

¿Pruebas, análisis y toma de decisiones?

En esta serie, Paul Gerrard, director de Gerrard Consulting, trata algunas de las cuestiones más cruciales del sector en relación con las pruebas. Paul indaga en la creación de informes de pruebas y los análisis que ayudan a tomar decisiones. ¿Cómo se pueden cuantificar las pruebas? Lo más fácil es sumar los costes, pero ¿cuántas pruebas hace falta realizar? ¿Cuál es, en definitiva, el valor de las pruebas? Estas preguntas tan difíciles son las que los evaluadores deben responder.

El análisis de pruebas es una disciplina que comprende todo el ciclo de vida, desde la idea hasta la retirada final pasando por el desarrollo y la producción e incluso más allá, empleando simplemente los datos que proporcionan las herramientas de gestión de pruebas.

Paul Gerrard
Gerrard Consulting

Con el patrocinio de



Introducción

Básicamente, el propósito de las pruebas es recabar información sobre ciertos aspectos de un sistema para poder tomar una decisión en función del resultado de una o varias pruebas.¹ Estas pruebas aportan la información más valiosa que precisan los desarrolladores (para corregir defectos), los directores de proyectos (para comprender y gestionar el avance) y los implicados (para estar al tanto y saber que todo marcha bien). A este respecto, todo lo que ofrecen las pruebas es poder: es la única fuente de información que permite conocer el progreso de los proyectos de sistemas.

En el ámbito móvil o digital, las empresas lanzan aplicaciones que obtienen información sobre los usuarios y los patrones de uso de las propias aplicaciones. Las aplicaciones de hoy recaban datos en tiempo real y los envían al proveedor igualmente casi en tiempo real. Estos datos se analizan para identificar tendencias, patrones de conducta, preferencias de los usuarios y oportunidades de mejora o nuevas iniciativas de mercado. Las aplicaciones se convierten en instrumentos para reunir información en función de las cuales poder tomar decisiones.

Introducción a los análisis de pruebas

En mi artículo, *Thinking Big: Introducing Test Analytics*² (Pensando en grande: Introducción a los análisis de pruebas) abogo por considerar la monitorización de la producción y las pruebas previas al lanzamiento, por un lado, y el análisis posterior al lanzamiento, por otro, como dos actividades que se deben realizar conjuntamente. Empleo el término “análisis de pruebas” como etiqueta para una disciplina que comprende todo el ciclo de vida desde la idea hasta la retirada final, pasando por el desarrollo y la producción. Hallará un completo estudio sobre las pruebas y los análisis de dispositivos móviles en *The Mobile Analytics Playbook*³ (El libro de estrategias de los análisis de dispositivos móviles).

Esta es mi definición de análisis de pruebas:

Obtención, integración y análisis de los datos de monitorización de las pruebas y la producción en función de las cuales tomar decisiones empresariales y de desarrollo de software informadas.

Por lo general, los evaluadores han tratado los datos de las herramientas de gestión de pruebas como la fuente principal de información con la que elaborar informes, pero este punto de vista es considerablemente limitado. También son interesantes los recuentos de casos de pruebas, la satisfacción de los requisitos (signifique lo que signifique), la comunicación, la categorización, la corrección o la omisión de informes de errores o de incidencias, y las series resumidas y puntuales, pero resultan igualmente limitadores.

Prácticas modernas: oportunidades para realizar pruebas

Los artículos que publiqué anteriormente de esta serie arrojaron algo de luz acerca de cómo podemos mejorar la elaboración de informes y los análisis. La mentalidad tras la técnica del desplazamiento a la izquierda, los modelos de pruebas, DevOps y la automatización, así como las disciplinas de pruebas emergentes a las que instan nos están obligando a replantearnos a qué nos referimos por informes de pruebas. El principal objetivo de la realización de pruebas (recabar y analizar datos para tomar decisiones informadas) no ha cambiado, pero existen muchas oportunidades para mejorar la forma en la que elaboramos informes de pruebas:

1. El propósito de la disciplina del desplazamiento a la izquierda es el de reducir los malentendidos (cuando no acabar con ellos) en los requisitos de forma anticipada. Posteriormente, ninguna crisis, defecto ni requisito costoso interferirá en nuestros informes.
2. El paso a la automatización masiva en los regímenes de DevOps genera automáticamente la mayoría de los datos que necesitamos. El resultado: la obtención y los análisis ya no son manuales; los informes se generan casi al momento.

3. La publicación *A New Model for Testing*⁴ (Un nuevo modelo de pruebas) sitúa el modelado en el centro del diseño de pruebas. Cuando los evaluadores emplean modelos significativos, disfrutan de unas mediciones de cobertura más significativas.
4. Los análisis de pruebas también se pueden obtener durante las sesiones de exploración. Se pueden crear modelos de uso y de sistemas antes de las sesiones de prueba y plantearlos como misiones, mientras que se puede obtener la cobertura de las pruebas durante las sesiones en los procesos.
5. La realización de las pruebas en la fase de producción es cada vez más frecuente. ¿El motivo? Nadie puede realizar pruebas de todos los tipos de dispositivos móviles (existen más de 24 000 dispositivos en el ecosistema de Android).⁵ El valor de los análisis de la monitorización de la producción es evidente.
6. Los productos de monitorización, registro, alertas y análisis se encuentran cada vez más a nuestro alcance. Las aplicaciones se convierten en instrumentos y en parte de la infraestructura. Lo que los evaluadores no pueden probar en el laboratorio se puede monitorizar (evaluar) sobre el terreno.
7. Algunas empresas están abandonando los procedimientos de gestión de incidencias burocráticas. Cuando se detecta algún defecto, se arregla.
8. Es el fin de las incidencias. La herramienta de control del código fuente y las pruebas automatizadas de unidades que promueven el cambio y lo protegen aportan la evidencia y nuevas visualizaciones del cambio y la repercusión. Eche un vistazo a la visualización del código fuente de *History of Python* (Historia de Python) para ver un ejemplo de lo que se puede lograr.⁶

Gracias a estas nuevas prácticas, técnicas, herramientas y visualizaciones, la naturaleza de las pruebas está cambiando. Contamos con una oportunidad real para mejorar nuestro producto clave: la información que ofrecemos a los implicados. Pero ¿en qué ayudan las pruebas a tomar de decisiones?

Pruebas y toma de decisiones

Durante todos estos años, he participado en muchos proyectos de pruebas de todos los tamaños. En cada uno de ellos, al final de una fase de pruebas llega el inevitable momento de tomar una decisión: ¿qué hacemos ahora? ¿Pasamos a la siguiente fase? ¿Lanzamos el producto a la producción o a los clientes? ¿Retrasamos el proyecto y ampliamos el plazo? ¿Nos detenemos para replantearnos los objetivos? ¿Abandonamos y cancelamos el proyecto?

Todas estas decisiones son de calado, pero las decisiones se toman en varios niveles. Por ejemplo, un evaluador que se encuentra con un problema y lo debate con un usuario y el desarrollador. Las propuestas surgen rápido: ¿Es un error? ¿Se puede arreglar? ¿Se puede arreglar rápido? ¿Qué impacto tendrá un cambio? ¿Existe una solución viable? ¿Merece la pena arreglarlo? ¿Le preocupa siquiera el problema al usuario?

Las pruebas respaldan la toma de decisiones en todos los niveles. Los implicados en las pruebas (ejecutivos, clientes, usuarios, gestores de proyectos, departamentos de operaciones y desarrolladores) pueden desempeñar varias funciones, por lo que la información que requieren de las pruebas varía según su perspectiva y la necesidad de tomar decisiones. No existe ninguna fórmula mágica para decidir qué información se precisa; debemos preguntar a los implicados qué consideran más útil, valioso y económico.

Sin embargo, existe un problema. ¿Cómo se pueden cuantificar las pruebas? Lo más fácil es sumar los costes, pero ¿cuántas pruebas hace falta realizar? ¿Cuál es, en definitiva, el valor de las pruebas? Estas preguntas tan difíciles son las que los evaluadores deben responder.

Soy un gran aficionado a la física y he utilizado por diversión algunos términos científicos para denominar un principio y dos teorías.

Pruebas funcionales a escala

Cuando evaluamos la funcionalidad de los componentes de los niveles superiores de la arquitectura, en especial los de los integradores y las aplicaciones, quizá debamos recurrir a una simulación con miles o millones de dispositivos sobre el terreno. Puede que la cantidad de combinaciones y permutaciones trascienda la capacidad de computación o predicción. Nuestras simulaciones generarán una y otra vez casos que se deben evaluar, registrarán los resultados y quizá vuelvan a reproducir las simulaciones para estudiarlas posteriormente.

Los componentes del nivel superior se deben poder evaluar. Necesitaremos ciertas herramientas como identificadores de excepciones y otras utilidades que inserten datos, obtengan y recreen o vuelvan a reproducir supuestos. Cem Kaner ha escrito bastante acerca de lo que él llama *High Volume Automated Testing* (Pruebas automatizadas con volúmenes elevados).⁷ Supone un buen punto de partida.

A estas técnicas también se las puede llamar “pruebas de grandes datos”. Para realizarlas necesitaremos lo siguiente:

- Hallar datos que sirvan para este propósito
- Generar, etiquetar, editar y contribuir con datos de serie para poder realizar el seguimiento de su uso
- Herramientas para monitorizar el uso de los datos etiquetados y capacidad para reconciliar los datos de la obtención, el almacenamiento, el uso y el desecho
- Nuevas herramientas de visualización de pruebas para respaldar los diagnósticos y la depuración de errores

Se trata de un juego de volúmenes. Las pruebas independientes pueden ser importantes o no, pero dedicaremos mucho tiempo a tratar con grandes volúmenes de resultados, visualizaciones y tomas de decisiones.

El principio de incertidumbre de las pruebas

Si alguna vez se ha planteado realizar “suficientes” pruebas y fijar una fecha de finalización, recordará lo difícil que resulta obtener una estimación fiable. Donde cabe esperar problemas (y lo que conforma la variable más importante en la ejecución de pruebas planificadas) es en la cantidad de pruebas que se van a realizar y cuánto reprocesamiento de pruebas se requerirá. Sin embargo, ello depende principalmente de la cantidad de errores que halle y lo difícil que resulte corregirlos y evaluarlos. No lo sabrá hasta que haya culminado el grueso de las pruebas, las correcciones y el reprocesamiento de las pruebas; otra de las paradojas de las pruebas.

El reto de la predicción y la planificación de pruebas se refleja claramente en el principio de incertidumbre de las pruebas:

- Podemos predecir el estado de una prueba, pero no cuándo finalizará.
- Podemos predecir cuándo finalizará una prueba, pero no su estado.

Puede definir un criterio de finalización o de salida (es decir, que todas las pruebas se ejecutan y resultan correctas), pero nunca sabrá con certeza si ha llegado a ese punto precisamente hasta que llegue a él. ¿Cuántas veces establecemos criterios de salida y nos cumplimos? Debemos tratar los criterios de salida como supuestos de planificación.

Si no cumplimos los criterios, nuestros supuestos serán incorrectos, nuestro plan no será correcto y deberemos volver a plantearnos el proyecto o su alcance.

Podemos establecer un plazo máximo para las pruebas y garantizar una fecha de finalización. Pero ¿quién sabe cuántas pruebas podemos realizar en ese periodo y si son bastantes?

La teoría de la relatividad de las pruebas

El valor de una prueba está condicionado por varios aspectos. Una única prueba puede ocupar cinco líneas de código en la prueba de la unidad de un componente. Otra puede ocupar cinco millones de líneas de código en un sistema grande. Es evidente que la segunda prueba tiene más valor. Pero ¿quién puede decir cuál es exactamente su valor?

Debemos plantear una pregunta menos ambiciosa. Puede que el evaluador tenga su opinión, pero debemos preguntar a los implicados para determinar el valor. No podrán asignar una calificación a las pruebas, pero normalmente podrán decirnos cuál de todas las pruebas realizadas es la que más valor tiene. A este respecto, no podemos asignar un valor absoluto, pero podemos tratar de discernir un valor relativo: una prueba tiene más o menos valor que otra. Suena como una gran desventaja, pero en realidad no lo es.

Lo más difícil de la planificación de pruebas se reduce al alcance. Sabemos que no podemos evaluarlo todo, así que debemos priorizar de alguna forma. Conocer el valor relativo de las pruebas significa que podemos seleccionar las más valiosas y dejar a un lado el resto de las pruebas de menor valor... De momento.

Solo los implicados en cuyo nombre realizamos las pruebas pueden determinar su valor.

Pero ¿qué hay de los riesgos de fracasar? ¿No está relacionado el valor de una prueba de alguna forma con los riesgos del modo de errores que la prueba está diseñada para abarcar? Puede que sí: la evaluación de un riesgo (cuantificado o no) es algo que se debe tener en cuenta en el valor de las pruebas. Pero aún nos queda otro problema.

Si tomamos una prueba valiosa, ¿posee el mismo valor una segunda prueba que hace algo parecido a la primera? Por lo general, no. Para debatir esto, debemos recurrir a la teoría cuántica.

La teoría cuántica de las pruebas

Cuando realizamos una prueba, obtenemos un resultado y lo comparamos con unas expectativas.

- Si el resultado coincide con las expectativas, satisface de forma incremental la necesidad o el requisito del usuario. La prueba aumenta de forma incremental nuestros conocimientos y nuestra confianza.
- Si el resultado no coincide con las expectativas, no satisface de forma incremental la necesidad ni el requisito del usuario. La prueba aumentará de forma incremental nuestros conocimientos, pero minará nuestra confianza (hasta que el sistema se haya corregido y el resultado de las pruebas sea satisfactorio).

Cada prueba genera un pequeño cuanto de evidencia. Un sí o un no, un verdadero o un falso, un 1 o un 0. Cuando se han realizado todas las pruebas, reunimos todos esos cuantos de evidencia y obtenemos una perspectiva más completa de la situación. (Por cierto, esto sucede en el bucle “aplicar prueba, interpretar resultado” de *A New Model for Testing*⁸).

Lo importante es que cuando ejecutamos una prueba, solo tiene valor si descubrimos algo que antes desconocíamos. Si lo que descubrimos es poco o nada, la prueba tendrá poco o ningún valor. Una prueba de valor es significativa porque nos aporta conocimientos acerca de un sistema sometido a una prueba. Cuando debamos determinar si dos pruebas son mejor que una, hemos de fijarnos en el valor potencial de la segunda prueba y su relevancia.

Puede que una segunda prueba presente valor por sí misma, pero si la situamos junto a otra casi duplicada, su relevancia y su valor real serán escasos. La relevancia de una prueba se aplica directamente al aumento incremental de cobertura que proporciona.

El evaluador (o al menos la persona que diseñó el modelo de prueba) es el mejor juez para decidir si una prueba es significativa. Para que alguna medida de cobertura sea relevante, siempre se debe medir con respecto a algún modelo de pruebas.

Resumen

Tras este debate tan científico, ¿qué debemos concluir?

Los emergentes enfoques de DevOps, la entrega continua y el desplazamiento a la izquierda brindan una oportunidad fantástica para reducir al mínimo el problema de los malos requisitos, trasladar la fase de pruebas a un momento anterior del proceso, y hallar y corregir problemas con más facilidad. No obstante, el creciente uso de la automatización también significa que recabar datos de los resultados de las pruebas será más fácil que nunca. ¿Cómo podemos aprovechar esta oportunidad?

Debemos crear modelos de pruebas que los implicados consideren significativos. De esta forma, podrán apreciar el valor de las pruebas que les presentamos. Como evaluadores y modeladores, podemos asesorarles acerca del significado de estas pruebas (la cobertura, si lo prefiere) con respecto a los modelos en sí mismos. De esta forma, podemos obtener pruebas de valor y reducir al mínimo la duplicación.

El debate acerca del valor y relevancia de las pruebas también nos ayuda a medir mejor el valor de nuestro sistema de automatización de pruebas. Una de las dificultades que ha observado en muchos de los proyectos de automatización de pruebas se ha dado cuando se trata de automatizar una prueba de un sistema completo. Cuando se ejecutan por primera vez como pruebas funcionales, adquieren valor. Sin embargo, cuando estas pruebas se ejecutan repetidamente como pruebas de regresión, su relevancia (y, por lo tanto, su valor) se reduce en gran medida.

El objetivo de lo que haríamos mejor en llamar una prueba antirregresión⁹ es detectar conductas no deseadas tras aplicar los cambios. Quizá solo haga falta dar la voz de alarma (por así decirlo) del 10 % de las pruebas de sistema. El otro 90 % cubre el mismo terreno y no son relevantes a este respecto.

Este artículo debería ayudarle a mantener debates mejor informados sobre la toma de decisiones, y a determinar la cantidad de pruebas y el valor de su labor como evaluador. Después de todo, puede que la teoría de la relatividad y la cuántica le sean de ayuda.

Acerca del autor

Paul Gerrard es consultor, profesor, escritor, *webmaster*, desarrollador, evaluador, conferenciante, entrenador de remo y editor. Ha realizado tareas de consultoría en todos los aspectos de las pruebas del software y el control de calidad, y se ha especializado en el control de pruebas. Ha presentado tutoriales y charlas destacados en congresos sobre pruebas en toda Europa, EE. UU., Australia, Sudáfrica y, en ocasiones, ha sido galardonado por ello.

Formado en la Universidad de Oxford y el Imperial College de Londres, Paul ganó en 2010 el premio Eurostar European Testing Excellence Award y, en 2013, fue galardonado con el European Software Testing Award (TESTA) en reconocimiento de su carrera profesional.

En 2002, Paul escribió *Risk-Based E-Business Testing* junto a Neil Thompson. Posteriormente, escribió *The Tester's Pocketbook* en 2009. Paul fue coautor del libro *The Business Story Pocketbook* con Susan Windsor en 2011 y escribió *Lean Python* en 2014.

En 2014, Paul fue el director de programa del congreso EuroSTAR Conference que tuvo lugar en Dublín.

Es el director de Gerrard Consulting Limited, de TestOpera Limited y anfitrión del Test Management Forum.

Correo electrónico: paul@gerrardconsulting.com

Twitter: @paul_gerrard

Sitio web: gerrardconsulting.com

Para obtener más información, visite **Desarrollo y pruebas** con CA Technologies.



Comuníquese con CA Technologies en ca.com/es



CA Technologies (NASDAQ: CA) crea software que impulsa la transformación de las empresas y les permite aprovechar las oportunidades que brinda la economía de las aplicaciones. El software se encuentra en el corazón de cada empresa, sea cual sea su sector. Desde la planificación hasta la gestión y la seguridad, pasando por el desarrollo, CA trabaja con empresas de todo el mundo para cambiar la forma en que vivimos, realizamos transacciones y nos comunicamos, ya sea en entornos distribuidos o de nube pública o privada, plataformas móviles o mainframe. Para obtener más información, visite ca.com/es.

Referencias

- 1 Paul Gerrard: *The Tester's Pocketbook*, <http://testers-pocketbook.com>
- 2 Paul Gerrard: *Thinking Big: Introducing Test Analytics*, octubre de 2013, <http://blog.gerrardconsulting.com/?q=node/630>
- 3 Julian Hartly y Antoine Aymier: *The Mobile Analytics Playbook*, <http://www.themobileanalyticsplaybook.com/>
- 4 Cem Kaner: *The Insapience of Anti-Automationism*, febrero de 2013, <http://context-driven-testing.com/?p=69>
- 5 Paul Gerrard: *A New Model for Testing*, 2012, <http://gerrardconsulting.com/?q=taxonomy/term/281>
- 6 Cory Goldberg: *History of Python*, visualización del desarrollo de Gource, junio de 2012, <https://www.youtube.com/watch?v=cNBtDstOTmA>
- 7 Paul Gerrard: *Regression Testing—What to Automate and How*, enero de 2010, <http://gerrardconsulting.com/?q=node/547>
- 8 Paul Gerrard: *A New Model for Testing*, 2012, <http://gerrardconsulting.com/?q=taxonomy/term/281>
- 9 Cory Goldberg: *History of Python*, visualización del desarrollo de Gource, junio de 2012, <https://www.youtube.com/watch?v=cNBtDstOTmA>