Arquitectura de software, ADS

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

Contents

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 Desarollo - 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, librearías 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 brainstroming 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 raiz 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:

Quality Attribute Attribute Refinement ASR

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 descompocicion 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 los 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

2.7 Atributos de calidad, QA

Atributos de calidad, QA

Def.- Los atributos de calidad son una medida o testabilidad de las propiedades del sistema que esta en uso para indicar como de bien el sistema satisface las necesidades de los stakeholders

Entre las principales categorías son las propiedades del sistema, propiedades de los desarrolladores y las propiedades del negocio

Atributos de calidad especificos, SQA

Los atributos de calidad específicos deben de ser no ambiguos y testables, entre los tipos que hay están:

- Estímulos - son eventos que llegan al sistema
- Fuente de estímulos - de donde se emite el evento o estimulo
- Respuesta - como el sistema responde ante los estímulos
- Medida de respuesta - determina como la respuesta es de satisfactoria

Los ciclos de estos elementos es el siguiente, se produce un estimulo desde la fuente pasa por el entorno donde stan los artefactos que generan una respuesta que se puede tomar y medir

Escenarios QA

Los QA´s se representan como escenarios y los escenarios son tablas donde se relacionan los elementos con su descripcion. Los principales tipos de escenarios son:

- General - el sistema es independiente y su representacion pertenecea cualquier sistema
- Concreto - es dependiente del sistema y su representación es especifico de un caso de estudio concreto

Recopilacion de QA´s

Principal clasificación de QA´s:

- Sistema

- Disponibilidad - Availability
- Interoperabilidad - Interoperability
- Rendimiento - Performance
- Seguridad - Security
- Usabilidad - Usability

- Negocio

- Comerciabilidad - Marketability
- Coste/Beneficio
- Linea del tiempo del proyecto
- Integración con sistemas legados
- Planificación de Roll Back

- Desarrollo

- Modificabilidad - Modifiability
- Testabilidad - Testability
- Portabilidad - Portability
- Mantenibilidad - Mantainability
- Reusabilidad - Reusability

Lesson 3 - Vistas de Arquitectura

3.1 Introducción

Conceptos básicos

Entre los lenguajes que existen para las arquitecturas esta UML que sirve para modelado de arquitecturas, el problema de estos lenguajes es que no todo el mundo los entiende. Por lo que para la representación de arquitecturas habrá que utilizar tanto UML como otro sistema que nos permita explicar de una manera mas sencilla lo que se va a realizar. No hay una mejor aplicación para representar las arquitecturas pero debe de ser concisa y clara. Además, es imposible representar toda la arquitectura mediante un solo diagrama lo que no permite poder personalizar el tipo de diagrama en función de las personas a las que nos queremos dirigir con el mismo

Elementos

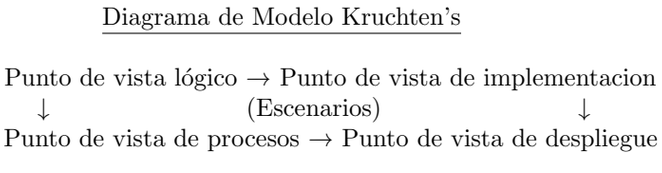
- Punto de vista (Viewpoint) - colección de patrones, plantillas y conversiones para la construcción de un tipo de vista. Es definido por los stakholders a los que involucre el punto de vista o linea guía, principio y plantilla para la construcción del modelo
- Especificación del punto de vista (Viewpoint specification) - nombre del punto de vista, stakeholders a los que involucra, el problema a resolver y las técnicas de modelado y lenguaje usado
- Vista (View) - representación de una o mas aspectos de una estructura que muestra como es la arquitectura respecto al punto de vista de las preocupación de unos stakeholders
- Metafores (Metaphors)

Vistas comunes usadas

- Punto de vista lógica - este punto de vista soporta los requisitos funcionales y los servicios de los que provee el sistema. Normalmente, se muestra el sistema descompuesto en un conjunto de abstracciones (clases y sus interacciones), utilizando de esta manera los principios de abstracción, encapsulación y herencia. Además hay que tener en cuanta también los servidores para identificar los mecanismos comunes y diseñar elementos entre varias partes del sistema
- Punto de vista físico - Captura las entidades físicas (hardware) del sistema y como están interconectadas
- Punto de vista de despliegue (deployment) - Captura como las entidades lógicas son mapeadas sobre las entidades físicas, se definen elementos de las vistas lógica, de proceso y de implantación respecto a las redes, proceso y tareas. Los objetos son mapeados dentro de varios nodos.
- Puntos de vista de la concurrencia (concurrency) - Captura como la concurrencia y el threading va a ser administrado por el sistema
- Punto de vista de comportamiento (behavioral) - Captura el comportamiento que se espera de parte o todo el sistema
- Punto de vista de proceso - Se representan los diferentes aspectos de concurrencia en tiempo de ejecución, además de algunos requisitos no funcionales. Su utiliza como primera abstracción de la vista lógica dentro de la arquitectura de proceso
- Punto de vista de implementación - Centrado en la organización delos módulos software dentro del entorno, como las librerías o los subsistemas, su organización se realiza de manera jerarquizada en capas y entre las capas se definen interfaces para su comunicación. Además se incluyen algunos requisitos no funcionales
- Punto de vista de escenario - Pequeño subconjunto de escenarios importantes que se muestran sus elementos para las otras vistas al mismo tiempo, es redundante con los puntos de vista anteriores, pero son críticos para el desarrollo de la arquitectura

3.2 Kruchten's view Model

El modelo Krutchen's se basa en las anteriores vistas descrita, en el centro colocamos todos los escenarios y a cada uno de los lados las siguientes vistas relacionadas:



3.3 Tipos de inconsistencias

- Inconsistencias directas
- Inconsistencias de refinamiento
- Inconsistencias en aspectos estáticos vs dinámicos
- Inconsistencias entre aspectos dinámicos
- Inconsistencias funcionales y no funcionales