

SERVICIOS ESPECIALIZADOS

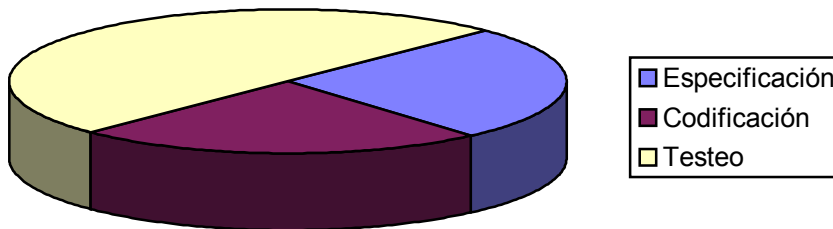
TESTEO



febrero, 2005

SERVICIO EXTERNO DE TESTEO

El control de calidad (SQA) y la prueba de los sistemas se ha transformado en una actividad crítica, especialmente en ambientes en los que, como el financiero, un error puede tener graves consecuencias para la organización. La actividad de prueba de sistemas se caracteriza por su complejidad en dichos entornos, donde se trata con varias capas de equipos no homogéneos, con sistemas operativos, bases de datos, sistemas de comunicaciones y herramientas de programación variados y heterogéneos, en los cuales una transacción recorre los diferentes ambientes on-line y batch, expandiendo sus efectos a través de muchos programas y archivos. La organización compartimentada de los recursos, conlleva a que cada especialista desconozca, a menudo, las consecuencias que sus decisiones pueden tener en el resto del sistema. Adicionado a ello, un error que desde el punto de vista de codificación puede ser relativamente simple de corregir, resulta muy difícil de detectar y puede tener efectos gravísimos en la organización, tanto en lo interno, como en su imagen con los clientes - el no cobro de una comisión del sistema de cuentas corrientes puede causar pérdidas económicas importantes,



Frente a ello, las áreas de sistemas se enfrentan a una serie de interrogantes, de los que remarcamos algunos de los más comunes:

- ¿Cómo optimizar el uso de los recursos humanos y equipos dedicados a las pruebas?*
- ¿Cómo asegurar la calidad de los resultados sin afectar el cronograma de implementación?*
- Y en relación a lo anterior, ¿Qué criterio utilizar para dar por finalizadas las pruebas?*
- ¿Cómo garantizar que el trabajo realizado no se pierda con cada nueva versión del sistema?*
- ¿Cuál es la relación óptima en cuanto al número de Testeadores versus Desarrolladores?*

posiblemente irrecuperables; el cobro erróneo de una comisión que no corresponde causa un daño no menor en la imagen de la entidad. Todo ello, sin mencionar los costos políticos para las gerencias de sistemas -. Simultáneamente, el equipo de desarrollo de las aplicaciones sufre continuamente presiones para cumplir fechas que a menudo son determinadas más por las necesidades del negocio que por consideraciones técnicas de factibilidad de cumplimiento, dejando de lado el hecho de que “nueve mujeres embarazadas de un mes no hacen un niño” . Como consecuencia de ello, el proceso de testeo se realiza en forma más o menos desordenada, teniendo como objetivo el tratar de que “las cosas funcionen”.

De allí surge la conveniencia de contar con profesionales especializados en estas tareas, comprometidos con la calidad de los productos, y que se dediquen a ello sin la presión de la coyuntura.

En éste sentido, es conocido que, en general, la proporción de las inversiones realizadas durante el ciclo de desarrollo de un sistema es similar a la indicada en la figura:

¿Que significa en términos de costos y tiempos la introducción de herramientas para el manejo y la automatización de las pruebas?

Apuntando a esos objetivos, son factores decisivos en el éxito de esa tarea:

- La experiencia y formación de los recursos humanos, tanto en Ingeniería de Sistemas como en el “Universo” que es objeto del aplicativo.
- La utilización de una metodología que ayude a sistematizarla, apuntando a ejecutar pruebas con una alta probabilidad de detectar errores.

- La administración controlada del proceso de instalación de nuevas versiones.
- El empleo de herramientas automáticas o semiautomáticas para cubrir un

amplio rango de situaciones ante cada nueva versión.

Y, aunque no suele tenerse en cuenta, un elemento **psicológico**:

Que los Testadores estén comprometidos prioritariamente con la detección de errores, y no con la “defensa” del producto.

Por otro lado, la realización metódica de esta actividad trae aparejadas una serie de mejoras debidas justamente a los factores psicológicos y metodológicos: El diseño se piensa y expresa con mayor precisión, los programadores se obligan a entregar programas mas elaborados, y la documentación, concebida como *herramienta* de este proceso, se realiza mas cuidadosamente. Hay datos y ejemplos de situaciones problemáticas en sistemas en los que no se tienen en cuenta estos factores.

Según el Standish Group de Maryland, USA, en una investigación, se obtuvieron los siguientes resultados:

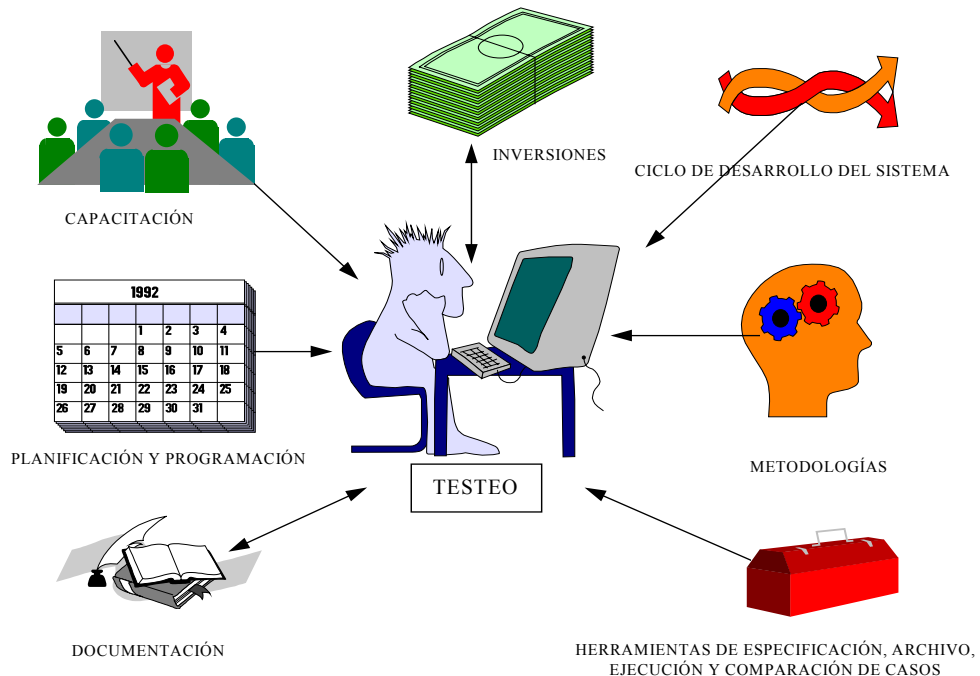
Proyectos	Europa	EEUU
Cancelados	10%	31%
En Duda	60%	53%
Analizados	70	8.380
Empresas	30	365

De acuerdo a un estudio de tres años de 70 grandes proyectos Cliente/Servidor en 30 firmas europeas, (Compuware, 12/95), la situación de las aplicaciones era la siguiente:

10% fueron abandonadas o recomenzaron de cero.
 20% sufrieron fuertes alteraciones para evitar el desastre.
 40% se liberaron con fuertes atrasos.

Por el contrario, de acuerdo a estudios realizados por el Software Engineering Institute y según una investigación del Corporate Computing Inc. Sobre 300 clientes, el resultado de la aplicación de mejoras en la construcción de sistemas se reflejan en:

28% mas de productividad en desarrollo.
 35% mas de productividad **de los usuarios finales**.
 19% menos en tiempo de desarrollo.
\$5 de retorno por cada \$1 invertido en la mejora del proceso de desarrollo.



Alcances del servicio

Cuando se trata el tema de **outsourcing en sistemas**, generalmente se piensa en la tarea de **programación**, partiendo de especificaciones construídas por los especialistas de la organización. Sin embargo, **justamente por los factores antes mencionados, la prueba de sistemas es un área especialmente apta para la contratación de servicios externos.**

En función de ello, SOLUCIONES RACIONALES S.A.C. ofrece este servicio, apelando al concurso de

profesionales especializados en esta actividad, con fuerte experiencia en el área financiera y comercial y con un background de implementaciones en instituciones de primera línea.

Las tareas se dividen en dos grupos, que adquieren mayor o menor relevancia de acuerdo al tipo y complejidad de sistema a implementar (sistema batch en un único computador, sistema en tiempo real, sistema distribuido en una red de equipos heterogéneos, etc.):

1. Preparación del entorno de testeo:

- Análisis del entorno (Equipamiento, red de comunicaciones, etc.)
- Especificación de herramientas para la automatización de las pruebas.
- Implementación de las herramientas.
- Capacitación.

2. Testeo propiamente dicho:

- Revisión de la documentación de diseño
- Definición del plan de prueba
- Diseño de casos de prueba según dicho plan
- Proceso de los casos de prueba definidos
- Registro y clasificación de los errores detectados
- Seguimiento y verificación de las correcciones
- Informe final de estado de cada subsistema

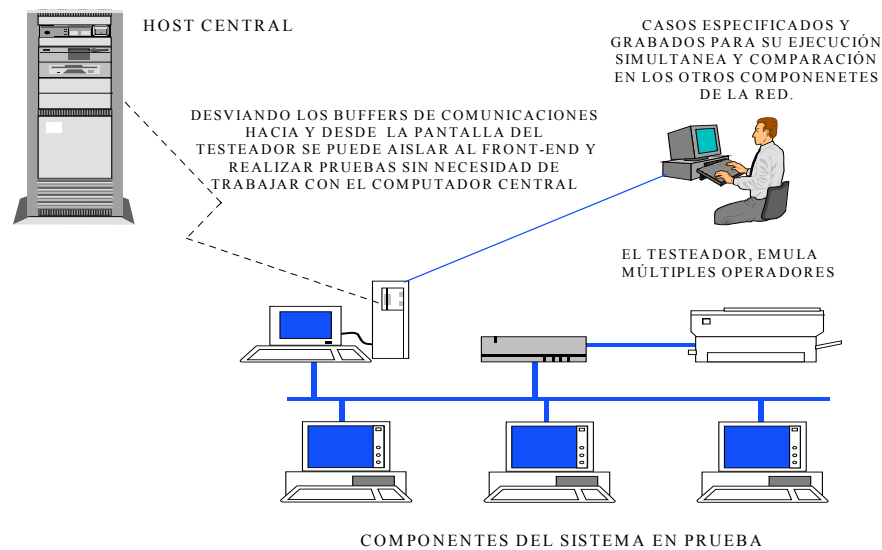
En el contexto de las mismas, se implementan las formas de **documentación** que se requieran (ej.: diseño de casos, registro de resultados, registro de errores), los **procedimientos** (ej.: recepción e instalación de nuevas versiones, criterios de aceptación o rechazo), los **instrumentos computacionales de soporte** (ej.: base de datos de seguimiento de errores) y **las herramientas automatizadas de testeo.** Todos estos aspectos se personalizan al entorno del aplicativo y facilitan y automatizan las actividades de prueba, dejando preparado el ambiente de testeo **para las próximas versiones de los programas.**

En definitiva, esto conlleva a la implementación de una metodología formal de testeo y control de calidad, que abarca no sólo la prueba del software, sino también aspectos como la administración de versiones, el control de la documentación de

desarrollo y la revisión de los manuales y ayudas para los usuarios finales.

Herramientas de testeo

Las herramientas disponibles en el mercado, graban las digitaciones del testeador y/o disponen de algún lenguaje especializado para la especificación de casos y resultados. De esta manera, el trabajo de diseño de casos realizado por el especialista se complementa con su transformación en archivos electrónicos, que luego podrán ser ejecutados repetitivamente, ante cada nueva versión de los programas, con facilidades para reflejar los cambios de versión en versión. En ese sentido, **SOLUCIONES RACIONALES SAC.** está en condiciones de utilizar las herramientas disponibles al usuario, en caso contrario puede usar sus propias herramientas.



Revisión de la documentación de diseño

Se estudian las especificaciones de los módulos a testear, consultando con los analistas a cargo de cada uno las modificaciones no incorporadas a las mismas. Si esas diferencias son menores, el testeador las volcará en los documentos, de lo contrario, se aconsejará la re-escritura por parte del analista.

Elaboración de un plan de prueba

Para cada subsistema, se establecerá el nivel de confiabilidad requerido, definiendo objetivos de la prueba en función del mismo. Se fijará el tiempo asignado a la tarea, acordándolo con el responsable del proyecto. Se establecerán pautas para las pruebas de módulos y de integración, estableciendo cuáles son las funciones críticas a las que se deberá dedicar mayor esfuerzo. En esta etapa se decidirá sobre qué módulos se efectuarán reuniones de revisión de código (structured walkthroughs).

Según el avance del proyecto y las características del subsistema, se definirá si se efectúan pruebas generales del sistema al culminar la etapa de integración y se determinará el plan para las mismas en caso afirmativo.

Diseño de casos de prueba según dicho plan

Se aplican técnicas de caja negra, tales como clases de equivalencia, análisis de valores límites o

gráficos de causa - efecto (en los casos que se justifique) y de caja blanca, con especificación del nivel de cobertura a alcanzar.

Los casos luego se vuelcan a archivos, de acuerdo con las herramientas a utilizar.

Proceso de los casos de prueba definidos

Se ejecutan las funciones del sistema partiendo de los datos especificados en los casos definidos y se confrontan los resultados obtenidos con los esperados, analizando también los posibles efectos colaterales indeseados.

Durante esta etapa se utilizan y / o construyen herramientas para la automatización de esta tarea, tanto en la realización de los casos como en la comparación de los resultados.

Registro y clasificación de los errores detectados

Los errores detectados son registrados en la base de datos, clasificados en distintos niveles según su gravedad. Se establece una escala de niveles de gravedad, en función del impacto de la aparición de ese error durante la operación del sistema, independientemente de la dificultad de su corrección o de las consecuencias sobre el diseño. La definición de la escala se hace en conjunto con el responsable del proyecto. A modo de ejemplo, mencionamos los siguientes criterios:

Nivel 1: Detalles (errores de ortografía en pantallas, estética de campos, alineación en impresos, etc.)
Nivel 2: Errores leves (errores en presentación de datos secundarios, no adecuación a estándares, comportamientos correctos pero diferentes en situaciones similares, dificultades de operación, etc.)
Nivel 3: Errores comunes (errores en documentos impresos que se entregan a personas ajenas a la organización, errores en presentación de datos, incumplimiento de objetivos en funciones secundarias, caídas de programas auxiliares, etc.)
Nivel 4: Errores graves (información crítica presentada erróneamente, información mal registrada en la base de datos, caídas de programas, incumplimiento de objetivos en funciones principales, etc.)
Nivel 5: Errores invalidantes (caídas de programas centrales, corrupción de la base de datos, imposibilidad de operar funciones básicas, errores en imputaciones que afecten cuentas monetarias, etc.)

El testeador asigna el nivel a su criterio. El error debe ser luego revisado por los responsables del sistema, quienes podrán rechazarlo (por interpretación incorrecta del testeador, condiciones inválidas de prueba, u otras razones), modificar el nivel y definir si debe corregirse o no y en qué momento, en función de las prioridades y fechas de instalación.

Esta categorización permite decidir mas objetivamente si determinada versión del sistema está o no en condiciones de ser liberada. Por ejemplo, puede resolverse instalar provisoriamente una versión que contenga errores de nivel 3, y postergar su reparación para después de la implementación.

Seguimiento y verificación de las correcciones

Por cada nueva versión entregada a testing, se repiten las pruebas necesarias (normalmente, se reiteran todos los casos de prueba para cada módulo modificado). Se definen casos de prueba específicos para verificar la corrección de los errores detectados anteriormente y se registra en la base de datos si fue efectivamente corregido o no. Se emiten informes periódicos estadísticos que reflejan el avance de las tareas. En las figuras siguientes se muestran ejemplos basados en casos reales a modo de ilustración.

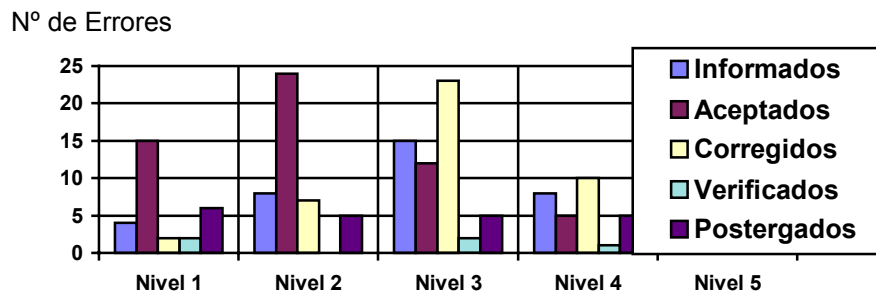
Errores detectados en el mes de Agosto:

	NIVEL 1	NIVEL 2	NIVEL 3	NIVEL 4	NIVEL 5	TOTAL
INFORMADOS	7	14	23	10		54
ACEPTADOS	17	33	37	9		96
CORREGIDOS		3	2	7		12
VERIFICADOS	2	3	3	9		17
RECHAZADOS	8	6	8	2		24
POSTERGADOS		2	6	3	1	9
TOTALES	34	61	79	40	1	212

Durante el seguimiento, los errores se van clasificando adicionalmente, acumulándose como Activos e Inactivos. Estos últimos son los que fueron rechazados por los Analistas y los que se han

subsanado. Un agrupamiento adicional permite verificar en que etapa de desarrollo (análisis, diseño, codificación, etc.) se origina cada error.

Errores Activos por Nivel al 31 de Agosto:



Informe final de estado del subsistema

Al terminar el último ciclo de prueba de cada subsistema se entrega un informe indicando si está o no en condiciones de ser instalado, una evaluación del riesgo asumido en la decisión de darlo por bueno y recomendaciones y/o advertencias acerca de acciones a tomar para resolver las situaciones provocadas por errores remanentes.

Equipo de trabajo y recursos

SOLUCIONES RACIONALES S.A.C. provee el equipo de especialistas de testeo, en función del tipo de aplicación y su entorno.

La empresa contratante asignará un analista para la interacción con los testeadores, durante las etapas de diseño de casos de prueba, procesamiento y análisis de errores detectados.

Para la elaboración del plan de prueba, se requerirá la colaboración del responsable del proyecto y/o el analista a cargo del subsistema, cuyas funciones principales serán la evaluación de la propuesta del

equipo de testing, la definición de prioridades y la identificación de funciones críticas.

Cada testeador ocupa, en general, un puesto de trabajo en el que se puedan ejecutar las funciones a testear, y con acceso a la base de datos en otra(s) ventana(s) para registrar los errores y hacer el seguimiento de la ejecución del programa.

La organización contratante podrá optar por incorporar personal propio al equipo de testeo.

Capacitación

Junto con las tareas inherentes al testeo en sí, y como complemento de este, SOLUCIONES RACIONALES S.A.C. ofrece el dictado de cursos sobre economía de las pruebas, su inserción en el ciclo de desarrollo, etapas, organización, planificación y criterios de finalización, métodos de diseño de casos, métodos de búsqueda del error e ideas de construcción y capacitación en el uso de herramientas, así como talleres de intercambio de experiencias y reflexión