



Concepto de integración de software

El término "integración" hace referencia a una actividad de desarrollo de software que combina componentes de software diferentes en un conjunto. La integración se realiza en varios niveles y fases de la implementación.

- La integración del trabajo de un equipo que trabaja en el mismo subsistema de implementación antes de liberar el subsistema para los integradores del sistema.
- La integración de subsistemas en un sistema completo.

La propuesta de integración de Rational Unified Process consiste en integrar el software **en incrementos**. En la integración incremental, el código se escribe y se prueba en partes pequeñas que, a continuación, se añaden a un conjunto de trabajo de una en una.

La propuesta contraria a la integración incremental es la **integración por fases**. La integración por fases se basa en la integración de varios componentes (nuevos y cambiados) a la vez. El principal inconveniente de la integración por fases es que introduce muchas variables y dificulta la localización de errores. Esto se debe, principalmente, al hecho de que un error podría estar en cualquiera de los componentes nuevos, en la interacción entre los componentes nuevos en el centro del sistema o en la interacción entre los componentes nuevos.

Las ventajas de la integración incremental son:

- **Los errores son fáciles de localizar.** Cuando se produce un problema nuevo durante la integración incremental, el componente nuevo o cambiado, o su interacción con los componentes integrados anteriormente, son los lugares obvios para buscar un error. La integración incremental hace más probable que los defectos se descubran de uno en uno, lo que facilita la identificación de errores.
- **Los componentes se prueban de forma más completa.** Los componentes se integran a medida que se desarrollan y, después, se prueban. Esto significa que los componentes se ejercitan más a menudo que si la integración se realiza en un solo paso.

- **La ejecución se produce antes.** Los desarrolladores ven los primeros resultados del trabajo y no tienen que esperar hasta el final, lo que es mejor para su moral. Esto también hace posible obtener información de retorno antes.

Es importante comprender que la integración se produce, como mínimo, una vez **en todas las iteraciones**. Un plan de iteración define qué deben utilizar los guiones de uso para el diseño y qué clases se deben implementar. Lo principal de la estrategia de integración es determinar el orden de implementación y combinación de las clases.

Mantenimiento de software



En la ingeniería de software, el mantenimiento de software es la modificación de un producto de software después de la entrega, para corregir errores, mejorar el rendimiento u otros atributos. El mantenimiento de software es una de las actividades más comunes en la ingeniería de software.

El mantenimiento del software es también una de las fases del ciclo de vida del desarrollo del sistema (SDLC), que se aplica al desarrollo de software. La fase de mantenimiento es la fase que sigue al despliegue (implementación) del software en el campo.

Una percepción común del mantenimiento es que se trata simplemente de la corrección de defectos. Sin embargo, un estudio indicó que la mayoría, más del

80%, del esfuerzo de mantenimiento se utiliza para acciones no correctivas (Pigosky 1997). Esta percepción se perpetúa cuando los usuarios envían informes de problemas que en realidad son mejoras en la funcionalidad del sistema[cita requerida].