soporte de herramientas integradas, datos y métricas rastreables. La administración efectiva de calidad hace que el proceso de entrega sea más manejable y menos traumático, y ayuda a construir confianza con los equipos de operaciones. Esencialmente, rompe el triángulo de hierro mencionado más arriba en este documento, debido a que permite a los administradores optimizar el alcance, costo y tiempos, al mismo tiempo de mejorar la calidad.

Activar la administración de calidad a través del ciclo de vida y detectar defectos tempranamente en el proceso, reduce costos y mejora la credibilidad. La comunicación constante a través de un conjunto de requerimientos comunes y bien entendidos – y respuestas rápidas a los cambios en dichos requerimientos – puede inyectar calidad en el desarrollo de software desde el primer momento. Además, el control diligente de la construcción y procesos de ejecución de pruebas, para corregir cursos y distribuir recursos, puede ayudar a las organizaciones a tornarse más flexibles, resolver regulaciones de conformidad, reaccionar más rápidamente a las cambiantes condiciones del mercado y, en definitiva, a impulsar el crecimiento de los negocios. En esta economía que no perdona, el éxito o el fracaso a menudo depende de una opción: cuál producto es el de la más alta calidad.

## Para obtener más informaciones

Para aprender más acerca de soluciones de administración de calidad de IBM, llame a su representante de IBM, o visite:

[ibm.com/software/rational/offerings/quality](http://ibm.com/software/rational/offerings/quality)

Un agradecimiento especial a Ron French de IBM Global Business Services por compartir sus hallazgos provenientes de su trabajo y el trabajo de su equipo.

## IBM Latin America HQ

One Alhambra Plaza  
Coral Gables, FL 33134  
U.S.A.

La home page de IBM se la puede encontrar en:

**ibm.com**

IBM, el logotipo IBM, ibm.com y Rational son marcas registradas de International Business Machines Corporation en los Estados Unidos, en otros países o en ambos casos. Si estos u otros términos de marcas comerciales IBM están marcados en su primera aparición en esta información con un símbolo de marca comercial (® o ™), estos símbolos indican que son marcas comerciales registradas o marcas bajo el derecho consuetudinario en los EE.UU. propiedad de IBM para el momento en que esta información se publicó. Tales marcas también pueden ser registradas o marcas según el derecho consuetudinario en otros países. Existe una lista actualizada de las marcas comerciales IBM disponible en la web en "Copyright and trademark information" en

[ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos, en otros países o en ambos.

Otros nombres de empresas, productos y servicios pueden ser marcas registradas o marcas de servicios de terceros.

Las referencias en esta publicación hacia productos o servicios IBM, no implican que IBM pretenda ponerlos a disposición en todos los países en los cuales IBM opera.

La información contenida en esa documentación se proporciona sólo con propósitos informativos. Pese al empeño para comprobar que la información contenida en esa documentación es precisa y completa, la información se proporciona "como está", sin garantía de ninguna clase, expresa o implícita. Además, esa información se basa en la estrategia y los planes de producto actuales de IBM, que están sujetos a cambios por parte de IBM sin aviso. IBM no será responsable de ningún daño que provenga de esa información o cualquier otra documentación o esté relacionado con ella en otra forma. Nada en el contenido de esa presentación tiene la intención o el efecto de crear garantías o representaciones de IBM (o de sus proveedores o licenciadores) ni de cambiar los términos y condiciones del contrato de licencia aplicable que rige el uso del software IBM.

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

<sup>2</sup> Walker Royce, administración de proyectos de software: A Unified Framework, Addison-Wesley Professional, Indianapolis, 1998.

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

© Copyright IBM Corporation 2006  
Todos los Derechos Reservados.