Proceso de Construcción de Aplicaciones de Software Libre

Fase de Construcción

En esta fase se codifican las funcionalidades de la aplicación de software correspondientes a la iteración actual, se construye la interfaz de usuario y la base de datos. De esta manera, en esta fase, con cada iteración de desarrollo, se obtiene una nueva versión de la aplicación, clasificada como versión beta, es decir, una versión sobre la cual se deben realizar un conjunto de pruebas como pruebas funcionales y no funcionales.

A continuación se indican las actividades y tareas que componen esta fase.

Actividad: Construcción de la base de datos

Tarea: Construir la base de datos conforme al modelo de datos persistentes.

Herramientas:

- Algunas herramientas que permiten generar los scripts para base de datos conforme al modelo de datos: ER Visual, BD Designer Fork (principalmente para trabajar con base de datos MySQUL).

Productos:

- Base de datos.

Responsables:

- Programadores.

Colaboradores:

- Equipo de Desarrollo.

Actividad: Adaptación y/o codificación de los componentes requeridos para construir las funcionalidades del software

Tarea: Codificar los componentes requeridos para construir las funcionalidades asociadas a la iteración actual.

Recomendaciones:

- Para agilizar la fase de codificación y mejorar la calidad del software se recomienda el uso de framework de desarrollo, así como de patrones de programación y componentes reutilizables que puedan ser adaptados según se requiera.

Entre los patrones de programación que se pueden utilizar se encuentran: patrón de acumulación, patrón lectura de datos, patrón de conteo (Mateu, s.f.); patrón Super Loop, patrón Background / Foreground (Alvarez, s.f.); entre otros.

Productos:

- Componentes implementados, componentes adaptados.

Responsables:

- Programadores.

Tarea: Realizar las pruebas unitarias a los componentes adaptados y/o codificados y corregir los errores encontrados.

Recomendaciones:

- A fin de evitar errores al momento de integrar los componentes del software se recomienda elaborar pruebas unitarias para verificar el funcionamiento, por separado, de cada uno de estos componentes.
- Para agilizar las actividades asociadas a la aplicación de pruebas unitarias se recomienda la automatización de éstas.

Herramientas:

- Para aplicar pruebas unitarias se pueden utilizar herramientas como: PhpUnit? (para lenguaje PHP), Junit (para lenguaje Java) , xUnit (para distintos lenguajes de programación), entre otras.

Productos:

- Errores corregidos.

Responsables:

- Programadores.

Actividad: Construcción de la interfaz de usuario

Tarea: Construir la interfaz de usuario correspondiente a las funcionalidades asociadas a la iteración actual.

Herramientas:

- Para crear interfaces de usuario se pueden utilizar herramientas como: TkInter?, wxPython, PyGTK, PyQt?, estas herramientas permiten crear interfaces gráficas en Python. También se pueden utilizar: Codeblocks, CodeLite?, Gtkmm, entre otras.

Productos:

- Interfaz gráfica del software.

Responsables:

- Diseñador Gráfico, Programadores.

Actividad: Documentación del código fuente

Tarea: Documentar el código fuente.

Recomendaciones:

- La documentación del código fuente representa una actividad fundamental para facilitar los procesos de apropiación del software (mantenibilidad), por lo cual se recomienda realizar la documentación de todos los componentes de éste, tanto los codificados como los reutilizados.
- Se sugiere que la documentación de cada componente (función, método, clase), como mínimo, haga referencia a qué hace el componente, qué parámetros hay que pasarle y qué devuelve (Fernández, 2008).

Herramientas:

- Para generar la documentación del código fuente se pueden utilizar herramientas como: Doxygen, phpDocumentor, entre otras.

Productos:

- Código documentado.

Responsables:

- Programadores.

Actividad: Gestión de las versiones del software a través de un sistema de control de versiones

Tarea: Colocar el código fuente desarrollado en cada iteración en el sistema de control de versiones que se utilice en el proyecto.

Recomendaciones:

- El uso de sistemas de control de versiones es muy importante para el manejo organizado de las distintas versiones que se desarrollan para un software, pues éstos permiten gestionar los diversos cambios que se realizan sobre estas versiones, facilitando así el trabajo colaborativo y cooperativo en torno al desarrollo de software.

Herramientas:

- Existen muchas herramientas para el control de versiones de software, entre ellas tenemos: CVS, SVK, Subversion, SourceSafe?, Mercurial, Bazaar, GIT, Darcs.
- La mayoría de las plataformas para gestión de proyectos de software contienen sistemas para el control de versiones, tal es el caso de la plataforma Trac que se utiliza en la Fundación Cenditel.

Productos:

- Código fuente del software contenido en el sistema de control de versiones.

Responsables:

- Programadores.

Actividad: Automatización del proceso de instalación/desinstalación del software

Tarea: Codificar los pasos requeridos para instalar y desinstalar el software desarrollado.

Recomendaciones:

- Es importante que durante el proceso de instalación se permita al usuario suspender dicho proceso, y que se muestren a éste mensajes de información sobre el éxito o no de la instalación. De igual manera, para facilitar el uso del software se recomienda que durante el proceso de instalación se creen los iconos para el acceso directo a éste.

Productos:

- Proceso de instalación/desinstalación automatizado.

Responsables:

- Programadores.

Actividad: Corrección de errores

Tarea: Corregir los errores reportados.

Recomendaciones:

- Al utilizar un sistema de control de errores cada Programador puede tener acceso a los errores que se le asignen para su respectiva corrección, así como también podrán acceder a la información donde se describe el error respectivo (reporte).
- Una vez corregido el error se recomienda al Programador registrar en el sistema de control de errores la causa del error y la corrección realizada. Esto permite llevar un historial sobre formas de resolver errores, el cual puede ser utilidad para solventar en el futuro errores iguales o similares que se puedan presentar.

Productos:

- Código corregido.
- Registro de correcciones en el sistema de control de errores.

Responsables:

- Programadores.

volver a metodología