



Software Testing as Science

La Prueba del Software como Ciencia

Ingrid Galledic, George Killiospy

Centro Australiano de Excelencia para la Computación Cuántica y las Tecnologías de la Comunicación. *Galledic(AT)cqt.edu.au*

INFORMACIÓN DEL ARTÍCULO

Tipo de artículo: Reflexión

Historia del artículo

Recibido: 27-03-2013

Correcciones: 28-05-2013

Aceptado: 10-06-2013

Categories and Subject Descriptors

K.6.3 [Management of computing and information systems]: Software Management – *Software process.*

General Terms

Computer Science, Software Engineering, Software Testing.

Keywords

Software testing, scientific method, experiments, hypothesis.

Palabras clave

Prueba del software, método científico, experimento, hipótesis.

ABSTRACT

The most widespread opinion among people who have some connection with software testing is that this activity is an art. In fact, books have been published widely whose titles refer to it as art, role or process. But because software complexity is increasing every year, this paper proposes a new approach, conceiving the test as a science. This is because the processes by which they are applied are the steps of the scientific method: inputs, processes, outputs. The contents of this paper examines the similarities and test characteristics as science.

RESUMEN

La opinión más ampliamente difundida entre las personas que tienen alguna relación con la prueba del software es que esta actividad es un arte. De hecho, se han publicado libros de amplia difusión cuyos títulos se refieren a ella como arte, oficio o proceso. Pero debido a que la complejidad del software se incrementa cada año, en este artículo se propone un enfoque nuevo, concebir a las pruebas como una ciencia. Esto se debe a que los procesos mediante los cuales se aplican siguen los pasos del método científico: entradas, procesos, salidas. En el contenido del presente trabajo se analizan las semejanzas y características de la prueba como ciencia.

© 2013 IAI. All rights reserved.

1. INTRODUCCIÓN

La prueba de software ha sido descrita como un arte [1], un oficio [2], y un proceso [3], pero en este artículo se presenta un concepto un poco diferente: la prueba de software como ciencia.

En las comunidades de software comercial, las pruebas del software fueron vistas alguna vez como una idea de último momento. Los gerentes de producto novatos e incluso los desarrolladores las consideraban como una disciplina fuera de foco, que casi cualquier persona podía realizar. Incluso, algunos libros de divulgación sobre el tema incluyen en sus títulos palabras como *arte y oficio* [1,3], lo que puede haber llevado a algunas personas en la industria a la conclusión de que las pruebas para la calidad no eran una verdadera disciplina de la Ingeniería de Software. Por el contrario, la prueba efectiva es una disciplina de esta ingeniería y un componente crítico del ciclo de vida del desarrollo de software, cuyo objetivo es ayudar a mejorar la calidad y fiabilidad del producto mediante la identificación de defectos, la prevención de errores, y la observación y presentación de reportes.

Esencialmente, la prueba proporciona información para un análisis de riesgos racional y analítico [4].

En este trabajo se presenta una comparación entre los procesos de la ciencia y las pruebas de software. Es un análisis reflexivo en el que se busca mejorar la imagen de los procesos que se aplican para mejorar la calidad y fiabilidad de los productos software. Somos una sociedad que depende ampliamente del software, por lo que todos los aportes que ayuden a mejorar estos aspectos en sus productos deben tratarse y difundirse, porque el objetivo es que cada vez los usuarios puedan tener acceso a aplicaciones de mejor calidad.

2. EL MÉTODO CIENTÍFICO

El método científico tradicional ha sido el método predominante para las personas observar y comprender el funcionamiento del mundo y el universo. En los últimos años, algunos científicos han desarrollado métodos que son menos rigurosos [5], pero el tradicional [6] es el que se utiliza como base para este artículo. El algoritmo de este método es:

1. Observar algún aspecto del universo
2. Inventar una teoría que sea consistente con lo que se ha observado
3. Utilizar la teoría para hacer predicciones
4. Probar esas predicciones con experimentos o nuevas observaciones
5. Modificar la teoría de acuerdo con los resultados
6. Volver al paso 3.

Existen diferentes puntos de vista entre los científicos de hoy en cuanto a lo que constituye una teoría, una hipótesis y un hecho, y de cómo se definen estos términos depende su visión de la ciencia. Exponer plenamente los diferentes puntos de vista acerca del método científico y cómo se definen los términos es una cuestión que está más allá del alcance de este trabajo. Sin embargo, es importante entender que a menudo hay un nivel de sesgo, porque las personas aceptan las definiciones, acerca de cómo funciona el mundo, siempre que sean coherentes con sus creencias. Este es un razonamiento circular, porque si se trata de explicar cómo y por qué sucede algo, esto está basado en cierto grado en un marco en el que ya se cree [7], por lo que uno de los grandes retos de la ciencia es mantener la objetividad. Con el propósito de enmarcar las definiciones que se utilizan en este artículo a continuación se describen algunos términos, aunque puede ser que no sean aceptadas totalmente.

- **Observación.** Es el acto de reconocer y notar algún hecho u ocurrencia en la naturaleza, como una aurora, una corona, o la estructura de un animal. Específicamente, es el acto de medir, con instrumentos adecuados, cierta magnitud, como el tiempo de una ocultación con un reloj, la ascensión recta de una estrella con un instrumento de tránsito y un reloj, la altura del sol o la distancia a la luna desde una estrella con un sextante; la temperatura con un termómetro, etc.. Es la información así adquirida.

Nota: Cuando un fenómeno es analizado como una ocurrencia en la naturaleza, el acto se denomina una *observación*, y cuando las condiciones en las que se produce el fenómeno son artificiales, o han sido dispuestas de antemano por el observador, el proceso se denomina un *experimento*. El experimento incluye a la observación.

- **Experimento.** Es un ensayo u observación especial, hecha para confirmar o refutar algo dudoso; es un acto u operación realizada con el fin de descubrir algún principio o efecto desconocido, o para probar, establecer, o ilustrar alguna sugerencia o verdad conocida; es un examen práctico; es una prueba.
- **Hipótesis.** Es una suposición, una proposición o principio que se supone o se da por sentado, con el fin de llegar a una conclusión o inferencia para probar el punto en cuestión; algo que no se ha probado, pero que se asume para el propósito del argumento, o para contar un hecho o un acontecimiento, como la hipótesis de que los vientos de proa detienen un barco. En Ciencias Naturales, es una teoría tentativa o suposición adoptada provisionalmente para explicar

ciertos hechos, y para orientar en la investigación de otros, por lo que con frecuencia se llama *hipótesis de trabajo*.

- **Presunción.** Lo que supone; un postulado o proposición asumida; una suposición.
- **Teoría.** Es una doctrina o esquema de cosas, que termina en especulación o contemplación, sin una visión práctica; es una hipótesis; es una especulación. Es una exposición de los principios generales o abstractos de cualquier ciencia. Es ciencia a diferencia del arte. Es una explicación filosófica de los fenómenos, ya sea física o moral.
- **Hecho.** Es un efecto producido o logrado; cualquier cosa sucedida o que va a pasar; es un acto; es un evento; es una circunstancia. Es una realidad; una actualidad; una verdad. La afirmación o declaración de un acto realizado o existente
- **Ley.** En filosofía y física, es una regla de ser, operar o cambiar, con certeza o confirmación, algo que se concibe como impuesto por la voluntad de alguna autoridad de control, como la ley de la gravedad, las leyes del movimiento, las leyes del pensamiento, las leyes de causa y efecto, la ley de la auto-conservación. En matemáticas, es la regla según la cual nada procede. En las artes, el trabajo o los juegos, son las reglas de construcción, o de procedimiento, establecidas para lograr el éxito.

3. LAS PRUEBAS COMO CIENCIA

Algunos métodos de prueba se llevan a cabo al nivel de la *ciencia basura*, porque a menudo se basan en pequeñas muestras y experimentos mal controlados o documentados [8]. En el desarrollo de software esto se suele llamar *demo*, y se lleva a cabo para ejecutar el software con casos de prueba contruidos y que se sabe de antemano que funcionan. Por el contrario, las pruebas rigurosas se basan en la observación de la diferencia entre el comportamiento real y el comportamiento esperado del software bajo prueba (la hipótesis). Las pruebas se deben ver como Verificación (probar contra las especificaciones) y como Validación (probar contra el mundo real), y ambas son necesarias porque las especificaciones no siempre son perfectas.

Esto se debe a que el software es similar a una hipótesis científica, y ambos son inherentemente falibles. El marco básico del proceso de depuración del software es similar a la práctica del ensayo y error que insinúan las hipótesis científicas. El software es una simple conjetura técnica. A través de pruebas rigurosas los ingenieros de prueba tratan de refutar la presunción de un software sin defectos, destinadas sobre todo a demostrar la existencia de los mismos (falsificación). El proceso de falsificación contrasta con los enfoques basados en datos, como la inducción y la deducción, que intentan justificar la validez basada en la repetición de los hechos. La mayoría de enfoques justificativos tratan de soportar las peticiones a través de la confirmación, y de esta forma *confirman* ante

todo los esfuerzos para validar la incertidumbre sin errores utilizando datos específicos que demuestren un funcionamiento correcto [9]. Este enfoque claramente demuestra que el software sólo funciona correctamente bajo ciertas condiciones, pero no prueba que el software está libre de errores. Los aspectos científicos de las pruebas del software son:

3.1 Pre-definición de los resultados esperados

Pre-definir los resultados esperados en la prueba es similar al científico que predice el resultado de un experimento, antes que se lleve a cabo, mediante la propuesta de una hipótesis. Hay algo acerca de predecir el resultado de antemano que añade un grado de rigor a los hallazgos: si se espera hasta que el experimento haya terminado para tratar de interpretar los resultados, a la luz de la comprensión y la observación, es fácil convencerse a sí mismo y a los demás de que lo observado fue una validación de la hipótesis, porque después de todo se consideraron todas las cosas. Cuando los resultados reales del experimento no coinciden con los resultados esperados pre-definidos, la discrepancia puede llevar a cuestionar el experimento, la hipótesis, o a ambas.

3.2 Observación

Sin la observación es imposible determinar el resultado de una prueba o experimento. Aunque esto tiene sentido, es tentador diseñar pruebas y experimentos que sean difíciles, si no imposibles, de observar. Es posible que se desee comprobar o probar algo, pero las limitaciones del mundo real dificultan la construcción de un experimento exacto. Es por eso que no se puede probar todo, porque no todo es comprobable.

3.3 Repetitividad

En ciencia, un experimento se puede realizar miles de veces antes de poder establecer una tendencia. La primera vez que un científico observa un resultado no está seguro de si se debió a un aspecto desconocido del experimento o a un comportamiento predecible del objeto, por lo que para proporcionar una confirmación del experimento puede ser necesario repetirlo muchas veces. Del mismo modo, cuando en las pruebas se observa un defecto, la primera prueba puede ser vista como el indicador y las siguientes como la confirmación. Después que un defecto ha sido corregido, la prueba se debe repetir exactamente como antes de encontrarlo, para así asegurar que se ha superado. Aunque esto suena simple, en la práctica puede ser muy difícil obtener un segundo entorno de prueba configurado exactamente como en el primero.

3.4 Construir el Experimento

En la investigación científica, los experimentos son cuidadosamente planificados y controlados. El ambiente del laboratorio surgió de la necesidad de probar las condiciones durante un experimento y poderlo repetir nuevamente. En esta analogía de las pruebas como ciencia, muchas personas parecen realizar los experimentos en la *cocina*, no en un entorno de laboratorio controlado. Esta es una cuestión crítica, porque el entorno de la prueba puede afectar muchos

factores externos e internos de la misma, que podrían dar lugar a resultados erróneos. Se puede afirmar que en ninguna otra disciplina se podrían hallar resultados con los métodos simples que muchos utilizan en las pruebas del software, especialmente cuando se trata de los entornos [10]. En las pruebas, a veces se necesitan entornos cuidadosamente contruidos y controlados para obtener el nivel de confiabilidad adecuado al riesgo. ¡La prueba es tan buena como el entorno de la misma!

3.5 Realizar el experimento

La realización del experimento es un ejercicio en el que se sigue cuidadosamente el diseño previo. El científico investigador no improvisa a menos que esté trabajando por fuera del plan. Por supuesto, algunos de los grandes descubrimientos científicos se han producido porque el investigador intentó algo más que no estaba planeado, pero son excepciones más que reglas. En las pruebas, es importante apegarse a al plan de pruebas. Está bien probar otros casos, pero se debe documentar el proceso para, si es necesario, poderlo repetir.

3.6 El grupo de control

En los experimentos científicos, los grupos de control se utilizan como una línea base para comparar los resultados. Por ejemplo, un investigador podría probar un fármaco de prueba en un grupo de personas, mientras que administra un placebo a otro grupo. Las personas en el experimento no saben si se les ha suministrado el medicamento real o el placebo. Esta investigación *dobles ciego* ayuda a contrarrestar los prejuicios subconscientes. En la prueba también se tiene como referencia una línea base del funcionamiento correcto del sistema. Interpretar los resultados y llegar a conclusiones es uno de los grandes desafíos de la ciencia, y consiste en observar la prueba de una hipótesis y hacer una interpretación razonada de los resultados. El reto de hacer esta tarea es mantener la objetividad y tener el valor de reportar lo que realmente se observó, como algo contradictorio a lo que se esperaba ver. Eso suena bastante familiar para algunas ciencias [11]. En las pruebas, sólo se puede hablar de lo que se observa, y no es realista ni prudente predecir resultados de lo que podría ser visto a partir de pruebas no realizadas.

3.7 Modificar la hipótesis

A menudo la hipótesis principal en las pruebas es que el sistema debe funcionar en determinadas condiciones. Sin embargo, existe otra hipótesis opuesta según la cual, aunque el sistema funciona, necesariamente no quiere decir que no tiene defectos. La segunda hipótesis es la más segura. Cuando los resultados de las pruebas demuestran que la segunda hipótesis es verdadera, entonces comienza a producirse un cambio fundamental en las mentes y actitudes de las personas que trabajaron con la primera hipótesis. Esto sucede cuando, en lugar de modificar la hipótesis, se trata de desacreditar o invalidar el experimento, y a menudo consiste en culpar a los probadores de los defectos, que es lo mismo que culpar a un científico por los resultados de la investigación, aunque haya llevado a cabo el experimento correctamente.

Sin embargo, suponga por un momento que las personas llegan a un acuerdo de que la primera hipótesis era incorrecta y que el software tiene defectos que necesitan ser determinados. Esto puede implicar que se entregue el software y esperar a que otros tengan que rendir cuentas. Aunque se puedan producir estas consecuencias, las personas tienen que enfrentar la realidad y corregir los problemas en lugar de centrarse en sus propias agendas. Tal vez este aspecto de la prueba está más estrechamente alineado con la ciencia que se practica actualmente [4]. Si la investigación confirma la hipótesis, se hace amplia difusión del suceso, pero si la investigación confirma la hipótesis contraria, especialmente una que resulta ser políticamente incorrecta, los resultados pueden no ser publicados.

3.8 Visión a largo plazo

La razón por la que se realiza investigación científica es para explicar la forma en que se comporta la naturaleza observable. Se ha dicho que lo que distinguía a Thomas Edison de otros inventores era que siempre tenía un agudo sentido de cómo la ciencia podía ayudarles a las personas a mejorar sus vidas. También tenía un buen sentido de los negocios, y sabía cuándo dejar de investigar para construir el proyecto. En las pruebas, el objetivo inicial es encontrar defectos, sin embargo, esta es una visión miope y desaprovecha los recursos que han sido invertidos en la creación y fijación del defecto. La visión a largo plazo de la prueba consiste en construir formas para evitar problemas similares en el futuro, mediante la mejora de los procesos utilizados para fabricar el producto. Al parecer estamos lejos de ver las pruebas como un proceso científico, pero nos da algo para pensar y relacionar, especialmente cuando se trata de evaluar el rigor y la fiabilidad de los resultados de las pruebas.

3.9 Creatividad

Aunque algunos pueden considerar a las pruebas de software como un arte, la única similitud es que implica una gran dosis de creatividad. Se podría argumentar que la prueba de un programa computacional a gran escala requiere más creatividad que su diseño. Myers [1] respalda esta teoría al afirmar que la prueba es una tarea extremadamente creativa e intelectualmente desafiante. De hecho, muchas personas que diseñan y desarrollan software admiten que se encuentra más satisfacción estética en la creatividad de las pruebas.

3.10 Visión filosófica

Desde el principio de los tiempos, los seres humanos han participado de las pruebas en diversas formas, ya fuera al cuestionar los diseños de herramientas o al criticar los procesos. Esta constante re-evaluación dio lugar a una mejor calidad y a ganancias significativas en eficacia y eficiencia. Pero la prueba basada puramente en ensayo y error es laboriosa y lenta. Karl Popper [12] aborda el crecimiento del conocimiento desde un proceso más científico, y plantea la hipótesis de que la ciencia progresa mediante la reducción de la falsedad. Esencialmente, el progreso se logra mediante la conversión de teorías (conjetura - *razonamiento que implica la formación de conclusiones a partir de pruebas incompletas*) en hipótesis

comprobables (refutaciones - *alguna evidencia que ayude a establecer la falsedad de algo*).

3.11 Establecer confianza

La mejor forma para establecer la confianza en la fiabilidad del software no es sólo demostrar que funciona dentro de los límites de la especificación del producto, sino también al exponer los defectos potenciales en el diseño o la funcionalidad del mismo. Sin embargo, cualquier conjetura puede consumir muchas horas de exhaustivas pruebas y todavía contener errores. La refutación no garantiza un software impecable, porque simplemente hay demasiadas hipótesis para poner a prueba en un plazo razonable de tiempo. William Hetzel [13] explica la futilidad de las pruebas exhaustivas y las limitaciones de la refutación, y opina que existen más casos de prueba en un programa con 70 funciones que cucharaditas de agua en el Océano Pacífico. Cuando un probador refuta con éxito el software (encuentra un error), informa el defecto; entonces, el desarrollador (con suerte) resuelve el problema, por lo que se supone que el software funciona correctamente, y el proceso de refutación se repite. Aunque la prueba no puede demostrar que el software está libre de errores, detectar errores demuestra problemas, lo que en última instancia se traduce en mayor fiabilidad y calidad del producto, o al menos ofrece una perspectiva fundamental para el análisis y la mitigación de riesgos.

4. QUÉ HACE REALMENTE UN PROBADOR

Cuando se le pide a los ingenieros de pruebas que describan su trabajo, la mayoría responde: *quebrar el software*. Tal vez por eso algunas personas consideran erróneamente al papel del probador como un proceso *destrutivo*. Sin embargo, los probadores realmente no *rompen* software, simplemente exponen los errores que ya existen, al refutar o falsear el supuesto de que el software es impecable a través de los resultados que observan, luego de aplicar pruebas estructuradas, de deducción lógica y de experimentos. Sin embargo, los errores que encuentran sólo son un aspecto de las pruebas; los ingenieros de pruebas realmente lo que hacen es agregar valor al proceso de desarrollo mediante la validación del software, la detección de errores, proporcionándoles a los desarrolladores información para evitar errores y a la administración para evaluar el riesgo, y actuando como defensores de los clientes. Entonces, ¿qué cualidades debe tener un gran probador de software? James McCaffrey [] las reduce a ocho:

1. Pasión por el análisis y las pruebas
2. Habilidades técnicas
3. Capacidad intelectual pura
4. Capacidad para priorizar y organizar
5. Capacidad de adaptación y aprendizaje
6. Capacidad para trabajar sin supervisión directa
7. Capacidad para comunicarse
8. Capacidad para comprender la estrategia de negocios

Como los sistemas software aumentan en complejidad, el valor de las pruebas de software se está incrementando y convirtiéndose en una actividad cada vez más importante.

5. CONCLUSIONES

Existen muchos puntos en común entre las pruebas del software y la ciencia tradicional. De hecho, las primeras pueden estar más cerca de la ciencia que cualquier otra cosa que podamos relacionar. Estas similitudes pueden ser útiles para su comprensión y para explicarlas a otros. Las similitudes también proporcionan un punto de referencia de la rigurosidad del proceso que se utilizan en la definición y realización de los procesos de prueba. Aunque no es necesario que cada prueba se realice con el rigor de la investigación científica, algunas sí requieren ese nivel, especialmente las de productos de alto riesgo.

Al igual que en la ciencia, en las pruebas de software se buscan respuestas correctas. O bien el producto cumple con los requisitos o no lo hace. A través de pequeños pasos incrementales las pruebas se aproximan a la *verdad* sobre el software. El objetivo es entregar el software con el menor número de errores posible, para que el usuario reciba un producto fiable y de calidad.

6. REFERENCIAS

- [1] Myers, G. (2004). [The Art of Software Testing](#). John Wiley & Sons.
- [2] Marick, B. (1995). [The Craft of Software Testing - Subsystem Testing Including Object-Based and Object-Oriented Testing](#). PTR Prentice Hall
- [3] Perry, W.E. (2006). [Effective Methods for Software Testing - Includes Complete Guidelines, Checklists, and Templates](#). Wiley.
- [4] Lensmar, O. (2013). [The new science of software testing](#). Online [Feb. 2013].
- [5] Lehrer, J. (2010). [The Truth Wears Off - Is there something wrong with the scientific method?](#) Online [Mar. 2013].
- [6] Platt, J.R. (1964). [Strong Inference - Certain systematic methods of scientific thinking may produce much more rapid progress than others](#). Science, 146(3642), pp. 347-353.
- [7] Chomsky, N. (2011). [How the World Works](#). Soft Skull Press.
- [8] Agin, D. (2006). [Junk Science - How Politicians, Corporations, and Other Hucksters Betray Us](#). Thomas Dunne Books.
- [9] Cerrosen, J. & Pong, J. (2012). Science in Computational Sciences. RACCIS, 2(2), pp. 12-20.
- [10] Eickehnann, N. & Richardson, D. (). [An Evaluation of Software Test Environment Architecture](#). In Proceedings of the 18th international conference on Software engineering ICSE'96 (Berlin, Germany, March 25-26), pp. 353-364.
- [11] Serna, M. E. [Social Control for Science and Technology](#). In Larrondo, M. et al. (Eds.), Proceedings 10th Latin American and Caribbean Conference for Engineering and Technology LACCEI'12, (Panama City, Panama, July 23-27), paper 35.
- [12] Popper, K. (2002). [Conjectures and Refutations](#). Routledge.
- [13] Hetzel, W. (1988). [The Complete Guide to Software Testing](#). QED Inforiation Sciences.
- [14] McCaffrey, J. (2008). [What Makes A Good Software Tester?](#) Online [Jan. 2013].