ARQUITECTURA Y DISEÑO DE SISTEMAS

ARQUITECTURA DE SOFTWARE – PARTE 1

ELSA ESTEVEZ
UNIVERSIDAD NACIONAL DEL SUR
DEPARTAMENTO DE CIENCIAS E INGENIERIA DE LA COMPUTACION

CONTENIDO

MOTIVACIÓN POR ANALOGÍA – METODOLOGÍA

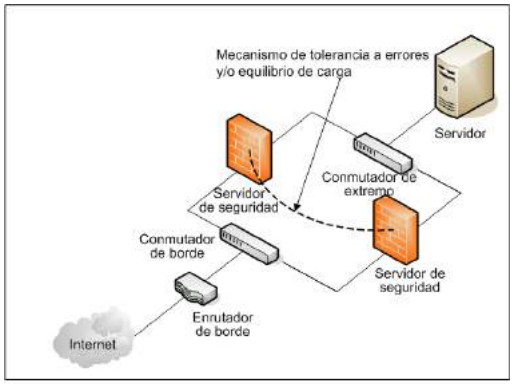
CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ Relevar requerimientos para la construcción	○ Relevar requerimientos para el sistema
○ Crear un diseño para satisfacer los requerimientos	○ Crear un diseño de alto nivel
○ Refinar el diseño para producir planos elaborados	○ Desarrollar los algoritmos en base a aquel diseño
○ Construir el edificio en base a esos planos	○ Escribir el código para implementar los algoritmos
○ El edificio es habitado y usado	○ Instalar el sistema para que luego sea usado.

MOTIVACIÓN POR ANALOGÍA – ARQUITECTURA

CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ El proceso arquitectónico hace foco en las necesidades de los futuros ocupantes	○ La construcción de software comienzo y enfatiza la especificación de los requerimientos
○ Quien diseña la estructura no necesariamente tiene que ser el contratista que lleva a cabo la obra	○ El diseño es creado por especialistas. La programación se puede tercerizar
○ El proceso tiene puntos intermedios donde hacer revisiones y medir progreso	○ Prototipos y mock-ups creados durante el desarrollo le permiten al cliente periódicamente verificar si el sistema cubre las necesidades

MOTIVACIÓN POR ANALOGÍA – ANÁLISIS 1

CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ La arquitectura es un concepto separado pero inexorablemente ligado a la construcción	○ Las principales decisiones de diseño (arquitectura) existen independientemente del código que la implementa
○ La propiedades de las estructuras son inducidas por el diseño de su arquitectura	○ Propiedades de aplicaciones de software son determinadas por su arquitectura



MOTIVACIÓN POR ANALOGÍA – ANÁLISIS 2

CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ Los arquitectos requieren un entrenamiento amplio. No basta con conocimientos técnicos.	○ Para un arquitecto no es suficiente con tener fuertes aptitudes de programación para crear software que sea útil para las personas
○ Los procesos no son tan importantes como la arquitectura	○ Los procesos están para servir a los fines del software a construir
○ La arquitectura es una disciplina milenaria, y el conocimiento adquirido es reutilizado y refinado constantemente. Los estilos arquitectónicos así lo demuestran: gótico, cabaña suiza, y otros.	○ Está madurando rápidamente como una disciplina robusta, donde el conocimiento compartido toma forma de estilos arquitectónicos y patrones de diseño.

MOTIVACIÓN POR ANALOGÍA – LIMITACIONES 1

CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ Tenemos un conocimiento básico de la construcción, lo cual ayuda a desarrollar la intuición en el tema.	○ La intuición no necesariamente está desarrollada: Hay que ser más metódico y analítico
○ La construcción es tangible y se puede evaluar sólo mirándola	○ El software es intangible, siendo más difícil de evaluar progreso y calidad.
○ Los cambios no son tan sencillo de hacer 	○ El software es mucho más maleable y permeable a cambios

MOTIVACIÓN POR ANALOGÍA – LIMITACIONES 2

CONSTRUCCIÓN DE EDIFICIOS	CONSTRUCCIÓN DE SOFTWARE
○ La industria es mucho más estructurada	○ La industria, si bien tiene sus estructuras, no está tan avanzada
○ Se construye sobre un lugar	○ Se construye una vez y se puede instalar muchas.
○ Las construcciones no son una máquina.	○ El software es una máquina, tiene un carácter dinámico

IDEA PRINCIPAL

La arquitectura de software debe ser el núcleo del diseño y desarrollo de un sistema de software.

- Debe estar siempre en el primer plano
- No solamente al principio del proyecto, sino durante todo el ciclo de vida, especialmente para sistemas con una larga vida útil.
- Para aplicaciones complejas o grandes, la arquitectura debe ser considerada de antemano
- Dándole importancia a la arquitectura, se podría lograr …
  - ✓ Integridad conceptual
  - ✓ Una adecuada y efectiva base para el reuso (conocimiento, experiencia, diseños y código)
  - ✓ Comunicación efectiva

CONTENIDO

DOS CASOS DE ESTUDIO

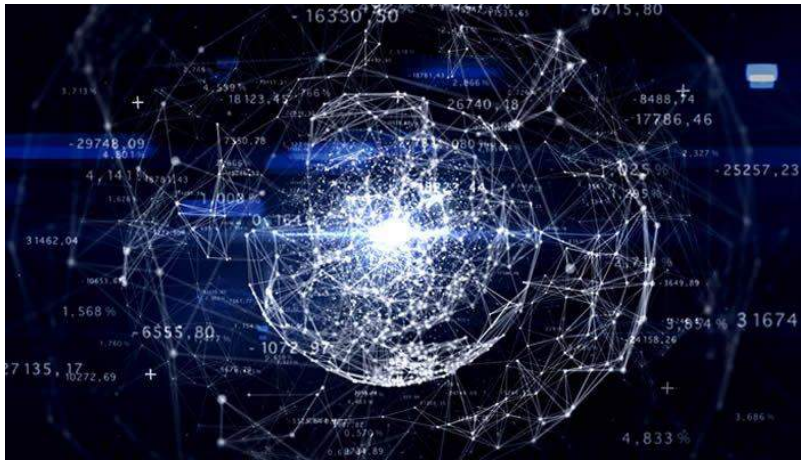
1) LA WEB

2) LíNEAS DE PRODUCTO

LA WEB

En informática, la World Wide Web (WWW) o red informática mundial es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet.

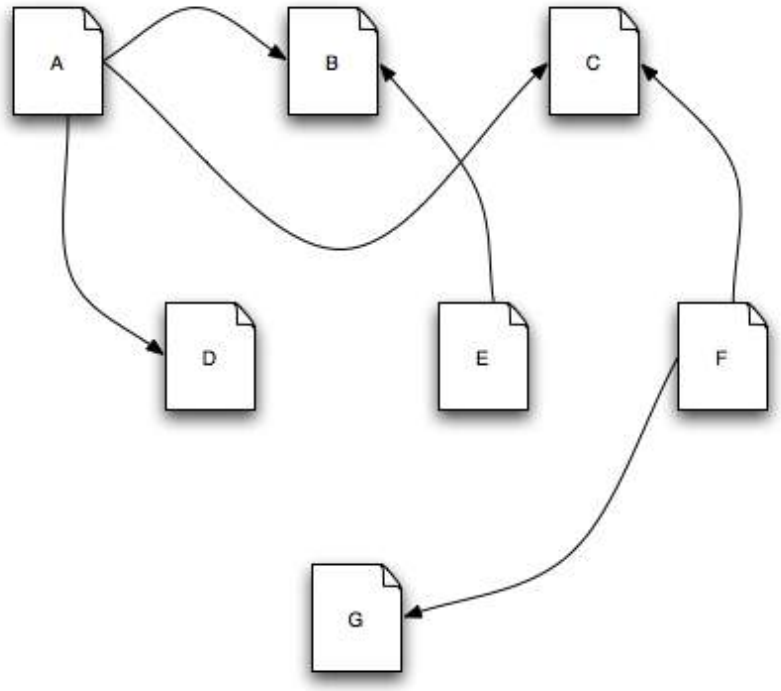
Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces."



Ref: Wikipedia

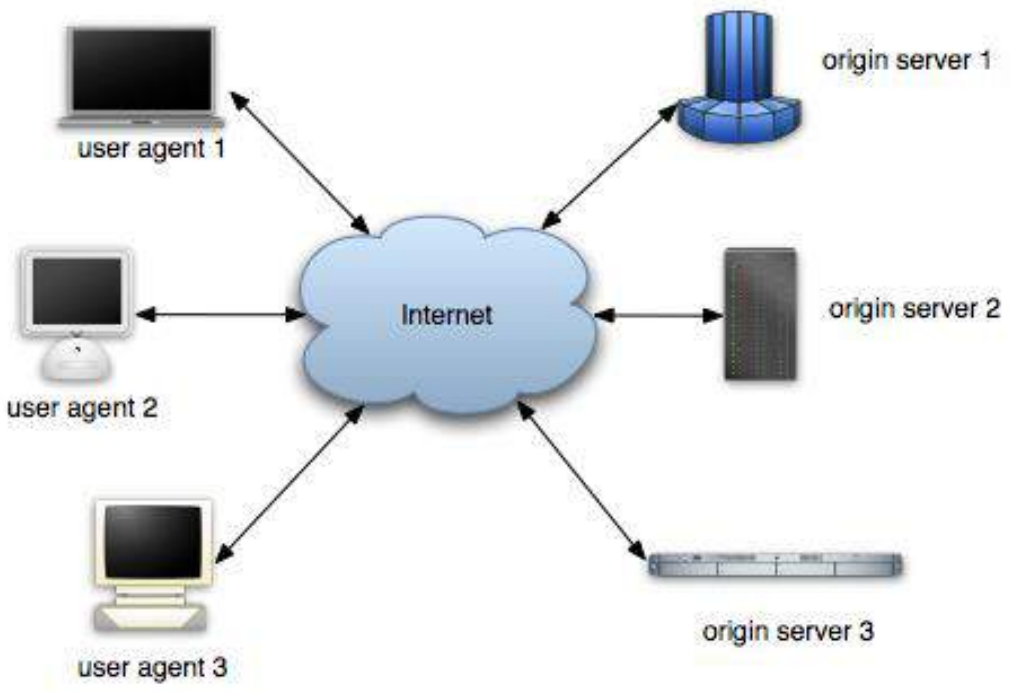
LA WEB – PERSPECTIVA DEL USUARIO

Un conjunto dinámico de relaciones entre colecciones de información.



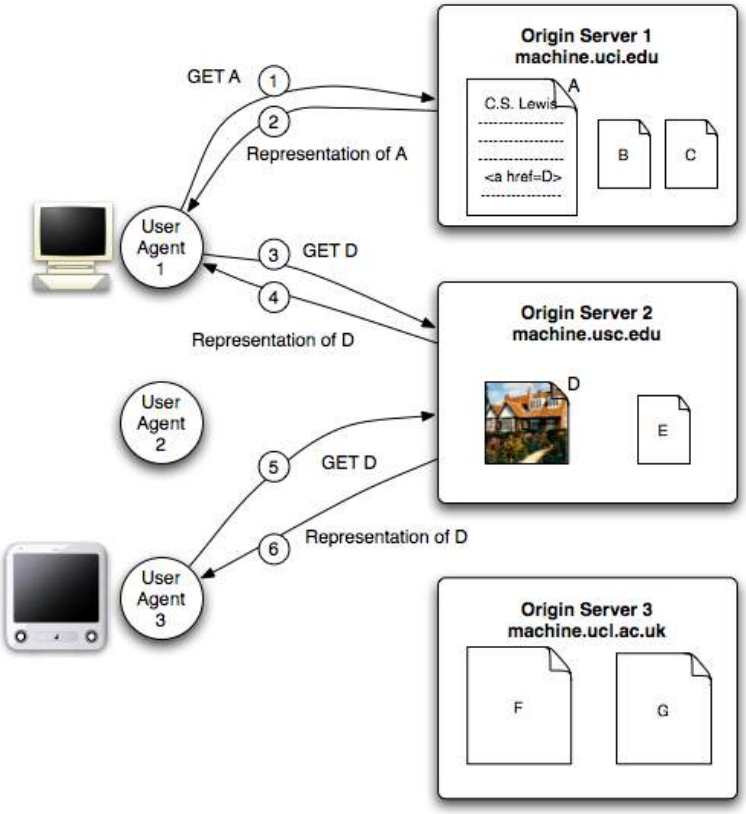
LA WEB – PERSPECTIVA DEL HARDWARE

Una colección de máquinas alrededor del mundo, propietarias e independientemente operadas, que interactúan entre sí a través de redes de computadoras



LA WEB – PERSPECTIVA DEL DESARROLLADOR

Una colección de programas escritos independientemente que interactúan entre sí de acuerdo a las reglas especificadas en los estándares HTTP, URI, MIME, HTML, etc.



LA WEB – DISCUSIÓN

¿Podemos decir que estos diagramas realmente explican cómo funciona la Web?

No… pero podemos mejorarlo si agregamos definiciones y restricciones:

- La Web es una colección de recursos, identificados por una URL
- Cada recurso denota, informalmente, alguna información
- URLs pueden ser usadas para determinar la identidad de una máquina en Internet (*origin server*)
- La comunicación es iniciada por los clientes (*user agents*), que hacen requerimientos sobre los servidores
- Los recursos pueden ser manipulados a través de sus representaciones
- Todas las comunicaciones entre *user agents* y *origin servers* deben ser realizadas por un protocolo simple y genérico: HTTP
- Todas las comunicaciones entre *user agents* y *origin servers* deben ser auto-contenidas (*stateless*).

LA WEB Y SU ARQUITECTURA

Describir las reglas por las cuales las distintas partes de la Web funcionan e interactúan (su ***estilo arquitectónico***) provee las bases para entender la web independientemente de su configuración e implementación específica.

Una de las aplicaciones más exitosas solamente es entendida adecuadamente desde la visión panorámica de la arquitectura.

LA WEB – CONCLUSIONES

- La arquitectura de la Web está totalmente separada del código que implementa a cada uno de sus elementos.

  La arquitectura es un conjunto de las principales decisiones de diseño que determinan los elementos clave de la Web y sus interrelaciones. Estas decisiones son un nivel de abstracción por encima del código.

- No hay una única pieza de código que implemente la arquitectura

  La Web está implementada por servidores de distinto diseño, browsers de distinto diseño, proxies, routers, etc.

- El estilo arquitectónico de la Web restringe al código en algunos aspectos (por ejemplo, en cómo debe interactuar con otros componentes), pero da libertad para la codificación interna del componente (por ejemplo, distintos navegadores)

- Las restricciones que constituyen la definición del estilo de la Web no necesariamente son evidentes en el código, pero sus efectos son evidentes en la Web

LíNEAS DE PRODUCTOS – PROBLEMA

MOTIVACIÓN

El mercado de electrónicos demanda distintas configuraciones de televisores:

Cada televisor podría contener un millón de líneas de código

DESAFÍO ECONÓMICO

- Producir un amplio rango de productos que el mercado global de consumidores sofisticados demanda

- Explotar simultáneamente las características comunes entre miembros de una familia de productos

LíNEAS DE PRODUCTOS – ESTRATEGIA

ESTRATEGIA

- Construir cada TV desde cero, sacaría a Philips fuera del negocio
- Reusar estructuras, comportamientos e implementación de componentes será cada vez más importante para el negocio, ya que:
  - ✓ Simplifica las tareas de desarrollo de software
  - ✓ Reduce el tiempo y costo de desarrollo
  - ✓ Mejora la confiabilidad del sistema

Ref: http://www.philips.es/c-m-so/televisores

LíNEAS DE PRODUCTOS – SOLUCIóN

SOLUCIÓN

- Phillips creó una familia de productos
- Variaciones: precio, salida, región
- Comparten mucha de la funcionalidad
- Crearon una tecnología (Koala) para explotar las características comunes y las variaciones
- Adaptaron sus procesos y prácticas organizacionales a las necesidades de construcción de líneas de productos

PHILIPS LíNEA DE PRODUCTOS – DISCUSIóN 1

PRODUCT LINE HALL OF FAME

Philips Product Line of Software for Television Sets

Philips is one of the world's largest consumer electronics companies, and a global leader in televisions and other consumer products. While initially solely consisting of hardware, TVs now contain fully equipped embedded computers to control the hardware and to implement extra features. These computers started small, with 1 kilobyte of code around 1980, but software size has grown according to Moore's Law, resulting in many million lines of code today. While this by itself makes *complexity* an important issue, the other important issue is *diversity*, since televisions are produced in many different variants. To efficiently manage complexity and diversity, we decided to set-up a software product line.

Ref: http://splc.net/fame/philips-television.html

PHILIPS LíNEA DE PRODUCTOS – DISCUSIóN 2

PRODUCT LINE HALL OF FAME

Philips Product Line of Software for Television Sets

The first step was the definition of a *software component model*, Koala, designed to allow the creation of different products by making new combinations of selections of reusable components.

For this, all context dependencies of components are made explicit in the form of provides and requires interfaces, while diversity interfaces allow for internal variation of components.

An explicit architectural description language, one of the first to be used extensively in industry, helps to manage complexity and to automatically generate product specific binding code. While Koala was modeled after Darwin and Microsoft COM, it is specifically tuned to build software product lines of resource-constrained products.

Ref: http://splc.net/fame/philips-television.html

PHILIPS LíNEA DE PRODUCTOS – DISCUSIóN 3

PRODUCT LINE HALL OF FAME

Philips Product Line of Software for Television Sets

The second step was the creation of a *product line architecture*, profiting as much as possible from (re-engineered) existing software. This architecture is not a generic architecture shared by all products, but rather a set of rules and principles that should be obeyed by individual products, allowing each product to have its own specific and optimized architecture. The architecture was split in three main layers to anticipate on the use of 2nd and 3rd party software in the future.

To effectively deploy the architecture, changes had to be made to the existing *development processes*, which were optimized for the creation of single products. Typical changes were the introduction of component and interface data sheets to replace the traditional requirement specifications, and the definition of 'infinitely' progressing asset development projects, as opposed to finite product development projects.

Ref: http://splc.net/fame/philips-television.html

PHILIPS LíNEA DE PRODUCTOS – DISCUSIóN 4

PRODUCT LINE HALL OF FAME

Philips Product Line of Software for Television Sets

The fourth step was the adaptation of the *development organization* to accommodate product line development. While we do separate asset from product development, all asset teams are aware of all products and vice versa, to ensure that the right level of generality is achieved. Another important issue was the alignment of the organization with the architecture, making sure that each subsystem is being developed on a single site.

The component model was created in 1996. The architecture was defined in 1998. The first product came on the market in 2000. Since 2002, all Philips' mid-range and high-end televisions have software derived from this product line. Today, there are 20 different software releases per year, where each release serving 1-5 different product types. The product line supports three different hardware platforms. When we started, diversity was one of the top three issues on the agenda of architects. Now, diversity has disappeared as issue entirely.

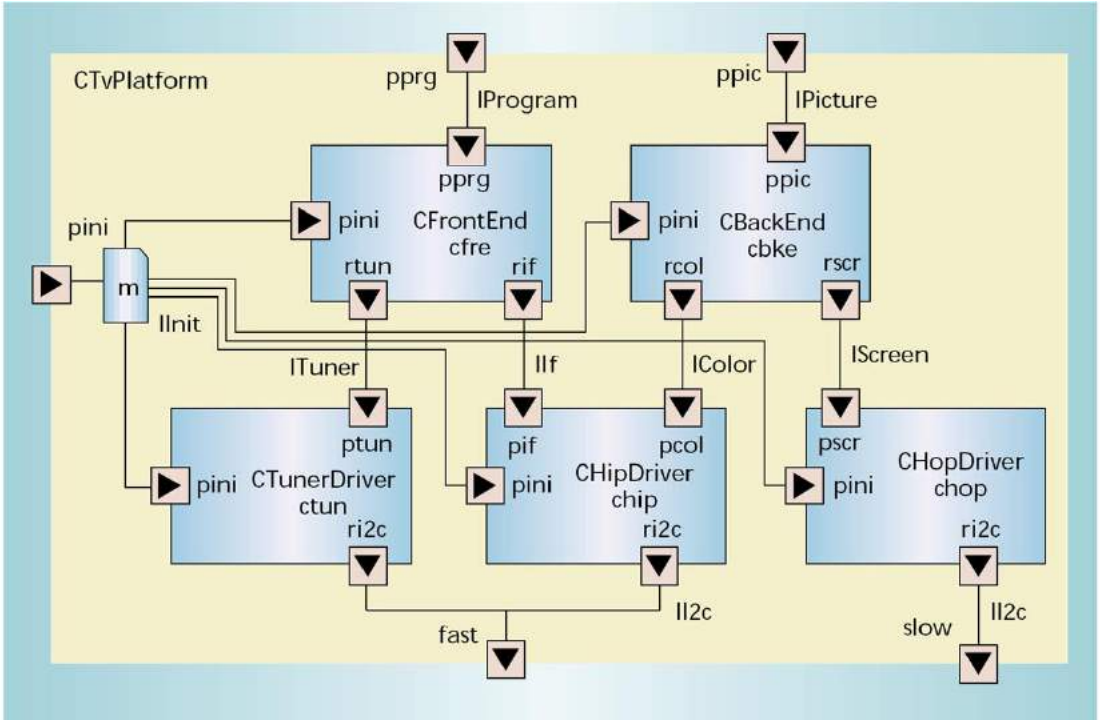
Ref: http://splc.net/fame/philips-television.html

PHILIPS LíNEA DE PRODUCTOS – IMPLEMENTACIóN

- Sistema usando Koala - colección de componentes interconectados, donde cada componente:
  - ✓ Exporta servicios a través de interfaces proveedoras
  - ✓ Define sus dependencias a través de interfaces consumidoras

- Los componentes pueden agregarse para conformar otro componente de mayor nivel.

- Permite construir y analizar una arquitectura con relativa facilidad



PHILIPS LíNEA DE PRODUCTOS – BENEFICIOS

- Reducción de costos
- Capacidad para rápidamente crear otra configuración para un TV
- Koala concentra la experiencia, conocimiento y ventaja competitiva de la compañía

PHILIPS LíNEA DE PRODUCTOS – CONCLUSIóN

Las arquitecturas de software proveen las abstracciones críticas que permiten que las variaciones y características comunes dentro de una familia de productos sean simultáneamente manejadas.

CONTENIDO

DEFINICIONES

La **organización general** del sistema.

Garlan & Shaw (94)

Partes que componen el software y sus **relaciones**.

Kruchten

La **organización fundamental de un sistema**, expresada a través de sus **componentes**, las **relaciones entre ellos y el ambiente**, y los principios que gobiernan su diseño y evolución.

IEEE, 1995

Conjunto de decisiones de diseño que se hacen en etapas tempranas del proyecto.

Anónimo

MAS DEFINICIONES

Una arquitectura es el conjunto de **decisiones importantes** sobre la **organización de un sistema de software**, la selección de los **elementos estructurales** y sus **interfaces** por los cuales el sistema está **compuesto**, junto con su **comportamiento** como es especificado en las colaboraciones entre estos elementos, la **composición** de estos elementos estructurales y de comportamiento en **subsistemas** progresivamente más grandes, y el estilo de arquitectura que guía a esta organización (estos elementos y sus interfaces, sus colaboraciones y su composición*).*

Booch, Rumbaugh y Jacobson en UML User Guide

(Addison-Wesley, 1999)

… MAS DEFINICIONES

Es el más alto nivel conceptual de un **sistema en su ambiente**. La arquitectura de un sistema de software (en algún punto en el tiempo) es su organización o **estructura de componentes importantes interactuando** a través de interfaces, y dichos componentes siendo **compuestos** de, sucesivamente, componentes e interfaces más pequeños.

RUP / IEEE

Ralph Jonhson

- ¿Qué es “el más alto nivel conceptual de un sistema”?
- Los clientes tienen distintos conceptos y visiones que los desarrolladores.
- Entonces la arquitectura sería el más alto nivel conceptual que los desarrolladores tienen del sistema y su ambiente
- ¿Qué hace que un componente sea importante?

OTRA DEFINICIÓN

En muchos proyectos de software exitosos, los desarrolladores expertos que trabajan en el proyecto tienen un entendimiento compartido del diseño del sistema. Este entendimiento compartido, llamado “arquitectura”, incluye cómo el sistema se divide en componentes, y cómo los componentes interactúan a través de interfaces. Estos componentes, generalmente, están compuestos de componentes más pequeños, pero la arquitectura solamente incluye a aquellos componentes e interfaces más relevantes para los desarrolladores.

Ralph Jonhson

Esta definición deja en claro que la **arquitectura es una construcción social**, porque no sólo depende del software sino también de:

- **Qué parte del software es considerada importante** por el grupo de consenso
- **Qué personas** toman las decisiones

Y OTRAS DEFINICIONES

La arquitectura trata de las **cosas importantes**. Sean lo que fueren esas cosas.

Ralph Jonhson

Toda la arquitectura es diseño, pero no todo el diseño es arquitectura. La arquitectura representa las decisiones de diseño que le dan forma a un sistema, donde la **importancia** es medida por el **costo del cambio**.

Grady Booch

ENUNCIADOS FALSOS

- La arquitectura y el diseño son la misma cosa
- La arquitectura y la infraestructura son la misma cosa
- *"Mi tecnología favorita"* es la arquitectura
- Una buena arquitectura es el trabajo de un único arquitecto
- La arquitectura es chata… un diagrama será suficiente
- La arquitectura es sólo la estructura
- La arquitectura del sistema precede a la arquitectura del software
- La arquitectura no puede ser medida y validada
- La arquitectura es una ciencia
- La arquitectura es un arte

CONTENIDO

IMPORTANCIA DE LA ARQUITECTURA – 1

1) Inhibe/habilita los atributos de calidad del sistema - si un sistema cumplirá con sus atributos de calidad es determinado (pero no garantizado) por la arquitectura.

Si se desea…	Poner atención a…
Performance	El tiempo que usa cada componente, el uso de recursos compartidos, la frecuencia y volumen de la comunicación entre componentes
Modificabilidad	Asignar responsabilidades a los componentes
Seguridad	Comunicación entre componentes, qué componentes pueden acceder a qué información, mecanismos de autenticación, etc.
Escalabilidad	Localizar el uso de recursos para facilitar su reemplazo por otros de mayor capacidad.
Reusabilidad	Restringir el acoplamiento entre componentes

IMPORTANCIA DE LA ARQUITECTURA – 2 y 3

2) Las decisiones tomadas en una arquitectura **permiten administrar los cambios a medida que el sistema evoluciona**

- Aproximadamente el 80% del costo total de un sistema de software ocurre luego de la primera instalación.
- Existen 3 tipos de cambios: Locales, No locales, Arquitecturales
- La arquitectura debe propiciar que la mayoría de los cambios sean locales.

3) El análisis de una arquitectura permite **predicciones tempranas** acerca de las **cualidades del sistema**

- Evaluando la arquitectura podríamos predecir si el sistema cumplirá con los atributos de calidad deseados.
- Herramientas: técnicas de evaluación cuantitativas, prototipos, pruebas de concepto, etc.

IMPORTANCIA DE LA ARQUITECTURA – 4

4) Una arquitectura documentada mejora la **comunicación entre los interesados**
   - La arquitectura provee una abstracción del sistema que los interesados pueden usar para entenderlo.
   - Cada interesado tiene interés en diferentes características del sistema
   - La arquitectura provee un lenguaje común en la cual los distinto intereses pueden ser expresados, negociados y resueltos a un nivel manejable para sistemas grandes y complejos.

Interesado	Interés
Usuario	Que el sistema sea rápido, confiable y esté disponible
Cliente	Que pueda ser implementado de acuerdo al plan y presupuesto
Gerente	Igual que el Cliente más que los equipos puedan trabajar independientemente, con interacciones controladas.
Arquitecto	Las estrategias para lograr todas esas metas

IMPORTANCIA DE LA ARQUITECTURA – 5

5) Es el portador de las decisiones de diseño más importantes y difíciles de cambiar
   - La arquitectura refleja las decisiones de diseño más tempranas que tienen gran influencia en el desarrollo, instalación y mantenimiento.
   - Cambiar alguna de estas decisiones causaría un efecto dominó sobre otras
   - Ejemplos de decisiones:
     - ✓ ¿El sistema ejecutará sólo en un procesador o en varios?
     - ✓ ¿El software estará diseñado por capas? ¿Cuántas? ¿Qué hará cada una?
     - ✓ ¿Los componentes se comunicarán sincrónica o asincrónicamente?
     - ✓ ¿El sistema dependerá de características especiales del sistema operativos y/o hardware?
     - ✓ ¿La información que fluye a través del sistema estará encriptada?
     - ✓ ¿Qué sistema operativo se usará?

IMPORTANCIA DE LA ARQUITECTURA – 6 a 8

6) Define un conjunto de **restricciones sobre la implementación** subsecuente
   - La implementación de un sistema debe conformar las decisiones de diseño prescriptas por la arquitectura
   - Quienes construyen los componentes, deben se idóneos para hacerlo, pero no deben preocuparse por los trade-offs de arquitectura (ya resueltos por la arquitectura)

7) Influye en la estructura organizacional
   - La arquitectura define la descomposición de más alto nivel del sistema
   - En grandes proyectos, muchas veces se asignan diferentes porciones del sistema a diferentes grupos.

8) Provee las bases para un prototipo evolutivo
   - Permite hacer un prototipo del esqueleto del sistema
   - Reduce riesgos potenciales del proyecto.

IMPORTANCIA DE LA ARQUITECTURA – 9 y 10

9) Permite al arquitecto y gerente del proyecto razonar acerca del costo y la planificación del sistema
   - La descomposición de alto nivel del sistema permite, además de crear equipos especializados en las distintas partes, tener un mejor conocimiento de esas partes y hacer estimaciones más precisas de los componentes y, luego, del sistema completo.

10) Puede ser creada como un modelo reusable y transferible que sea el corazón de una línea de productos
   - No solamente el código puede ser reusado entre sistemas, la arquitectura también.
   - Esto es fundamental para familias de productos que deberían utilizar una arquitectura que indique qué es fijo para todos los miembros de la familia y qué es variable.

IMPORTANCIA DE LA ARQUITECTURA – 11

11) El desarrollo basado en una arquitectura **pone atención en el ensamblaje de los componentes**, y no solamente en su creación
    - El desarrollo basado en arquitectura generalmente se enfoca en componer/ensamblar partes que fueron desarrolladas separadamente:
        - ✓ Componentes comerciales de terceros
        - ✓ Software de código abierto
        - ✓ Aplicaciones disponibles en espacios públicos
        - ✓ Servicios on-line.
    - La arquitectura restringe los componentes posibles de acuerdo a varios criterios.
    - Beneficios
        - ✓ Disminución del “time to market”
        - ✓ Incremento de la confiabilidad del sistema
        - ✓ Menor costo
        - ✓ Flexibilidad
        - ✓ ¿Qué sistema operativo se usará?

IMPORTANCIA DE LA ARQUITECTURA – 12 y 13

12) Restringiendo las alternativas de diseño, la arquitectura canaliza la creatividad de los desarrolladores, reduciendo el diseño y la complejidad del sistema
    - La arquitectura muestra el camino por donde la solución detallada será encontrada, restringiendo las posibilidades de elección a sólo algunas que se sabe serán apropiadas.
    - Beneficios: Mayor reuso, diseños más regulares y simples, tiempos de selección menores, mayor interoperabilidad.

13) Puede ser la base para la capacitación de un nuevo integrante del equipo
    - La arquitectura, incluyendo una descripción de cómo los componentes interactúan para resolver el comportamiento requerido, pueden servir como una introducción para un nuevo integrante de equipo
    - La arquitectura ayuda a la comunicación entre los distintos interesados

CONTENIDO

PARA DISCUTIR …

¿Qué pasaría si a dos arquitectos diferentes, trabajando en dos organizaciones diferentes les dieran la misma especificación de requerimientos para un sistema? ¿Producirían la misma arquitectura o diferentes?

INFLUENCIAS DEL ENTORNO

- Un arquitecto diseñando un sistema de tiempo real tomará decisiones diferentes que si diseña un sistema que no sea de tiempo real

- Bajo los mismos requerimientos, hardware, software de soporte y recursos humanos disponible, un arquitecto diseñando un sistema hoy lo haría diferente a cómo lo hubiese hecho hace 5 años

FACTORES QUE INFLUENCIAN LA ARQUITECTURA

- Interesados
- Organización
- Experiencia de los arquitectos

FACTOR 1 – INTERESADOS

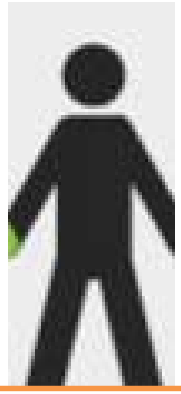
INTERESADOS

Persona con interés en el éxito del sistema: cliente, usuario final, desarrollador, gerente de proyecto, equipo de mantenimiento, persona de marketing.

Desarrollo Organizacional

Costos bajos, Mantener empleo

Marketing



Tiempos cortos para el mercado, costos bajos, competencia

Usuario Final

Comportamiento, performance, seguridad, confiabilidad, etc.

Cliente

Costos bajos, puntualidad, necesidad de pocos cambios

ARQUITECTO(A)

INTERESADOS Y SUS INTERESES – 1

INTERESADO	INTERES
Analista	Analizar la satisfacción de los atributos de calidad basado en la arquitectura
Arquitecto	Negociar y balancear requerimientos que compiten entre sí y enfoques de diseño.
Gerente de negocios	Comprender la habilidad de la arquitectura para satisfacer los objetivos del negocio
Controlador de calidad	Asegurar que las implementaciones respetan las prescripciones de la arquitectura
Cliente	Asegurar la funcionalidad y calidad requerida, medir progreso; estimar costos; definir expectativas sobre la entrega y el costo
Administrador de base de datos	Comprender como se crean, usan y actualizan los datos por los elementos de la arquitectura, y que propiedades deben tener la base de datos y los datos para satisfacer la calidad

INTERESADOS Y SUS INTERESES – 2

INTERESADO	INTERES
Persona que entrega	Comprender los elementos de la arquitectura que se entregan y donde se deben desplegar en los sitios del usuario final
Diseñador	Resolver conflictos de recursos y establecer performance y otros presupuestos de consumo de recursos en tiempo de ejecución.
Evaluador	Evaluar la habilidad de la arquitectura para entregar el comportamiento y los atributos de calidad requeridos
Implementador	Comprender las restricciones y los permisos para las actividades de desarrollo
Integrador	Producir planes y procedimientos de integración y ubicar fuentes de fracasos de integración
Persona de mantenimiento	Comprender las ramificaciones de un cambio

INTERESADOS Y SUS INTERESES – 3

INTERESADO	INTERES
Administrador de red	Determinar la carga de la red durante distintos escenarios de uso y comprender el uso de la red
Gerente de línea de producto	Determinar si un nuevo miembro potencial de la familia de producto esta fuera del alcance; y si lo está, por cuanto
Gerente de proyecto	Ayudar a definir presupuesto y cronograma, controlar progreso con lo planificado, identificar y resolver conflictos por recursos
Representante de otro sistema	Definir el conjunto de acuerdos entre sistemas

INTERESADOS Y SUS INTERESES – 3

INTERESADO	INTERES
Ingeniero de sistemas	Asegurar que el entorno del sistema provisto por el software es suficiente
Testeador	Crear casos de prueba basados en el comportamiento y la interacción de los elementos de software
Usuario	En el rol de revisores, podría usar la documentación de la arquitectura para: a) controlar que se entrega la funcionalidad prometida; b) entender los principales elementos del sistema.

FACTOR 2 - ORGANIZACIÓN

Una organización frecuentemente tiene inversiones en artefactos (tales como arquitecturas y productos basados en dicha arquitectura).

Por lo tanto, se esperaría desde la organización:

- Que el nuevo sistema sea el próximo en la secuencia de sistemas similares
- Alto grado de reuso de artefactos
- Alto grado de productividad de los desarrolladores

Una organización puede desear hacer una inversión de largo plazo, persiguiendo metas estratégicas.

- Por ejemplo: una organización decide que quiere desarrollar reputación sobre soluciones basadas en cloud computing. Esto afectará a la arquitectura y a la infraestructura de un nuevo proyecto: contratar gente con experiencia en cloud computing, planificar capacitaciones, …

FACTOR 2 – ORGANIZACIÓN …

La estructura organizacional puede darle forma a la arquitectura del software o viceversa

- Las organizaciones se organizan alrededor de tecnologías y conceptos de aplicación: equipo de UI, equipo de base de datos, equipo de analistas funcionales, etc.
- Un nuevo subsistema/tecnología podría conducir a la creación de un nuevo grupo

FACTOR 3 – EXPERIENCIA DE ARQUITECTOS

Experiencias previas

- Buena experiencia en arquitectura previa → Grandes chances de repetirla
- Mala experiencia en arquitectura previa → No volverán a usarla

Conocimientos

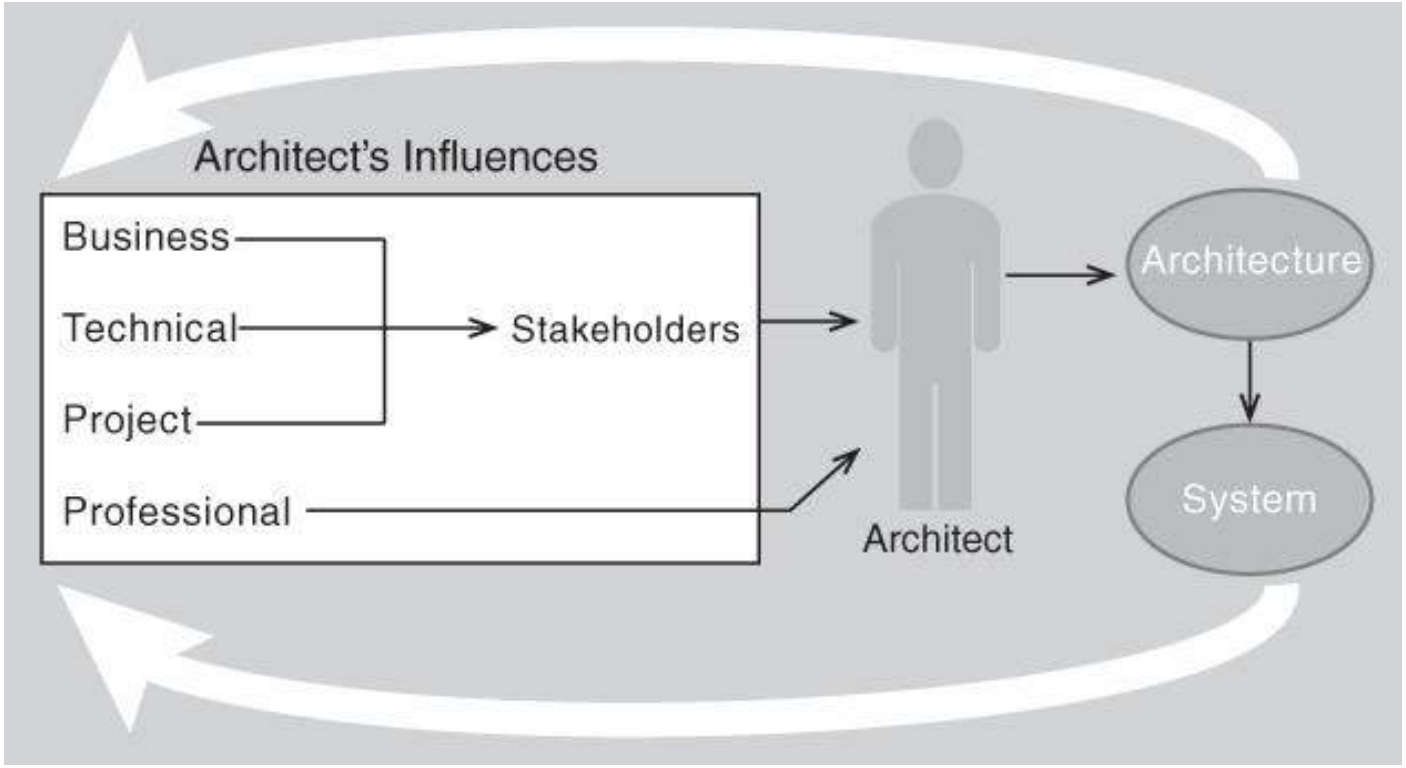
- Educación y entrenamiento
- Exposición a patrones arquitecturales exitosos
- Exposición a sistemas que funcionaron bien/mal
- Experimentar con patrones arquitecturales o técnicas aprendidas en un libro o curso.

Las arquitecturas son influenciadas por el ambiente técnico

- Prácticas estándares de la industria
- Técnicas predominantes en la comunidad de profesionales del arquitecto

CICLO DE INFLUENCIA DE LA ARQUITECTURA

La arquitectura también influencia al contexto técnico, de proyecto, de negocio y profesional que terminarán influenciando futuras arquitecturas



RESUMEN - 1

- Analogía entre la construcción de software y la construcción de edificios
- La arquitectura de software debe ser el núcleo del diseño y desarrollo de un sistema de software
- Importancia de la arquitectura de software en dos casos de estudio – la web, y una línea de productos de software (SPL)
- Definiciones de arquitectura

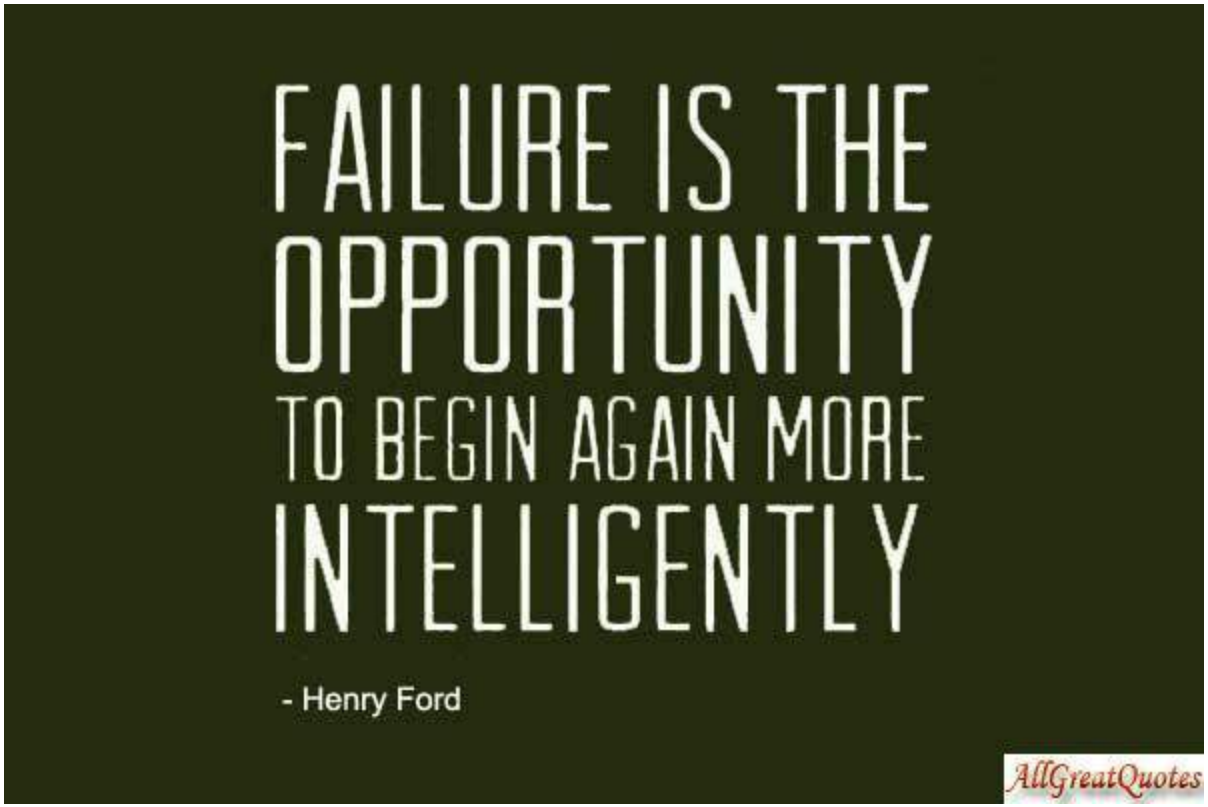
RESUMEN - 2

- Importancia de la arquitectura
  - ✓ Evaluación de atributos de calidad
  - ✓ Administración de cambios
  - ✓ Predicción de manera temprana la calidad
  - ✓ Comunicación con interesados
  - ✓ Contiene decisiones importantes de diseño
  - ✓ Define restricciones de implementación
  - ✓ Influye la estructura organizacional
  - ✓ Provee bases para prototipos evolutivos
  - ✓ Permite razonar sobre costos y planificación
  - ✓ Puede ser creada con modelos reusables y transferibles (SPL)
  - ✓ Atención a ensambles de componentes
  - ✓ Reducción de complejidad
  - ✓ Base para la capacitación de personal nuevo

RESUMEN - 3

- Factores de influencia
  - ✓ Interesados
  - ✓ Organización
  - ✓ Experiencia de los arquitectos

PARA PENSAR


FAILURE IS THE
OPPORTUNITY
TO BEGIN AGAIN MORE
INTELLIGENTLY
- Henry Ford
AllGreatQuotes

Elsa Estevez
ece@cs.uns.edu.ar