

## **EJECUCION SIMBOLICA. UNA FORMA DE ASEGURAR LA CREACION DE CODIGO DESDE EL INICIO EN LA ETAPA DE DESARROLLO.**

Investigadores USAL:

Directora Daldín Susana (susana.daldin@usal.edu.ar); López Pasaron, Christian

Alumnos Practicantes USAL:

Porta, Estefania; Monia, Sebastian; Vázquez, Víctor

### **Resumen**

Desde hace algunos años la sociedad se ha visto rodeada de cada vez más información.

Internet, como medio de transmisión, ha supuesto una gran evolución en materia de tiempos de difusión y realimentación entre editores (quienes generan la información) y lectores (quienes la consumen).

A su vez el desarrollo de lenguajes de programación multiplataforma como JAVA, unido a la fabricación de dispositivos que permiten su ejecución (desde tablets y smartphones hasta automóviles y electrodomésticos), ha rodeado al ser humano de innumerables dispositivos informáticos, dispositivos que para manejar información poseen programas de software. En este marco, el desarrollo de software se ha convertido en una pieza fundamental.

El Ciclo de Vida de un Proyecto (CVP) divide a un desarrollo de software en diversas etapas, dentro de la cuales se encuentra “testing”. Esta etapa insume un alto porcentaje de los recursos del proyecto. La técnica de ejecución simbólica simula la ejecución del código transformando variables de tipo en símbolos, permitiendo identificar qué caminos de ejecución se toman de acuerdo a los valores de entrada, pudiendo identificar claramente qué valores ha tomado la variable en cuestión y si lo hace de manera esperada o no.

Se están generando entrevistas semiabiertas con los departamentos de Aseguramiento de Calidad (AC) de diversas empresas de desarrollo, para cualificar y cuantificar los tipos de test que se realizan a lo largo de los proyectos y de esta manera poder determinar el estado actual del arte.

Dentro de esta subetapa se detectó que las empresas casi no utilizan la ejecución simbólica como método de detección de errores dentro de la etapa de testing. Las causas son diversas pero tienden a asociarse a un elevado costo computacional, lo que genera un consumo de recursos económicos mayores al de las demás técnicas usadas en la actualidad.

Los objetivos generales de la investigación son el análisis de la ejecución simbólica como metodología de aseguramiento de la calidad, escalabilidad, estabilidad y seguridad del desarrollo del software mediante la asistencia en la generación de test y, a partir de los resultados, el desarrollo de una metodología para analizar código fuente y/o compilado para agilizar la creación de test de prueba unitarios.

**Palabras clave:** ejecución simbólica; aseguramiento de la calidad; desarrollo seguro; caminos de ejecución; análisis de código.

## **Abstract**

For several year society has been surrounded by more and more information. Internet as communication tool has developed a great evolution in timing and feedback between publishers (who generate information) and readers (who consume it).

Furthermore the development of several multiplatform language (such as JAVA) in addition to several devices that helps it execution (such as tablets and smartphones up to cars and appliances) have surrounded the human being of several computer devices, which hold internal information to manage programs software. As a result of this it is pertinent to say that software development has become a key player.

The lifecycle of a project divides a software development into several stages in which we can find testing. At this moment the resources of the project are highly consumed. The symbolic execution technique simulates the execution of the code transforming type variables in symbolic variables letting us identify which values were taken by the variable and if these were expected or not.

Semi-open interviews with various departments of Quality Assurance (QA) development companies are been generated to qualify and quantify the types of tests that are performed along the project and be able to determine the current state of the art.

In this sub-step we detect that the companies hardly use the symbolic execution

as a method of detecting errors in the testing phase. The causes are varied but are usually associated with a high computational cost generating higher consumption of economic resources to the other techniques used today.

The general objectives of the research are the analysis of the symbolic execution as a quality Assurance, scalability, stability and security in the software development through assisting the creation of test and from the answers obtained the development of a methodology to analyze the source code and compilation to speed up the unit test.

**Keywords:** symbolic execution; quality assurance; safe development; execution paths, code analysis.