

El Proceso de la Prueba de Software

LENGUAJES DE DEFINICIÓN DE PROCESOS



Luis Vinicio León Carrillo es profesor-investigador del Departamento de Electrónica, Sistemas e Informática del ITE-SO, y director general de e-Quality S.A. de C.V., empresa especializada en prueba de software. Luis Vinicio es doctorando por la Universidad Técnica de Clausthal, Alemania. Luis Vinicio es coautor de un marco tecnológico que hoy permite a e-Quality desarrollar empresas de prueba de software. En el contexto de ese marco, su tesis doctoral tiene que ver con aplicaciones de métodos y lenguajes formales para hacer más eficiente y efectiva la prueba de software. Es co-fundador del Capítulo Guadalajara de la AMCIS y su Secretario actual.

En el número anterior definimos formalmente (utilizando BNF) lo que en la literatura especializada se conoce como un Process Definition Language (PDL). En este número utilizaremos ese lenguaje para definir de manera formal los fragmentos de un proceso de prueba de software.

Antes de hacerlo, quisiera que recordemos que, aunque el lenguaje presenta aspectos importantes de un PDL, se trata de uno muy pequeño y simple, en el que dejamos fuera componentes importantes por cuestiones de espacio, en particular lo referente al sistema de tipos y al paso de parámetros.

En su columna del número anterior, la Dra. Hanna Oktaba hablaba de dos escuelas con versiones antagónicas de lo que es un proceso. En buena medida porque nuestro pequeño PDL es procedural, podemos decir que se adhiere a lo que ella llamaba “la escuela del SEI”. Documentar un proceso con un PDL definido formalmente facilita también el intercambio de ideas y de experiencias relacionadas con procesos de software sobre bases muy precisas.

Esto puede ayudar a reducir discusiones sin fundamento, o a descartar argumentos de empresas que se proclaman a sí mismas líderes en su área porque dicen contar con procesos “novedosos” que ni siquiera han documentado con el rigor metodológico suficiente.

Un Subproceso de Prueba

A continuación presentamos pequeños fragmentos de un subproceso de prueba de software que podrían corresponder a la fase de las pruebas alpha en el “Modelo-V” que se mencionó en nuestra columna del número anterior. Cabe mencionar que en esta definición, el énfasis está, por un lado, en mostrar el uso de las construcciones del PDL, y por otro, en refinar un poco más el proceso de prueba de software aunque sin pretender ser exhaustivo (nos concentramos en lo que creímos que podría resultar más interesante); algunas actividades podrían documentarse de otra manera.

Hemos estructurado los fragmentos de manera que sea explícita la consideración de las áreas de la administración de proyectos propuesta por el Project Management Institute, parte de ello se encuentra listado en el subproceso planeación().

```

proceso pruebas_alpha( );
<Historia_de_Cambios>;
<Constantes_Tipos_Variables>

proceso gestación( );
<Constantes_Tipos_Variables>
  proceso establecer_alcances( ); ...
  proceso definir_entregables( ); ...
  proceso establecer_criterios_de_éxito( ); ...

  proceso establecer_criterios_para_regresiones( );
...
  inicio
    si (el SUT es un sistema crítico)
      criterio_para_regresiones = volver a aplicar
                                todos los casos de prueba

    si nó
      criterio_para_regresiones = volver a aplicar
                                sólo el 75% de los casos de prueba
  fin;

  proceso estimar_esfuerzo( ); ...
...
inicio
  paralelizar
    inicio
      establecer_alcances();
      definir_entregables();
    fin
  inicio
    establecer_criterios_de_éxito();
    establecer_criterios_para_regresiones();
  fin
  inicio
    estimar_esfuerzo();
  fin
  fin_p
fin;

proceso planeación( );
<Historia_de_Cambios>;
<Constantes_Tipos_Variables>
  proceso docum_admón_de_riesgos( ); ...
  proceso docum_admón_del_tiempo( );...
  proceso docum_admón_del_personal();...
  proceso docum_admón_de_la_calidad( );...
  proceso docum_admón_de_la_comunicación();...
  proceso docum_admón_de_proveedores( );...
  proceso docum_admón_de_costos( ); ...

```



e-Quantity^{Corp.}[®]

Nosotros
probamos
software...
¡Pónganos a prueba!

- Entrenamiento en Prueba
- Evaluación y mejora de Procesos de Prueba
- Puesta en marcha de Organizaciones de Prueba
- Buró especializado de Testers

Iniciamos
Diplomado
Básico en Prueba
Ciudad de México
Febrero 2007

incluir la generación automática de documentos en algún subconjunto del Español (un así llamado *restricted language*) que respete formatos establecidos. Entre otras ventajas que esto tiene, es poder agilizar (y hacer más sistemática) la actividad de la documentación de procesos en un proyecto de mejora bajo algún modelo de procesos como CMMI o MoProSoft. Así se contribuye a mejorar la relación costo-beneficio de dicho esfuerzo, que como Hanna nos comenta, es una de las principales preocupaciones de las organizaciones que consideran este tipo de iniciativas.

Por supuesto, esto aplica también si se trabajara en el marco de un modelo de calidad especializado en prueba de software, tal como TPI (*Test Process Improvement*) o TMM (*Test Maturity Model*). ©

—Luis Vinicio León

- Evaluación de Productos de Software



e-Quantity
Approved

+ Calidad
+ Utilidades
+ Tranquilidad

— Riesgos
— Esfuerzo
— Costos

www.e-quality.net

contact@e-quality.net

+52 (33) 3030 6909