

Arquitectura de Software

Introducción

Contenidos

- ¿Qué es la Arquitectura del Software?
 - Arquitecto del Software
- La arquitectura en el ciclo de vida
- Documentar la arquitectura
 - ¿Para qué ? ¿Para quién?
- Vistas y Patrones
- Estándares y métodos

Definición en la bibliografía

- La organización fundamental de un sistema plasmada en sus componentes, sus relaciones (entre ellos y con el entorno) y los principios que guían su diseño y evolución

IEEE Architecture Working Group. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Standard IEEE Std 1471-2000, IEEE Standards Association, 2000

Definición en la bibliografía

- La estructura de estructuras de un sistema, formada por elementos, sus propiedades visibles y las relaciones entre ellos

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., and Stafford, J. Documenting Software Architectures: Views and Beyond. SEI Series in Software Engineering. Addison-Wesley, 2003

Definición en la bibliografía

- La estructura y la organización que permiten interaccionar a componentes y sistemas para formar otros sistemas, y las propiedades de estos que se diseñan y analizan mejor a nivel de sistema

Kruchten, P., Obbink, H., and Stafford, J. The past, present and future of software architecture. IEEE Software 23, 2 (2006), 22–30

Definición

- De acuerdo a las definiciones anteriores (y otras) la arquitectura de un software es su estructura fundamental, expresada como componentes, sus propiedades, y las relaciones entre ellos y con el entorno

Definición en la bibliografía

- La descripción de los subsistemas y componentes de un sistema software y las relaciones entre ellos

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. Pattern-Oriented Software Architecture. John Wiley and Sons, 1996 (p. 384)

Definición en la bibliografía

- Conjunto de reglas para definir la estructura de un sistema y las interrelaciones entre sus partes

ISO / IEC. Information technology - Open Distributed Processing – Reference Model: Foundations, first edition ed. International standard; ISO/IEC 10746-2. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 1996

Dos tipos de definiciones

- Según las tres primeras definiciones, la arquitectura es una propiedad de los sistemas
- De acuerdo con las últimas dos, es un artefacto del proceso de diseño

Definición

- Se define la arquitectura de software como una propiedad de los sistemas software
- Hablaremos de especificación o documentación de la arquitectura cuando nos refiramos al proceso de diseño que establece o describe esta arquitectura

Términos

- ¿Especificación, representación, descripción o documentación?
 - Views and Beyond: documentación
 - **AADL**: especificación
- Lo esencial es escribir y mantener actualizados los resultados de las decisiones arquitecturales tomadas de manera clara y accesible

Notaciones arquitecturales

- Informales: generalmente gráficas. Semántica en lenguaje natural
- Semiformales: notaciones estándar que prescriben elementos gráficos y reglas de construcción (p. ej. UML)
- Formales: Lenguajes de descripción de arquitecturas (ADL). Gráficas y/o textuales con semántica formal

Arquitectos de software

- ¿Quiénes son los arquitectos de software?
- El término se usa bastante para designar puestos de responsabilidad en desarrollo de software en algunas empresas
 - Microsoft, Apple, Oracle, Samsung, Adobe, Autodesk, IBM...

Arquitectos de software

- ¿Qué hace un arquitecto de software?
- Enlace/Mediador
 - Entre el cliente y el equipo de desarrollo.
Entre los diferentes equipos del proyecto
 - Justifica decisiones, costos q Gestores
 - Ayuda a promocionar q Ventas
- Ingeniero de software
 - Excelentes dotes para el Diseño. Promueve buenas prácticas de IS. Entiende el impacto de las decisiones de Diseño

Arquitectos de software

- Conocedor de las tecnologías
 - Evalúan y escogen las tecnologías y los productos de terceros
- Gestor de riesgos
 - Evalúan los riesgos, los gestionan junto al equipo directivo, desarrollan estrategias para mitigarlos

Arquitectos de software

- Independientemente de su título y de su puesto, al responsable de establecer y documentar la arquitectura de un software le denominaremos arquitecto de software
 - Tendrá el rol de “arquitecto” aunque su título sea p. ej. “ingeniero” y su puesto sea p. ej. “analista senior”

La arquitectura en el ciclo de vida del software

- Requisitos, análisis, diseño, implementación, pruebas...
 - ¿Dónde va la arquitectura?
- La arquitectura es diseño
- Pero no todo el diseño es arquitectural

Ingeniería del Sw

Análisis	Diseño
<p>El Análisis especifica lo QUÉ el sistema debe hacer.</p> <p>Se centra en el dominio de la aplicación.</p> <p>Realiza una descripción del sistema desde un punto de vista “lógico”.</p>	<p>El Diseño establece CÓMO alcanzar el objetivo.</p> <p>Se centra en el dominio de la solución.</p> <p>Realiza una descripción del sistema desde un punto de vista “físico”.</p>

Ingeniería del Sw (Diseño del Sistema)

1. Introducción
2. Definición de los objetivos de diseño
3. Identificación de subsistemas
 - 3.1. Subsistemas, servicios e interfaces de subsistema
 - 3.2. Capas y particiones
 - 3.3. Patrones arquitecturales
4. Refinar la arquitectura para conseguir los objetivos de diseño
 - 4.1. Concurrencia
 - 4.2. Correspondencia de subsistemas a procesadores y plataformas
 - 4.3. Gestión de datos persistentes
 - 4.4. Definición de control de acceso
 - 4.5. Diseño del flujo de control global
 - 4.6. Identificación de condiciones frontera

¿Arquitectura o diseño?

- Las reglas arbitrarias no son útiles:
 - La documentación de la arquitectura no puede ocupar más de X páginas
 - Ciertos diagramas UML son de arquitectura y otros de diseño
 - La arquitectura termina cuando empieza el diseño “detallado”
 - ...

Arquitectura y diseño

- Diseño arquitectural: decisiones para que el sistema alcance los objetivos marcados (comportamiento, calidad)
 - Típicamente imponen restricciones sobre el diseño posterior y la implementación
- Diseño no arquitectural: lo demás
 - P. ej., propiedades no visibles de un módulo (estructuras de datos...)

¿Cómo representamos la arquitectura?

- ¿Cuales son los principales elementos funcionales de tu arquitectura?
- ¿Cómo interaccionan estos elementos entre sí y con el mundo exterior?
- ¿Qué elementos hardware y software necesitaremos para dar soporte a estos elementos?
- ¿Qué características operacionales y capacidades se proporcionarán?
- ¿Qué entornos de desarrollo, pruebas, soporte, capacitación proporcionaremos?

¿Cómo representamos la arquitectura?

- Son muchas preguntas para que un único modelo pueda dar respuesta a todas ellas
 - Si un sistema es complejo, su arquitectura lo será también
- La estrategia para afrontar esta complejidad es dividir la arquitectura de un sistema en diferentes *vistas*

Vistas

- Una vista es una representación de un conjunto de elementos del sistema y sus relaciones
 - Cada vista trata aspectos diferentes
 - Las vistas están interrelacionadas

Documentar una arquitectura consiste en documentar las vistas relevantes y luego documentar lo que atañe a varias vistas

Ejemplos de Vistas

- Contexto

- Describe las relaciones, dependencias e interacciones entre el sistema y su entorno (la gente, otros sistemas, ...)

- Funcional

- Describe los elementos funcionales del sistema, sus responsabilidades, interfaces e interacciones básicas

- Información

- Describe la forma en la que el sistema almacena, manipula, gestiona y distribuye la información

Ejemplos de Vistas ...

- Concurrency

- Describe la estructura de concurrencia del sistema y la hace corresponder con los elementos funcionales

- Desarrollo

- Describe la arquitectura que da soporte al proceso de desarrollo de software

- Despliegue

- Describe el entorno en el que se despliega el sistema y las dependencias de sus elementos con los de ese entorno

- Operacional

- Describe cómo se operará, administrará y se dará soporte al sistema en ejecución cuando esté en el entorno de

Pero ¿se debe documentar la arquitectura?

- Documentar cuesta tiempo (dinero)
 - Debe proporcionar beneficios
- Muchas actividades costarán más sin documentación arquitectural
 - Implementar, lanzar una nueva versión, localizar problemas que surgen en producción...
- Por ahí llegan los beneficios

¿Para qué y para quién?

- La base para el análisis y la construcción del sistema
 - Dice a los implementadores lo que tienen que implementar, contiene información para evaluar objetivos de calidad del sistema (seguridad, prestaciones...), puede incluir modelos para la generación automática de código

¿Para qué y para quién?

- Educación
 - Nuevos miembros del equipo, analistas externos, incluso un potencial cliente
- Comunicación
 - El arquitecto “del futuro”, el sitio para registrar y recordar decisiones fundamentales de diseño, un mecanismo para medir avances

¿Cuánto se debe documentar la arquitectura?

- El arquitecto decide cuánto se documenta. Depende de:
 - Los interesados (*stakeholders*), cuántas vistas necesitan, en qué formatos les serán más útiles...
- Común hasta en proyectos ágiles (<http://goo.gl/jxTVM>)
 - En Spotify: <http://goo.gl/t8B0h>

Proyectos muy dinámicos

- Documentar lo que no cambia (invariantes arquitecturales)
 - Serán más bien restricciones y guías
- Documentar mecanismos permitidos para cambiar la arquitectura
- Auto-documentar la arquitectura
 - P.ej. Log de los servicios web conectados en cada momento

Estilo arquitectural

- Un estilo arquitectural es una especialización de elementos y tipos de relaciones, junto con un conjunto de restricciones sobre cómo pueden ser usados
- A la documentación que describe en detalle un estilo la llamaremos guía de estilo

Patrones arquitecturales

- Hay soluciones arquitecturales recurrentes que aparecen en sistemas completamente distintos
 - Cliente/Servidor, Modelo-Vista-Controlador...
- Es importante conocer y aplicar estas soluciones comunes

Patrón arquitectural vs Patrón de diseño

- Patrón de diseño

- Proporciona un esquema para refinar un sub-sistema o un componente de un sistema software. Describe una estructura recurrente de componentes que se comunican para resolver un problema de diseño general en un contexto particular

- Patrón arquitectural

- Expresa una estructura fundamental para la organización de un sistema software. Proporciona un conjunto de sub-sistemas predefinidos, sus responsabilidades e incluye reglas y guías para organizar las relaciones entre ellos

Estilos arquitecturales

- La elección de estilos depende del sistema y de los interesados
 - Algunos estilos son aplicables a cualquier sistema software (p. ej. descomposición en módulos), otros se eligen explícitamente (p. ej. orientado a servicios)
 - Ningún sistema no trivial se construye sólo con un estilo

Estilos y patrones

- Los conceptos de estilo y patrón arquitecturales son similares (a veces se usan como sinónimos)
- Nosotros consideraremos que un patrón arquitectural parte de un problema y un contexto determinados y sugiere un estilo arquitectural para solucionarlo

Arquitectura y calidad

- Los patrones y estilos arquitecturales tienen atributos de calidad asociados
 - P. ej. el estilo cliente/servidor favorece la escalabilidad
- Los elementos arquitecturales (p.ej. un componente) pueden tener restricciones de calidad
 - P. ej. el servicio X debe responder en menos de T ms.

Siete reglas para una buena documentación (I)

- 1. Escribir desde el punto de vista del lector: conocer al lector, ser ordenado, evitar la jerga y el abuso de acrónimos
- 2. Evitar repeticiones innecesarias
- 3. Evitar la ambigüedad
 - 3a. Explicar la notación

Siete reglas para una buena documentación (II)

- 4. Usar una organización estándar
- 5. Incluir la exposición de razones
- 6. Mantener la documentación actualizada, pero no demasiado
- 7. Revisar la documentación para ver si cumple su propósito

Estándares y métodos arquitecturales

- Hay bastantes enfoques, algunos estandarizados internacionalmente, para diseñar, especificar y documentar arquitecturas de software
 - Generalmente basados en vistas
 - Difieren en conceptos, notaciones, tipos de vistas, en filosofía del proceso de desarrollo etc.

Estándares y métodos arquitecturales

- ISO/IEC 42010:2007 (a partir del ANSI/IEEE Std 1471-2000)
- RUP 4+1
- ISO/ITU-T RM-ODP
- Views and Beyond
- ...

ISO/IEC 42010:2007

- La arquitectura de un sistema de interés tiene una descripción, que incluye una o más vistas (desde diversos *viewpoints*) que cubren lo requerido por los interesados (*stakeholders*). Cada vista incluye modelos arquitecturales. Se incluye la exposición de razones (*rationale*)

Rational Unified Process/Kruchten 4+1

- 5 vistas: lógica (clases de diseño fundamentales), de implementación, de procesos, de despliegue y de casos de uso (la vista "+1")
- No prescribe documentación de interfaces, exposición de razones, mapeos entre vistas, guías de estilo etc. pero tampoco los prohíbe

ISO/ITU-T RM-ODP

- Un marco arquitectural para diseñar sistemas distribuidos, especialmente si son complejos
- Una norma posterior establece como expresar las arquitecturas en UML
- No prescribe un proceso concreto
 - Se puede usar dentro de cualquier metodología de desarrollo

ISO/ITU-T RM-ODP

- Basado en conceptos rigurosamente definidos de orientación a objetos
- 5 puntos de vista (*viewpoints*): de misión⁽¹⁾ (*enterprise*), de información, de ingeniería, de tecnología y computacional

(1) O de empresa, pero en el sentido de “Acción o tarea que entraña dificultad y cuya ejecución requiere decisión y esfuerzo”

Views and Beyond

- La documentación de la arquitectura incluye vistas y la información que se aplica a varias de ellas a la vez
 - El arquitecto determinará las vistas según el sistema y los interesados
- No prescribe un proceso concreto
- No prescribe una notación, pero siempre indica como usar UML

Views and Beyond

- Propone 17 estilos arquitecturales agrupados en tres tipos: de módulo, de componente-y-conector y de distribución (*allocation*)
- Una vista típica se basará en uno de estos estilos
 - También hay un estilo híbrido si fuera necesario mezclar varios

Documentación de una vista (I)

- Una representación primaria (típicamente gráfica) con los elementos y relaciones principales
- Un catálogo de los elementos con definiciones y propiedades

Documentación de una vista (II)

- Especificación de la interfaz y el comportamiento de los elementos
- Guía de variabilidad, explicando como adaptar la arquitectura
- La exposición de razones (*rationale*) y la información de diseño

Documentación que atañe a todas las vistas

- Introducción a la documentación de la arquitectura y guía para interesados
- Descripción de cómo las vistas se relacionan entre si y con el sistema
- Restricciones y exposición de razones que atañen a toda la arquitectura
- Datos de gestión del documento