CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Aplicación de la Tabla Ortogonal en el diseño de los Casos de prueba de Software.

Application of the orthogonal arrays in the design of the Cases of test of Software

Ing. Ángel Eduardo Pentón Saucedo.
Empresa Comercializadora de Combustibles Matanzas.
Matanzas, Cuba.
eduardo@ecmtz.cupet.cu
Dr. Alfredo del Castillo Serpa.
CEIS, Centro de Estudios de Ingeniería y Sistemas. CUJAE
La Habana, Cuba.
acastillo@mecanica.cujae.edu.cu

Resumen
En la actualidad es de vital importancia desarrollar aplicaciones informáticas con disimiles funcionalidades y alta complejidad de forma rápida y eficiente en todas las esferas. Realizar pruebas al software durante el proceso de desarrollo es garantía para la puesta en explotación de los sistemas; además de corroborar el grado de confiabilidad antes de ser entregado a sus usuarios finales; disminuyendo los defectos al utilizar técnicas apropiadas que posibiliten procesos de desarrollo de software eficaz, minimizando tiempo y costo. El objetivo de este trabajo es aplicar la propuesta de la tabla ortogonal como alternativa para la mejora continua de las estrategias de pruebas y la disminución del tiempo requerido en su ejecución. Al evaluar la literatura sobre pruebas robustas basadas en arreglos ortogonales se enfatizó en las técnicas empleadas por el Dr. G. Taguchi para el mejoramiento de calidad de los productos y procesos; aplicando las mismas en los procesos ingenieriles para el desarrollo de software. Planificar y ejecutar los casos de prueba con la propuesta anterior garantiza detectar el mayor número de errores en las interacciones dobles y triples de las funcionalidades del sistema.
Palabras clave: Diseño Ortogonal, Cuadrados Latinos, Confiabilidad, Sistema Software, Estrategias de prueba.

Abstract
At the present time it is of vital importance to develop computer applications with dissimilar functionalities and high complexity in a quick and efficient way in all the

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

spheres. To carry out tests to the software during the development process is a guarantee for the setting in exploitation of the systems; besides corroborating the grade of dependability before being given to their final users; diminishing the defects when using appropriate techniques that facilitate processes of development of effective software, minimizing time and cost. The objective of this work is to apply the proposal of the chart orthogonal arrays like alternative for the continuous improvement of the strategies of tests and the decrease of the time required in its execution. When evaluating the literature on robust tests based on arrangements orthogonal arrays it was emphasized in the techniques used by the Dr. G. Taguchi for the improvement of quality of the products and processes; applying the same ones in the processes engineering for the development of the software. To plan and to execute the cases of test with the previous proposal guarantees the detection of the biggest number of errors in the double interactions and triples of the functionalities of the system.
Key words: Design orthogonal arrays, Latin Squares, Dependability, System Software, test Strategies.

Introducción

En la actualidad se necesita disponer de programas de computación en tiempos verdaderamente record, es evidente que debemos diseñar y construir software más potente y confiable que garanticen las expectativas de los usuarios y sea fácil de usar y mantener, al mismo tiempo que se minimicen los plazos para la ejecución de prueba y se alcancen mejores parámetros de confiabilidad. Existen en la actualidad variadas formas y métodos de realizar pruebas al software con el objetivo de asegurar programas confiables y disminuir el número de errores que pueden producirse durante la utilización del software; además de otros aspectos que se deben tener en cuenta en toda la fase de desarrollo al utilizar metodologías de prueba que hagan posible la entrega en tiempo del sistema al cliente o usuario final. Las pruebas del software son un elemento crítico para la garantía de calidad del software y requieren presentar una revisión final de las especificaciones del diseño y de la codificación (Hetzel, 2001). La creciente percepción del software como un elemento del sistema y la importancia de los “costes” asociados a un fallo del propio sistema, está motivando la creación de pruebas minuciosas y bien planificadas según Hetzel (2001). Lo que trae consigo investigar los mejores métodos de pruebas para disminuir el tiempo empleado *(las organizaciones por lo general deben emplear entre el 30 y el 40 por ciento del esfuerzo total de un proyecto en las pruebas)* en realizar las pruebas al software (Pressman, 2003).
En este trabajo se expone la posibilidad que nos brinda el método de la tabla ortogonal para planificar el número de pruebas eficiente con el objetivo de minimizar el tiempo e incrementar la confiabilidad del sistema en construcción.

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Análisis crítico de propuestas existentes:

El diseño de pruebas para el software o para otros productos de ingeniería requiere tanto esfuerzo como el propio diseño inicial del producto aseveró Pressman (2010). Cualquier producto de ingeniería es probado de una de estas formas: Conociendo la función específica para la que fue diseñado el producto, se llevan a cabo las pruebas que demuestren que cada función es completamente operativa, al mismo tiempo se buscan errores en cada función; además conociendo el funcionamiento del producto, se desarrollan pruebas que aseguren que todas las piezas encajan correctamente, o sea, que la operación interna se ajusta a las especificaciones y que todas las componentes internas se han comprobado de forma adecuada (Jacobson, 2003). *Los investigadores del tema teniendo en cuenta estos criterios han desarrollado dos técnicas fundamentales: las pruebas de caja blanca o caja de cristal y la prueba de caja negra o prueba de comportamiento* (Pressman, 2010).
La prueba de caja blanca del software está basada en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proporcionando casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se examina el estado del programa en varios puntos para determinar si el estado inicial coincide con el esperado o mencionado, a primera vista parecería que una prueba de caja blanca muy profunda nos lleva a tener programas cien por ciento correctos. *Todo lo que se tiene que hacer es definir todos los caminos lógicos, desarrollar casos de pruebas que ejerciten y evalúen los resultados; es decir, generar casos de pruebas que ejerciten exhaustivamente la lógica del programa establecidas por* Taguchi (2009).

Al considerar el software de computadora, la prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, o sea, los casos de pruebas demuestran que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Según la literatura consultada los clásicos consideran que las pruebas de caja negra se centran en los requisitos funcionales del software; o sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca establecido por Pressman (2010). Más bien se trata de un enfoque

complementario que intenta descubrir tipos de errores diferentes a los métodos de caja blanca.

La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a base de datos externas, errores de rendimiento y errores de inicialización.

En las técnicas de caja negra se presenta una prueba denominada prueba de la tabla ortogonal conocida en la literatura como prueba robusta basada en arreglos ortogonales la cual es interesante en la planificación de pruebas; la misma será objeto de estudio en este trabajo por ser una de las menos empleadas y resultando útil al ingeniero de pruebas durante la etapa de prueba, fundamentalmente antes de la puesta en explotación del sistema. Hay muchas aplicaciones en las que el dominio de entrada es relativamente limitado. Es decir, el número de parámetros de entrada es pequeño y los valores de cada uno de los parámetros están claramente delimitados. Cuando estos son muy pequeños (por ejemplo, tres parámetros de entrada tomando tres valores diferentes), es posible considerar cada permutación de entrada y comprobar exhaustivamente el proceso del dominio de entrada. En cualquier caso, cuando el número de valores de entrada crece y éste es diferente para cada elemento de dato, la prueba exhaustiva se hace imposible (Pressman, 2010).
La prueba de la tabla ortogonal se aplica a problemas donde el dominio de entrada es relativamente pequeño pero demasiado grande para posibilitar pruebas exhaustivas. El método de prueba de la tabla ortogonal nos permite encontrar errores asociados con fallos localizados. *PHADKE (2005) valora el resultado ortogonal de la siguiente manera: Estas pruebas detectan y aíslan todos los fallo de modalidad simple (un fallo de modalidad simple es un problema que afecta a un solo parámetro); detecta todos los fallos de modalidad doble (un fallo de modalidad doble es en el que están afectados los parámetros que intervienen conjuntamente); además pueden asegurar la detección de fallos de modalidad múltiple. Concluyendo que un arreglo ortogonal tiene la propiedad de balance, esto es, para cada parámetro (par de columnas) todas las combinaciones de parámetro-nivel ocurren la misma cantidad de veces* (Jacobson, 2003).

Las pruebas de software planificadas con arreglos ortogonales son basada fundamentalmente en utilizar criterios de diseños de expertos cuyo objetivo es optimizar la cantidad de pruebas a realizar para lo que se pueden utilizar software o tablas que están disponibles para la identificación del número de pruebas a ejecutar teniendo el probador (ingeniero de pruebas) la tarea de determinar el nivel

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

y tipo de prueba en base al orden de las iteraciones, tipo de proyecto, experiencias anteriores y otros estudios de casos reportados en la literatura (Pressman, 2009).

Los métodos de Taguchi son técnicas estadísticas para realizar experimentos que pueden determinar las mejores combinaciones de variables de productos y procesos para fabricar o desarrollar un producto. *El método del Dr. Taguchi para el diseño de experimentos utiliza técnicas que implican bajos costos y que son aplicables a los problemas y requerimientos de la industria moderna* (Taguchi, 2009). El propósito que se tiene en el diseño del producto es encontrar aquella combinación de factores que nos proporcione un desempeño más estable y costo de desarrollo más bajo. *Taguchi (1992) valora la ventaja fundamental de los arreglos ortogonales es que pueden ser aplicados al diseño experimental involucrando un gran número de factores*. Es muy frecuente que a la hora de diseñar un producto tengamos múltiples variables (*FACTORES*) a tener en cuenta. Cada uno de estos factores toma distintos valores (*NIVELES*) y es necesario elegir el más conveniente, sin embargo, cuando el número de factores y de niveles es elevado, el número de combinaciones posibles es elevado y el número de experimentos a realizar sería muy costoso (Taguchi, 2007).

En general, para un arreglo a dos niveles, el número de columnas (efectos o factores) que se pueden analizar, es igual al número de renglones más uno. Taguchi ha desarrollado una serie de arreglos para experimentos con factores a dos niveles, los más utilizados y difundidos según el número de factores a analizar son relacionados en la tabla No. 1 (Taguchi, 2005):

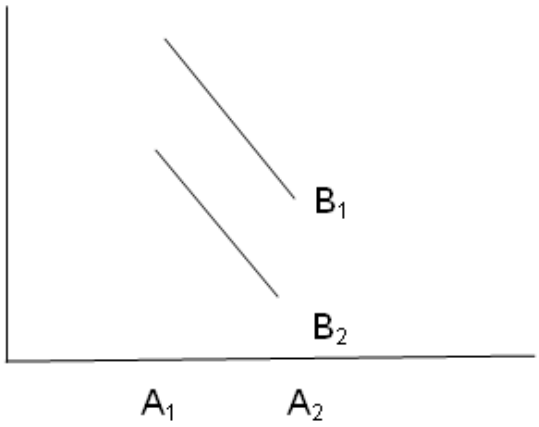
No de factores	Arreglo a utilizar	No. de condiciones a probar
Entre 1 y 3	L4	4
Entre 4 y 7	L8	8
Entre 8 y 11	L12	12
Entre 12 y 15	L16	16
Entre 16 y 31	L32	32
Entre 32 y 63	L64	64

Tabla No. 1: Serie de arreglos para experimentos con factores a dos niveles.

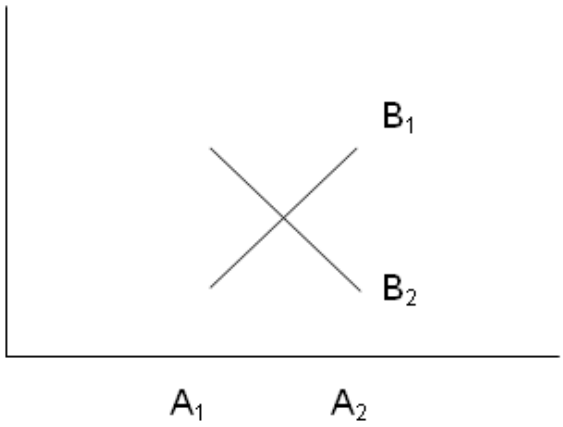
CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Arreglos ortogonales para factores con interacciones:

En los procesos de prueba de software y fundamentalmente al realizar las pruebas de caja negra o pruebas de comportamiento se producen interacciones entre los procesos o datos de entradas. Cuando el efecto de un factor depende del nivel de otro factor, se dice que existe una interacción entre los factores (Taguchi, 2009). Al planificar las pruebas se encuentran los siguientes factores “Tipo de operación” (*Factor A*) y la “Naturaleza contable de la operación” (*Factor B*), los cuales afectan la variable de respuesta (contabilización de la operación) impidiendo de esta forma realizar correctamente los comprobantes contables al termino de cada operación. Existe interacción entre los factores principales figura No. 1. Al observar la grafica analizamos cuál sería el efecto del “Tipo de operación” (Factor A) sobre la correcta contabilización de los hechos contables, entonces concluimos que depende de la naturaleza de la operación. Si el usuario selecciona “Debito” la operación disminuye el saldo a contabilizar, en caso contrario o sea, si el usuario selecciona “Crédito”, la operación incrementa el saldo a contabilizar y por lo tanto hay un aumento de la cuenta contable a la que tribute visualizando de esta forma el efecto del factor “A” sobre el “B”.



Las dos líneas son paralelas, no existe interacción entre los factores “Tipo de operación” y “Estado de la operación”.



El efecto del “Tipo de operación” depende del nivel de la “Naturaleza de la operación”.

Figura No. 1: Interacción entre los factores.

Al incluir interacciones en un arreglo ortogonal debemos tener presente el análisis realizado por Taguchi (1992).

a) Los arreglos ortogonales a utilizar para los casos con interacciones, son exactamente los mismos que se usan para el caso sin interacciones.

CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

b) Al asignar dos factores A y B por ejemplo a ciertas columnas, automáticamente la interacción de esos dos factores AXB se reflejará en otra columna del arreglo. Por lo tanto, esta tercera columna ya no podrá ser utilizada para algún otro factor o interacción a menos que se pueda suponer la interacción AXB como inexistente.
c) Una interacción significante que se desee probar, tomará una columna y en consecuencia un grado de libertad. Por lo tanto, si deseamos analizar el efecto de seis factores y cuatro de las interacciones entre ellas, requeriremos por lo menos de diez grados de libertad, esto es de diez columnas, o sea un arreglo L16 y no un arreglo L8. Que sería suficiente sin interacciones.
d) Se deberá tener cuidado especial en la manera como se asignan los factores a las columnas, para que sus interacciones no se confundan con otros factores principales u otras interacciones que también deseamos probar.

En cuanto a software se refiere planificar o probar todas las posibles variantes que se solapan traería como consecuencia una complicación adicional por la presencia de interacciones. Para lidiar con estas, los expertos en la materia hacen las observaciones siguientes (Taguchi, 1992).

- Por lo general existen pocas interacciones dentro de las múltiples posibles entre factores.
- El efecto de las interacciones sobre la variable de respuesta, es por lo general menor que el efecto de los factores individuales solos.
- Recuerde que algunos arreglos ortogonales, le permiten analizar un problema sin preocuparse por las interacciones. El L12 es un ejemplo de ellos.
- Se sugiere que, en caso de dudas sobre las interacciones, siempre sea preferible incluir más factores, en lugar de interacciones. Si estas últimas no son muy fuertes, se pueden considerar como ruido.
-

De todos los factores que afectan un proceso, se pueden extraer dos grupos: Factores de ruido. Aquellos que no podemos, querremos o deseamos controlar, y más bien deseamos que nuestros procesos y productos sean insensibles a su impacto. Factores de diseño. Aquellos que si podemos controlar en nuestro proceso de producción, y deseamos encontrar a qué nivel operarlos, a fin de optimizar el producto o proceso, esto es, que los productos sean de alta calidad y bajo costo (Taguchi, 1992).

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Durante el análisis de los casos de uso o funcionalidades de cualquier sistema para determinar los prototipos de interfaces o definirlas desde fases tempranas del desarrollo del software que se quiere construir, es importante durante la planificación de las pruebas de caja negra se conozcan a priori por parte de los ingenieros de pruebas las interacciones fundamentales que pudieran afectar el funcionamiento del software evitando los casos críticos con el propósito de incrementar la confiabilidad y disminuir los tiempos al planificar las pruebas.

Materiales y métodos

Descripción del sistema utilizado para la realización de las pruebas: El sistema de combustible SGCPro2008 (Sistema Gestor de Combustible por tarjetas magnéticas) desarrollado por Pentón (2009), utilizando Visual Basic .net como lenguaje de programación básico; la base de datos diseñada e implementada en SQL Server 2005. Software basado en arquitectura cliente – servidor. El objetivo del sistema es gestionar la contabilidad controlando al mismo tiempo las tarjetas magnéticas de combustibles utilizadas en la Empresa Comercializadora de Combustibles Matanzas.

Resultados y siscusión

El método que propone Taguchi (1999) se basa en la utilización de "matrices ortogonales". Se realizó pruebas al software SBCPro2008 para lo cual se tuvo en cuenta los casos de usos más significativos, tomando 35 para la ejecución de las mismas. Para cada uno de los casos se programó el correspondiente caso de prueba con las matrices propuestas en la tabla No. 1. Estas matrices indican qué y cuántos experimentos (pruebas al software) deben realizarse para un número de factores y de niveles determinado. Al planificar las pruebas al software para el control de tarjetas magnéticas de combustibles cuya interfaz facilita al usuario la captura de las operaciones contables figura No. 1:

CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

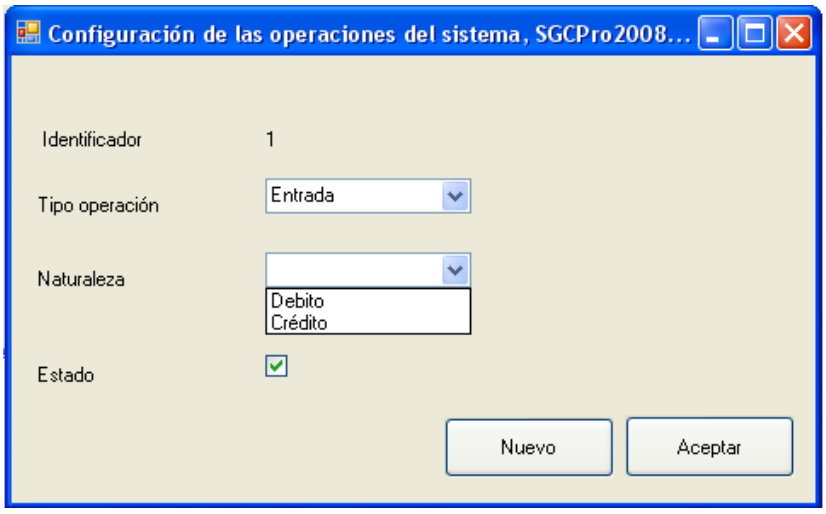


Figura No. 2: Interfaz del sistema SGCPro2008.

La GUI (Interfaz Grafica de Usuario) dispone de tres factores (el tipo de operación, la naturaleza contable de la operación y el estado de la misma) para cada uno de estos factores existen dos niveles (no se considera el identificador de la operación por ser tratado por codificación y no pueden ser cambiados por ningún usuario) (Pentón, 2009).

Factores	Nivel 1	Nivel2
Tipo de operación	Entrada	Salida
Naturaleza contable de la operación	Debito	Crédito
Estado de la operación	Verdadero	Falso

Tabla No.2: Factores y niveles correspondientes a la interfaz "Configuración de las operaciones del sistema, SGCPro2008".

Entonces el número de pruebas será ocho (efecto de elevar el número de niveles a la cantidad de factores). La mejor forma de identificar los errores es realizar las ocho pruebas (prueba exhaustiva), en las pruebas de software el número de casos de prueba que se deben planificar teniendo en cuenta la cantidad de factores y niveles de los mismos hace impracticable ejecutar todas las combinaciones (Taguchi, 2006). Este método permite racionalizar el número de pruebas sustancialmente, con solo cuatro pruebas según muestra la tabla No. 2, se puede garantizar encontrar el mayor número de errores, disminuyendo el tiempo y el esfuerzo de desarrollo del software.

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

L_4 (2^3)			
Col./No.	**1**	**2**	**3**
1	**1**	**1**	**1**
2	**1**	**2**	**2**
3	**2**	**1**	**2**
4	**2**	**2**	**1**

Tabla No. 3: Configuración del arreglo ortogonal L4.

De acuerdo con la notación empleada por Taguchi el arreglo mostrado para el caso de estudio se le llama arreglo L4 representado en la tabla No. 3, por tener cuatro renglones, los cuales son equivalentes al número de pruebas que se desarrollaran.

Preparar los casos de pruebas:

El ingeniero de pruebas durante la fase de análisis tiene una idea más solida de la importancia de las funcionalidades o casos de uso del sistema en construcción, que van refinándose posteriormente durante cada iteración hasta el diseño; en esta fase ya cuenta con elementos para analizar las propuestas de interfaz de usuarios y de esta forma preparar los casos de prueba cumpliendo el objetivo inicial de cada caso de uso. Identificar los requerimientos funcionales a probar y en qué orden según el grado de importancia para la aplicación y el nivel de acoplamiento en el proyecto (Hetzel, 2001).

Analizar el diseño de la GUI (interfaz grafica de usuario) para determinar la complejidad de cada una, especificar el número de parámetros y niveles para determinar la tabla que se ajusta según el caso y planificar las pruebas a realizar.

El caso de uso “Gestionar la configuración de las operaciones del sistema” correspondiente al modulo de interfaz objeto de estudio como variante de prueba en el trabajo de investigación (Pentón, 2009).

Al concluir las pruebas planificadas las estadísticas demostraron que de los siete errores detectados al realizar las pruebas planificadas según la tabla ortogonal, el 14.29 % fueron detectados sin interacción principales entre los factores (datos de entrada). Es importante observar que el 57.14 % de los errores se detectaron con interacciones dobles y el 28.57 % con las interacciones triples, lo que demuestra que las técnicas de Taguchi aseguran al menos el 90 % de detección de errores reduciendo considerablemente las pruebas a desarrollar. En las 35 GUI probadas del propio sistema SGCPro2008 los datos estadísticos demuestran la anterior afirmación [8]; para dichas pruebas se utilizaron fundamentalmente las tablas confeccionadas por Taguchi L4, L8, L12 y L32 para considerar otras interacciones fundamentales por las características de cada una de la interfaces. En la tabla No.

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

4 puede observarse que los parámetros de detección de errores continúan siendo mayores del 80 % para las interacciones dobles y triples (Pentón, 2009).

Tabla	Interfaces probadas	Pruebas ejecutadas	Interacciones entre los factores principales		
			Simples	Dobles	Triples
L4	15	60	2	6	4
L8	8	64	3	13	8
L12	10	120	3	17	5
L32	2	64	1	9	2
Total	**35**	**308**	**9**	**45**	**19**
Porcientos de las interacciones			**12.33 %**	**61.64 %**	**26.03 %**

Tabla No. 4: Resumen estadístico de los casos de prueba realizados.

Conclusiones.

Después de desarrollar un conjunto de pruebas utilizando las técnicas de Taguchi para la mejora continua de la calidad de los productos y procesos, en el caso específico del software de computadoras se constató que al aplicar la tabla ortogonal se reduce considerablemente el tiempo de pruebas obteniendo además resultados positivos en la calidad y confiabilidad del software. Al detectar el mayor número de errores con sólo revisar las interacciones principales se llegan a obtener aplicaciones más robustas y capaces de cumplir con los objetivos de los requisitos funcionales pactados con el cliente en la fase inicial. La propuesta de Taguchi demuestra que se puede detectar el mayor número de errores en el software con las interacciones dobles y triples entre los factores principales, pudiendo liberar el producto al usuario final con el mínimo de errores.

Referencias bibliográficas

Hetzel, W. (2001). The Complete Guide to Software Testing. QED Information Sciencie, Inc; Wellesley, Ma.

Jacobson I. Booch y G. Rumbaugh J. (2003). El proceso unificado de Desarrollo de Software. The Unified Software Development Process. ISBN 0 – 201 – 57169 – 2.

Kaner, C.; J. Falk; H. Q. Nguyen. (2007). Testing Computer Software, Van Nostrand Reinhold.

Pentón Saucedo, A. E. (2009). Tesis. Sistema Gestor de Combustible por Tarjetas

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Magnéticas SGCPro2008. Matanzas, Cuba.
Phadke, M. S. (2005). “Planning Efficient Software Test”, Cross Talk, vol. 10. USA.
Pressman, Roger S. (2009). Ingeniería de Software. Un enfoque práctico. 5th, Ed. Madrid, España.
Pressman, Roger S. (2010). Ingeniería de Software. Un Enfoque Práctico. Madrid, España.
Http://www.scribd.com/…/Ingenieria-de-Software-Un-Enfoque-Practico-Pressman-5th-Ed.
Taguchi, G. (1999). Arreglos Ortogonales y Graficas Lineales. Ed. ASI PRESS. Madrid, España.
Taguchi, G. (2005). El diseño experimental y los métodos de Taguchi. Consultado 12 de Diciembre, 2009, disponible
Http://www.ideas.repec.org/p/cem/doctra/258.html.
Taguchi, G. (2006). Diseño de Cuadrados Latinos. Consultado 4 de Marzo, 2010, disponible
Http://www.buenastareas.com/ensayos/Cuadrados-Latinos/2057921.html.
Taguchi, G. (2007). El diseño experimental y los métodos de Taguchi. Consultado 20 de Octubre, 2009, disponible
http://www.ucema.edu.ar/publicaciones/documentos/258.pdf.
Taguchi, G. (2009). Filosofías de la aplicación de las Técnicas de Calidad. Consultado 5 de Septiembre, 2009, disponible
Http://www.mitecnologico.com/Main/FilosofiaDeGenichiTaguchi.
Taguchi, G. (1992). Introduction to orthogonal arrays. In Techniques for Quality engineering.

Fecha de recepción: 28/12/2011 Fecha de aprobación: 6/06/2012