**Desarrollo idea tema de tesis**

**“Idea inicial test de software producido en el trabajo”**

Hace varios años que me dedico a la mantención y extensión de un ERP llamado GP. Básicamente a este software se le fue agregando funcionalidades que no estaban disponible en la región de latino américa. Al principio las extensiones o modificaciones eran pequeñas y simples como un reporte o agregar un par de campos a un formulario. Con el pasar del los años los módulos se fueron haciendo cada vez más grandes y en algunos de los casos lo que había iniciado como productos o módulos aislados y pequeños comenzaron a crecer en complejidad e interactuar entre ellos. Dado este nuevo escenario las extensiones y mantenimiento de los módulos comenzaban a hacerse cada vez más difíciles y comenzaron a surgir algunos problemas tales como la reutilización de funciones o formulario, que al no cumplir completamente con lo requerido se modificaban. Esto en algunos casos afectaba funcionalidades que ya habían sido testadas.

Por otra parte GP no tiene un marco de trabajo para poder realizar el testing como proveen otros lenguajes de programación como java o python. El testing es realizado por el mismo desarrollador o el área funcional. Al pasar el tiempo el testing se refinó haciendo documentos con capturas que muestran el paso a paso para poder reproducir el caso testeado pero nunca pasó al siguiente escalón en mejoras, por ejemplo una automatización.

Una situación común que encuentro es la siguiente. Al iniciar un desarrollo se prueba por ejemplo que el sistema guarde info nueva (alta) en la base de datos. Este testing se hace de forma manual. Luego se pasa a la modificación y una vez terminada la codificación se testea esta nueva funcionalidad. En este ejemplo se podría probar el alta y la modificación de forma conjunta y ver si el módulo está funcionando de forma correcta. Pero este ejemplo muestra que el testing es acumulativo. Es decir que con cada nueva funcionalidad necesito garantizar el funcionamiento de lo que estaba funcionando anteriormente. Esta situación aplica siempre, es decir, para desarrollos nuevos, extensiones o modificaciones a módulos existentes. El costo que tiene hacer pruebas para garantizar la correcta funcionalidad es muy alto en tiempo y al no tener una automatización de esto, queda en responsabilidad del desarrollador y el funcional realizar el testing completo.

Está situación me me lleva a pensar cómo desarrollar una herramienta que pueda ayudarme en este sentido. El tema del testing hecho por el desarrollador es que si es manual y lo hace el mismo, muchas veces se prueban cosas básicas pero a medida que va incrementando la complejidad deja pasar las pruebas básicas y hace pruebas más complejas. Lo cual es bueno, pero a veces la implementación de funcionalidades nuevas puede romper comportamientos correctos que el software tenía antes de la modificación. Lo ideal sería ser poder tener un conjunto de pruebas programadas las cuales puedan ir ejecutando cada vez que se hace una modificación o se sube una corrección o una nueva funcionalidad. Poder tener una herramienta que pueda garantizar una correctitud de un conjunto mínimo de funcionalidades y funcionalidades anteriores.

Por otra parte frente a la ausencia de un marco de trabajo para el testing veo que también sería bueno determinar qué cosas o características debería tener una herramienta de testing o el marco para desarrollar una. Es decir el lineamiento general que requiere esto y que esta “Idea” pueda ser aplicada no solo a GP sino a cualquier software cuyo entorno de desarrollo o lenguaje carezca de este tipo de herramientas.

De lo que he leído entiendo que hay 3 grandes grupos o categorías dentro del testing de software. El testing funcional, el no funcional y el de regresión.

El testing funcional es uno de los puntos que sería interesante ver. Entiendo que el testing funcional es la base para el testing de regresión ya que que el testing de regresión es la ejecución parcial o total de los testing funcionales.

Ahora dentro del testing funcional está el testing unitario, este tipo de testing toma segmentos o partes del producto (unidades) como funciones procedimientos. El propósito de este testing es validar que cada unidad de software se comporte como se espera. Una vez que se tenga un poco sobre este testing, el test de regresión solo seria poder administrar y ejecutar estos test unitarios en forma conjunta y poder analizar los resultados.

Esto es una idea un poco más desarrollada de lo que veo que esta pasando en mi trabajo.

Mi dudas serían:

* ¿Es viable para tema de tesis?
* ¿Si es viable, por donde sigo investigando o que investigo?
* Estoy en el plan fines y me estan quedando 6 meses para presentar el trabajo. La idea es hacer sencillo que me permita terminar.
* Que disponibilidad tiene para poder hacer una reunión. Así lo vemos en más detalle y mas formal.