

MODELO DE DESARROLLO DE BASE DE DATOS v2.0

Alain Osorio Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba, aorodriguez@uci.cu

The methodologies and software development processes do not structure the development of databases. Rational Unified Process (Rational Software Corporation) addresses the development of database as a design task, lacking the necessary specifications in all the effort to develop and maintain a database. Agile methodologies do not distinguish between application development and database development within the application development, lacking, as RUP, general details that are always applicable (Hassan, et al.). Agile Data Method is a good approach because it defines a set of best practices from the agile perspective, but does not define continuity throughout the development cycle (Ambler, 2003). Oracle Unified Process is a methodology developed by Oracle, based on RUP, to structure the development of applications developed only by the company, including databases (Oracle® Corporation, 2012). The aim of this paper is to present a database development framework, which enables the development team to establish a baseline in terms of database development. By applying this framework, it is possible a justified and purposeful coordination of the main elements of database software in the development of a generic way.

INTRODUCCIÓN

El presente trabajo pretende generalizar una práctica que cada empresa, equipo y persona desarrolla de forma particular según su experiencia y resultado: desarrollo de bases de datos desde el punto de vista de la Ingeniería de Software. Se pueden obtener resultados positivos desarrollando base de datos para sistemas de software organizada o desorganizadamente. La diferencia fundamental se evidencia en el esfuerzo de desarrollo, la calidad de la arquitectura de datos medida por el balance entre sus atributos de calidad y las facilidades de soporte y administración una vez que todo el sistema esté en funcionamiento en su entorno real de ejecución. No existe una metodología de desarrollo de software universal que resuelva todos los problemas, ya que cada desarrollo de software tiene sus propios problemas. De la misma forma no existe un consenso en cuanto a definiciones genéricas de base de datos dentro de la Ingeniería de Software dentro de proyectos de desarrollo de software.

DESARROLLO

El framework transita por tres fases o etapas fundamentales:

1. Inicio o Definición: se definen los principales requerimientos y la arquitectura de datos.
2. Desarrollo o Construcción: desarrollo de la solución de software.
3. Despliegue o Transición: puesta en marcha y soporte del sistema. Incluye la etapa de pruebas de liberación.

El framework está dividido en cuatro actividades las cuales incluyen tareas generales:

1. Modelado de Datos: responde a la estructuración del modelo de datos.
2. Configuraciones: responde a las configuraciones de la Arquitectura de Datos.
3. Implementación: responde a la estructuración de todas las funcionalidades de base de datos.
4. ADTP: por las siglas de Acceso a Datos, Tuning (optimización por su traducción del inglés) y Prueba. Responde a la creación de la interfaz de comunicación entre la base de datos y los usuarios.

La figura 1 muestra un aproximado del porcentaje de esfuerzo de ejecución de cada una de las actividades, en cada una de las fases generales de desarrollo, donde cada fase representa un 100% de esfuerzo. Las actividades de Implementación y ADTP tienen dependencia directa de la actividad Modelado de Datos, y todas dependen de la actividad Configuraciones. Las actividades de Implementación y ADTP suelen estar muy relacionadas por las tareas generales que definen, pero es conveniente dividir las lógicamente por el volumen de trabajo que suelen contener a medida que avanza el desarrollo.

Las actividades se definen según tareas generales de desarrollo, siendo la interrelación entre estas actividades el ciclo de vida y núcleo principal del framework. No es objetivo del framework definir las tareas generales de cada actividad. Definir con exactitud cada tarea que se deba realizar en el desarrollo de software no es práctico ya que cada sistema tiene sus particularidades. Sí es responsabilidad del equipo de desarrollo identificar qué elementos desarrollar y aplicar mejores prácticas. Las tareas que intervienen en cada actividad son aquellas identificadas por el equipo de desarrollo para lograr el objetivo de cada actividad.

El framework define los siguientes artefactos:

1. Arquitectura de Datos

2. Modelo de Datos
3. Configuraciones
4. Código Fuente
5. Scripts de Escenarios
6. Acceso a Datos

El framework asume 3 roles:

1. Arquitecto de Datos.
2. Administrador de Base de Datos.
3. Desarrollador de Base de Datos, incorporando a sus deberes el rol de Diseñador de base de datos.

El Arquitecto de Datos es el encargado de la realización de la actividad de Modelado de Datos y la elaboración de los artefactos Arquitectura de Datos y Modelo de Datos. Tiene mayor importancia en la fase de Inicio. El Administrador de Base de Datos es el encargado de la realización de la actividad de Configuraciones y la elaboración del artefacto Configuraciones. Tiene mayor importancia en la fase de Despliegue. El Desarrollador de Base de Datos es el encargado de la realización de las actividades de Implementación y ADTP, y la elaboración de los artefactos Código Fuente, Scripts de Escenarios y Acceso a Datos. Tiene mayor importancia en la fase de Desarrollo.

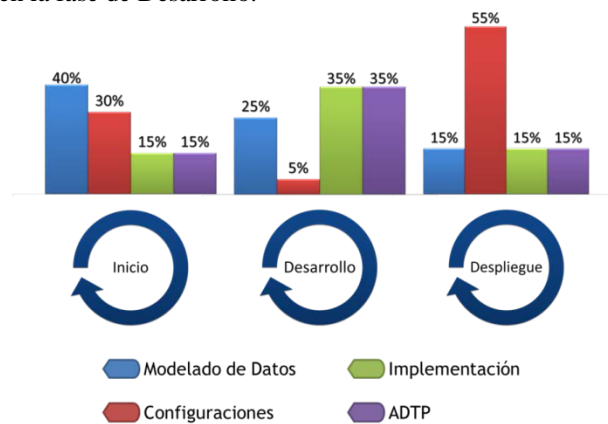


Figura 1: Actividades del framework

CONCLUSIONES

Mediante la aplicación de este framework se logra coordinar intencionalmente un grupo de elementos, siempre presentes en el desarrollo de bases de datos, con el objetivo de ganar en productividad y calidad de la base de datos que se desarrolle. Los artefactos, contrario a la opinión de muchos, son necesarios para el desarrollo de software y de gran importancia en el mantenimiento del mismo. Pero a su vez, no deben entorpecer el desarrollo. El desarrollo de este framework ha sido el resultado de varios años de experiencia en el desarrollo de bases de datos en proyectos de software, aplicando mejores prácticas desde el punto de vista de ingeniería y tecnológico.

REFERENCIAS

- Ambler, Scott W. 2003.** *Agile Database Techniques—Effective Strategies for the Agile Software Developer*. s.l. : John Wiley & Sons, 2003. 0-471-20283-5.
- Hassan, Ahmed M and Elssamadisy, Amr.** *Extreme Programming and Database Administration: Problems, Solutions, and Issues*.
- Oracle® Corporation. 2012.** Oracle Unified Method Knowledge Zone. [Online] Abril 13, 2012. <http://www.oracle.com/partners/en/knowledge-zone/applications/oracle-unified-method-359405.html>.
- Rational Software Corporation.** Database Designer. [Online] [Cited: Noviembre 21, 2011.] http://www.ts.mah.se/RUP/RationalUnifiedProcess/process/workers/wk_dbdsr.htm.