

Bloque 1 - Arquitectura de software, Diseño Arquitectónico y Vistas de Arquitectura

Contents

Lesson 1 - Arquitectura Software	2
1.1 Arquitectura aplicada a software	2
1.1.1 Decisión y Balance	2
1.2 Ciclo de vida de la Arquitectura	2
1.3 Aspectos importantes	2
1.4 Roles de la Arquitectura Software	2
Lesson 2 - Diseño de Arquitectura	3
2.1 Modelado de arquitectura	3
2.2 Tipos de modelos	3
2.3 Requisitos significativos de arquitectura (Architecture Significant Requirement), ASR	3
2.4 Business Goals	3
2.5 Diseño de estrategias	3
2.6 Diseño basado en atributos (Attribute Driven Design), ADD	4

Lesson 1 - Arquitectura Software

1.1 Arquitectura aplicada a software

Def.- La arquitectura software de un sistema son las estructuras del sistema, los elementos software, las propiedades visibles externas y las interacciones entre todo ello

La arquitectura no es algo nuevo, describe cuales las características que hacen que un diseño sea bueno:

- Durabilidad
- Utilidad
- Atractivo

Esto mismo se puede aplicar al software, para que se realice un buen diseño tiene que ser:

- Fácil de implementar
- Entendible
- Realizable
- Evolutivo

1.1.1 Decisión y Balance

Decisión

Una decisión temprana es fundamental para una buena evolución del software las cosas que se tienen que tener en cuenta:

- Restricciones del sistema
- El sistema se descompone en componentes
- Los componentes interaccionan entre si

Decisión de diseño

En la decisión de diseño lo que realizaremos mas tarde es decir que atributos de calidad son necesarios y cuales no, cual es la estructura de la organización y las restricciones de implementación

Balance

Debemos de conseguir balance entre Requisitos y Calidad para conseguir el mejor resultado

1.2 Ciclo de vida de la Arquitectura

La Pre-Arquitectura es el inicio del ciclo de vida de la arquitectura software, basada en requisitos funcionales, estos requisitos funcionales se realiza varias iteraciones de refinamiento para luego poder derivar los requisitos del sistema en la estructura del sistema. Lo que realiza la arquitectura es diseñar el sistema
El flow del ciclo de vida es el siguiente:

1. Recogida de Requisitos y QA ´s de los stakeholders
2. Se realizan varias iteraciones para el refinamiento de los requisitos
3. Acuerdo entre Requisitos y QA ´s
4. Desarrollo - Diseño

- I Desarrollo - pasos:
- i. Arquitectura
 - ii. Diseño detallado
 - iii. Implementación

1.3 Aspectos importantes

Importancia de la arquitectura

La cosas mas importantes a tener en cuenta es que la arquitectura es el vehículo para la comunicación con los stakeholders, el manifiesto arquitectónico es un primero conjunto de decisiones de diseño y lo que se busca en la arquitectura es la abstracción de un sistema.

Frameworks

1. Conceptual framework - nivel de ideas
2. Software framework - nivel de software: los programas, librerías y lenguajes
3. Design framework - nivel de componente: entorno lógico para el marco de elementos

1.4 Roles de la Arquitectura Software

Los clientes, usuarios, ... y arquitectos se comunican con los requisitos, los desarrolladores y arquitectos se comunican con soluciones. Los arquitectos son los que crean el diseño de arquitectura, con esta permite que los usuarios visualicen como va a ser la apariencia y el comportamiento, y a los desarrolladores les permite observar cual va a ser la construcción y cooperación a realizar.

Lesson 2 - Diseño de Arquitectura

2.1 Modelado de arquitectura

Un modelo arquitectónico es un artefacto que captura algo o todo el diseño. El modelo debe de estar basado sobre los requisitos de calidad que hemos extraído de los stakeholders. Entre los principales elementos del modelo están:

- Componentes
- Interfaces
- Conectores

2.2 Tipos de modelos

Estático vs Dinámico

- Estático - Mientras que el sistema esta en ejecución no cambia el aspecto del sistema
- Dinámico - El aspecto del modelo cambia de aspecto en ejecución

Funcional vs No-Funcional

- Funcional - Capacidades del sistema y que cosas quiere el usuario que realice el sistema
- No-Funcional - Restricciones de contexto y diseño, y interfaces externas

Características de los modelos

Los modelos no deben de ser ambiguos, para que no den a mas de una linea de interpretación, deben de ser correctos, conforme con los requisitos y preciso/exacto

2.3 Requisitos significativos de arquitectura (Architecture Significant Requirement), ASR

Def.- Los requisitos específicos de arquitectura son los requisitos que tienen gran impacto sobre la arquitectura y deben de estar alineados con los objetivos de negocio (Business Goals), los repositorios de requisitos o los stakeholders

ASR desde los Repositorios de Requisitos

En el documento deberemos extraer la asignación de responsabilidades, la coordinación con otros modelos, el modelos de los datos, la administración de los recursos, el mapa de elementos del sistema, tiempo de decisión y la elección de las tecnologías

ASR desde los Stakeholders

Tenemos que encontrar cuales de los stakeholders que vamos a poder utilizar para extraerles información que nos interese para la arquitectura ya que solo entienden de QA ´s y no de ASR. Una de las dinámicas que se pueden utilizar son los workshops donde intentaremos extraer los accionadores arquitectónicos (Architectural drivers) y los escenarios del sistema. Para los "drivers" utilizaremos un prototipo inicial para que puedan identificar cuales son los "drivers" que creen necesarios, para los escenarios podremos utilizar brainstorming para después ver que todos estén de acuerdo, puedan priorizar lo mas importante y se pueda refinar

ASR desde los Business Goals

Los business goals son las razones por las cuales se debe de construir el sistema, es interesante saber cual es el impacto del sistema y cuales son sus restricciones, esto no hace falta formalizarlo.

2.4 Business Goals

Para la representación de los Business Goals se construye un árbol de utilidades (Utility Tree) es la forma de recoger todos los ASR's en un lugar, cada uno de los registros se establece una prioridad de ASR por medio de el impacto en la arquitecta y el valor de negocio del ASR, cada uno de estos ASR son capturados como escenarios.

Utility Tree

La raíz del utility tree es un nodo que se llama Utility, el segunddo nivel contiene las categorias de los QA y en el utlimo nivel encontramos las categorias refinadas. Quedaria de esta manera:

<u>Quality Attribute</u>	<u>Attribute Refinement</u>	<u>ASR</u>
--------------------------	-----------------------------	------------

2.5 Diseño de estrategias

Las principales fases del diseño de estrategias son:

1. Descomposición - los atributos de calidad, QA ´s, provocan un gran impacto en el sistema como un todo, por ello para reducir la complejidad se dividen en partes que heredan del QA, esta descomposicion afecta al diseño
2. Designar ASR - en este paso se cogen los requisitos que se quieren realizar y se idea en que orden debería de ir y que relevancia tienen, en esta parte no consideramos ningún otro QA
3. Generación y Test - Consideramos la hipotesis que hemos tomado como resultado del proceso como correcta y se testea que satisfaga todos los requisitos, si lo satisface es nuestro resultado y si no los satisface se vuelve a testear

Lesson 2 - Diseño de Arquitectura

Proceso de arquitectura

Se inicia con la información extraída de los stakeholders como input, estos se capturan en los QA´s, de los QA´s se extraen los ASR´s que serán representados en un Utility Tree, con todo esto se realizara el ADD que dará como resultado la Documentación de la Arquitectura.

2.6 Diseño basado en atributos (Attribute Driven Design), ADD

Método iterativo, cada iteración se compone de elegir la parte a diseñar, recabar todos los ASR de esta parte y generar y testear esta parte del diseño, estos pasos se realizan hasta que se pasa por todos los ASR´s. El método ADD no da resultado en el diseño completo, lo que proporciona es un conjunto de restricciones con responsabilidades y como interactúan los contenedores con el flujo de información. Se especifican dos espacios el espacio del problema donde dividimos el problema en subproblemas y el espacio de decisión donde se recaban todas las opciones de decisiones que se pueden tomar y se enlazan con los subproblemas. Además las estas opciones se agrupan en soluciones alternativas y se tiene que seleccionar cual es la mejor opción

Tipo de decisiones

- Implícitos/No documentados
- Explícitos/No documentados
- Explícitos/Explicitados No documentados
- Explícitos/Documentados