

Modelo para identificar defectos en Control de Calidad, basado en Oráculos de pruebas y Sistemas Expertos

Trinidad Choque Tucupa
 Postgrado en Informática
 Universidad Mayor de San Andrés
 La Paz – Bolivia
 trynydad7@gmail.com

Resumen—El objetivo principal del proceso de control de calidad es la identificación de defectos; para mejorar y perfeccionar las tareas del proceso se plantea la aplicación de oráculos de pruebas basados en sistemas expertos; donde el sistema experto opera sobre la lógica de negocios del programa bajo prueba; pero ¿Cómo relacionar el concepto y estructura de un caso de prueba con un sistema experto?; para tal objetivo se plantea el diseño de un modelo de variables; variables que forman la conexión entre ambos conceptos con el fin de obtener casos de pruebas evaluados sin la participación de un operador externo.

Palabras clave—Buchanan, Ciclo de vida detallado, Control de calidad, Oráculo de prueba, Sistema experto.

I. INTRODUCCIÓN

Cuando el software es un producto que se encuentra continuamente en procesos de mantenimiento, se puede identificar que dentro del proceso de control de calidad existen pruebas que deben ser ejecutadas en diferentes líneas de tiempo [1] con el objetivo de encontrar defectos críticos que pueden afectar al inicio de pruebas; también el personal que forma parte del equipo debe tener conocimiento sobre la funcionalidad actual del sistema y los nuevos cambios aplicados. Ante estas características se identifica que la aplicación de pruebas en la etapa inicial demora en su ejecución debido a que las pruebas son ejecutadas manualmente y ante un defecto encontrado se debe identificar el motivo o causa del mismo, aspectos que generan una retroalimentación tardía de los defectos hallados; entonces ¿Cómo se podría minimizar el trabajo manual de testeo sin tener que usar personal para validar las salidas de las pruebas?, en este sentido se identificó los siguientes trabajos relacionados con esta problemática.

Oráculos de prueba con un planteamiento heurístico de apoyo a la decisión, en este trabajo se plantea el estudio de modelos heurísticos para el diseño de oráculos que apoyan a la decisión de la aceptabilidad de una prueba en el ámbito de las aplicaciones Web, con la implementación de modelos heurísticos que validaran los diferentes caminos de aceptación del código con relación a las pruebas del software; se pretende validar si existen cambios en la funcionalidad del producto [2].

También se tiene el trabajo: Aplicación de un oráculo de prueba automatizado a la evaluación de salidas de programas

basados en XML. Este trabajo propone un oráculo de prueba automatizado dirigido a la prueba de programas de procesamiento de XML, el oráculo propuesto en este trabajo opera con una especificación del programa bajo prueba combinando dos niveles de especificación: uno de los requisitos de comportamiento particulares del programa bajo prueba, proporcionada por el ingeniero de pruebas, y una especificación invariante del mecanismo de evaluación del oráculo programada en jquery, el cual determina si el programa cumple los requisitos de comportamiento suministrados, la automatización de este oráculo está determinado por el uso de un lenguaje de especificación [3].

Ante estas consideraciones, el presente trabajo está enfocado en la definición, obtención y evaluación de un oráculo de prueba destinado a apoyar el desarrollo de las pruebas de software. Así también el oráculo de prueba basara su validación en el conocimiento almacenado en un sistema experto, con el objetivo de evaluar las salidas de las pruebas sin la intervención de un operador externo.

En base a lo descrito anteriormente se toma sistemas expertos basados en reglas [4] para aplicar el modelo y el concepto de oráculos de prueba. Se selecciona sistemas expertos por contar con los subsistemas de: control de coherencia y de explicación, con el primer subsistema se pretende validar la relación coherente de las reglas de negocio al momento de ingresar nuevo conocimiento al sistema; y el segundo subsistema nos permitirá proporcionar una explicación clara del resultado emitido luego de validar la salida de la prueba inicial[5], este resultado será base para la toma de decisiones en el inicio o no de pruebas de control de calidad.

II. MÉTODOS

La Figura 1 muestra los procesos llevados a cabo para el desarrollo del tema de investigación, y luego, se detallan las tareas realizadas en cada proceso.



Fig. 1. Procesos realizados. Fuente: Elaboración propia.

Para referenciar este artículo (IEEE):

[N] T. Choque, «Modelo para identificar defectos en Control de Calidad, basado en Oráculos de pruebas y Sistemas Expertos», *Revista PGI. Investigación, Ciencia y Tecnología en Informática*, n° 8, pp. 193-195, 2020.



A. Sistema Experto

Para iniciar con el desarrollo se llevó a cabo la obtención y consolidación del conocimiento del software a probar, para esto se hizo uso de la metodología Buchanan [6] y Modelo de ciclo de vida detallado [7]. Se aplicó ambas metodologías con el objetivo de estructurar las tareas que se ejecutarían, se desarrollaron las tres primeras fases de la metodología Buchanan.

Identificación: Se identifica el alcance y los actores con los cuales se trabajarán para la obtención de conocimiento. Se identifican las fuentes y repositorios de conocimiento, así como las técnicas que serán aplicadas para dicha tarea.

Conceptualización: Se hace uso de las tareas descritas en el modelo de ciclo de vida detallado. Se aplican plantillas en las cuales se documentan las tareas y objetivos alcanzados en las diferentes iteraciones realizadas. Se hace uso de entrevistas y cuestionarios para validar el conocimiento adquirido.

Formalización: Se diseñan arboles de decisión [8]; con el objetivo de comprender y plasmar de mejor manera la funcionalidad del software a probar. Con este diseño se valida los cambios que sufren los objetos al ser procesados por el software. Se identifican estados iniciales y finales.

B. Casos de Prueba

En base al conocimiento obtenido se diseñan los casos de prueba con el objetivo de validar el razonamiento modelado en el sistema experto. Los casos de prueba tienen la estructura de entradas, acciones y salidas esperadas.

C. Modelo

Como relacionar la estructura del caso de prueba con el sistema experto; para que dicho sistema experto actúe bajo las características de un oráculo de prueba (*Generador*. - Proporciona los resultados esperados para cada prueba; *Comparador*. -Compara los resultados obtenidos con los resultados esperados y *Evaluador*. -Determina si la comparación de los resultados es suficientemente cercana para considerar pasada la prueba [2]). Según lo comprendido hasta este punto se plantea el siguiente proceso para identificar las variables que nos permitirán realizar el nexo como se muestra en la figura 2.



Fig. 2. Modelo para la obtención de variables. Fuente Elaboración Propia.

1. Identificar las variables de entrada que se desea combinar para las pruebas, cada variable tiene una cantidad determinada de valores.
2. Determinar los valores de cada variable
3. Depurar las variables considerando:
 - a) No repetir variables
 - b) Las variables en sus diferentes valores deben afectar en el proceso de lógica de negocio

4. Considerar como serán obtenidos los valores de cada variable a través del software probado.
5. Agrupar las variables de modo que al ser relacionadas entre sí, puedan emitir un resultado en base a la lógica de negocios.

Con la aplicación de este proceso se identificaron las variables que formaran parte de los hechos del sistema experto.

D. Reglas Lógicas

Al tener identificadas las variables y los estados que estos pueden tener al ser procesados por el software. Se hace uso del método combinatorio para establecer el número de combinación que se realizara y ante cada combinación se establece la salida evaluada en base al conocimiento adquirido; esto significa que cada salida representa una verdad o un defecto del caso de prueba; así el sistema experto actúa como un generador, comparador y evaluador de casos de prueba.

E. Pruebas

A la conclusión del diseño de las reglas del sistema experto se realiza el análisis y codificación de pruebas automatizadas en el software a probar. Los casos de prueba que fueron diseñados en el caso B, son los casos que se encuentran automatizados.

Se establece la relación entre las salidas emitidas por el caso de prueba automatizado como entradas al sistema experto. El diseño de las salidas que emite la prueba automatizada hace uso del modelo propuesto. De este modo el sistema experto es retroalimentado con entradas, acciones y salidas, las cuales evaluará como oráculo de prueba. Proporcionando un resultado de defecto o validez para el caso de prueba ejecutado. El caso de prueba presenta defecto cuando la salida no es la misma que establece la lógica del sistema experto. El sistema experto proporciona información del motivo por el cual la salida es defectuosa.

Para las pruebas realizadas se generaron aleatoriamente objetos de prueba los cuales simulan todos los estados validos e inválidos que son procesados por el software. Para realizar las pruebas se hizo uso de pruebas de parwise [9] y tabla de decisiones; con el objetivo de identificar algún error o la falta de respuesta ante cierta combinación de entradas.

III. RESULTADOS

A continuación, se define el tamaño de la muestra haciendo uso de la ecuación estadística [10]:

$$n = \frac{k^2 * p * q * N}{(e^2 * (N - 1)) + k^2 * p * q} \tag{1}$$

Donde: N=314984 k=1.65
 e=5% p=0.5
 q=0.5

Por tanto, se tiene que: n = 272 es el tamaño de la muestra

Se ejecutó la prueba automatizada en los escenarios descritos en la tabla 1 obteniendo sus correspondientes resultados.

TABLA I. RESULTADOS DE PRUEBAS

Base de conocimiento	Exitosos	Defecto	Sin resultado
Sin modelo	50	0	222
Con modelo	149	123	0

IV. DISCUSIÓN

En base a los resultados obtenidos se puede apreciar que todo el conjunto de casos de prueba fue evaluado, proporcionando información del resultado para cada caso de prueba; al evaluar los resultados obtenidos se verifico que los datos son correctos. El uso de un sistema experto basado en reglas es útil cuando el software a probar se basa en reglas de negocio fija. Falta determinar si el enfoque de construcción del modelo es válido para un oráculo de prueba basado en otra área de la inteligencia artificial. Donde las reglas de negocio serán variables y el conocimiento modelado pueda actuar ante incertidumbres.

V. CONCLUSIONES

La aplicación de un modelo de variables para el diseño de oráculos de prueba apoyo el proceso de elaboración de reglas de lógica del sistema experto, debido a que se enfocó en la relación de las reglas con los procesos que realiza el software a probar y así relacionarlos con casos de prueba. La implementación de un sistema experto como un validador de casos de prueba es eficiente cuando los cambios del software son considerables en su lógica.

El sistema experto basado en reglas es una base adecuada para la construcción de reglas que tienen características relacionadas con las entradas de los casos de prueba

Se obtiene una respuesta validada sobre los defectos encontrados en un tiempo mínimo. Dichas respuestas apoyan a la toma de decisiones en cuanto a que si los cambios presentan defectos que pueden impedir el proceso de inicio de control de calidad o si los cambios realizados pueden ser enviados a producción para su uso por parte del cliente.

Ante la existencia de herramientas que realizan el proceso de pruebas automatizadas y validaciones, se observa que la diferencia radica en que un sistema experto evalúa todas las reglas del comportamiento del software en base al conocimiento adquirido; una herramienta automatizada programada con su

salida valida, requiere que para cada caso de prueba automatizado se describa la salida esperada y si la salida es incorrecta esta debe ser evaluada por una persona externa. El sistema experto hace la diferencia debido a que no requiere de un operador externo para validar la salida del caso de prueba, el sistema experto valida el caso en función del conocimiento adquirió y emite un resultado para la toma de decisiones en cuanto a la aplicación de los cambios. Así también explica donde se encuentra el defecto hallado.

REFERENCIAS

- [1] M. J. E. M. M. J. J. G. Arturo H. Torres, «Oráculos de prueba: Un planteamiento heurístico de apoyo a decisión.» de Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 3, No. 1, Sevilla, España, 2009.
- [2] C. d. I. R. y. J. T. Dae S. Kim-Park, «Aplicación de un oráculo de prueba automatizado a la evaluación de salidas de programas basados en XML.» Revista Española de Innovación, Calidad e Ingeniería del Software, p. 22, 2011.
- [3] G. Sandoval, Analista de Pruebas ISTQB Nivel Avanzado, Bolivia: Businnes Innovations, 2012.
- [4] G. o. M. A. S. H. Castillo Enrique, Sistemas Expertos y Modelos de Redes Probabilísticas, España: Universidad de Cantabria, 2011.
- [5] S. V. Soto, Inteligencia Artificial, Córdoba: Universidad de Córdoba.
- [6] L. Delgado, A. Cortez y E. Ibáñez, Aplicación de metodología Buchanan para la construcción de un sistema experto con redes bayesianas para apoyo al diagnóstico de la Tetralogía de Fallot en el Perú, Lima, Perú: Industrial Data, Universidad Nacional Mayor de San Marcos, 2015.
- [7] J. Giarratano y G. Riley, Sistemas Expertos Principios y Programación, Universidad de Houston-Clear LAke, 2002.
- [8] S. Norvig y J. Russell, Inteligencia Artificial un Enfoque Moderno, Madrid: Pearson Educación, S.A., 2006.
- [9] M. Polo Usaola, B. Pérez Lamancha y P. Reales Mateo, Técnicas combinatorias y de mutación para testing de sistemas de software, Madrid: RA-MA, 2012.
- [10] R. Moya y G. Saravia, Probabilidad e Inferencia Estadística., Perú: San Marcos.

Breve CV de la autora

Trinidad Choque Tucupa es Licenciada en Informática por la Universidad Mayor de San Andes 2007.

Ejerce profesionalmente como desarrolladora, QA y en docencia universitaria. Obtuvo reconocimiento como docente en la Universidad Privada Franz Tamayo 2011.

Autora de manuales de apoyo: programación básica y en análisis y diseño de software. Sus intereses investigativos son la inteligencia artificial, patrones de control de calidad y arquitecturas de software.

Email: trynydad7@gmail.com.