

Arquitectura Ágil

Índice

- ✓ Arquitectura de software
- ✓ Una necesidad
- ✓ El rol de Arquitecto
- ✓ Modularidad
- ✓ El equipo
- ✓ Reglas de buena documentación

Arquitectura de software

Arquitectura de software

- ✓ “Es la estructura o estructuras de un sistema definida por sus componentes, las relaciones entre los mismos y el conjunto de las propiedades externamente visibles” (Len Bass et al)
- ✓ Diseño estratégico
- ✓ Decisiones que deseamos acertar (Ralph Johnson)
- ✓ Lo que es difícil de cambiar (Martin Fowler)

Arquitectura de software

- ✓ Es tanto la actividad de diseño estratégico como parte del producto resultante.
- ✓ Puede no ser aparente.
- ✓ Afecta sensiblemente las características del producto.
- ✓ Sensible a la tecnología.
- ✓ Alto nivel de abstracción.

Una necesidad

Una necesidad

- Complejidad intrínseca
 - t No hay balas de plata (Brooks)
 - Flexibilidad como requisito mínimo
 - Escribir menos código (Poppendiecks)
- Tamaño creciente
 - Patrón de aumento constante del tamaño de los módulos (Clements y Shaw).
- Distribución
 - Generalizada en los '90
 - El número de usuarios se incrementa

El rol de arquitecto

- Referente técnico
- Con experiencias que aportar
- Mentor, guía y consejero para el equipo
- Con visión de negocio
- Responsable por los riesgos técnicos
- Aporta a la comunicación entre interesados
- Ayuda a definir la estructura del proyecto

Modularidad

- ¿Qué es un módulo?
- Estructura del producto en términos de asignación de trabajo (Len Bass et al)
- Permite disminuir las dependencias entre módulos y facilita trabajar en forma independiente.
- Pero, la pérdida de dependencias dentro de un equipo puede afectar el resultado eliminando la oportunidad para la interacción entre los que trabajan en módulos distintos (Devin)

Modularidad

- ✓ ¿Qué es un módulo hoy?
 - ✓ ¿Código?
 - ✓ ¿Procedimiento?
 - ✓ ¿Clase?
 - ✓ ¿Paquete?
 - ✓ ¿Aplicación?
 - ✓ ¿Unidad de trabajo? ¿Tarea?

Modularidad



- ✓ El tamaño de los módulos de un sistema depende del tamaño del sistema. La limitación en la cantidad de módulos la pone la mente humana.
- ✓ El tamaño de los módulos componentes aumenta en forma sostenida (Garlan y Shaw).

Modularidad: Criterios de diseño

- ✓ **Descomposición:** Divide et impera, dividir para conquistar.
- ✓ **Composición:** Reusar soluciones parciales preexistentes.
- ✓ **Comprensibilidad:** Independencia de las soluciones parciales para su comprensión

Modularidad: Criterios de diseño

- ✓ Continuidad: Se cumple si un pequeño cambio en el problema produce un pequeño cambio en la solución (análogo al concepto de localidad).
- ✓ Protección: Las condiciones anormales ocurridas sólo afectan a la solución parcial.

Modularidad



- ✓ Principio de unidades modulares lingüísticas:
El lenguaje de programación debe contener la sintaxis para describir módulos (Meyer).

El Equipo

- ✓ La arquitectura puede servir como base para entrenar nuevos miembros del equipo.
- ✓ Conviene establecer un lenguaje acotado de soluciones (patrones, frameworks, etc.)
- ✓ Conviene desarrollar un modelo único de análisis y solución (Domain Driven Design, Evans)

El Equipo

- ✓ El equipo y el cliente construyen un lenguaje único de la solución específica que desarrollan (Modelo de Dominio, Evans)
- ✓ La visión técnica guía el diseño produciendo un resultado armónico.
- ✓ El aporte de los integrantes del equipo es mejor que la solución base de compromiso: Reconcepción vs. Solución de compromiso
- ✓ El equipo disminuye la incertidumbre técnica.

Bibliografía

- ✓ Meyer, Bertrand, **Construcción de Software Orientada a Objetos**, Prentice-Hall, 1985, 2^{da}. Edición 1997.
- ✓ Software Architecture in Practice
- ✓ Documenting Software Architecture
- ✓ Artful Making

Muchas gracias