

Arquitectura de Software

Conceptos generales

Qué es arquitectura de software?

- Ideas erróneas (Kruchten y Booch):
 - Arquitectura es un simple papel.
 - Arquitectura y diseño son lo mismo.
 - Arquitectura e infraestructura son lo mismo.
 - *<mi tecnología favorita>* es la arquitectura.
 - Una buena arquitectura es el trabajo de un único arquitecto.
 - Arquitectura es simplemente estructura.
 - Arquitectura es algo chato que puede ser representado unidimensionalmente.
 - Arquitectura no puede ser medida o validada.
 - Arquitectura es una ciencia.
 - Arquitectura es un arte.

Qué es arquitectura de software?

Ideas erróneas

- Arquitectura es un simple papel
 - La arquitectura de un sistema reside finalmente en el código que ejecuta.
 - La arquitectura puede ser visualizada utilizando modelos.
- Arquitectura es diseño
 - La arquitectura es diseño, pero no todo el diseño es arquitectura.
 - La arquitectura se enfoca en decisiones de diseño significativas, decisiones importantes desde el punto de vista estructural y de comportamiento así como aquellas que tienen un impacto permanente en la eficiencia, costo, modificabilidad, etc.

Qué es arquitectura de software?

Ideas erróneas

- *<mi tecnología favorita>* es la arquitectura.
 - La tecnología solo sirve para implementar una dimensión de la arquitectura.
 - La tecnología solo le da forma a la arquitectura.
- Arquitectura es estructura
 - Arquitectura también implica comportamiento.
 - Arquitectura siempre se proyecta en un determinado contexto.

Qué es arquitectura de software?

Ideas erróneas

- La arquitectura puede ser representada unidimensionalmente
 - Solo en sistemas triviales.
 - Múltiples actores con múltiples necesidades sobre un sistema necesitarán visiones diferentes del mismo y representaciones particulares en base a dichas necesidades.
 - El uso de un único modelo para representar un sistema resulta semánticamente confuso.

Qué es arquitectura de software?

Ideas erróneas

- La arquitectura es una ciencia
 - Carecemos de métodos científicos o analíticos para crear arquitecturas de software que puedan ser aplicados fácilmente.
 - No existe un diseño perfecto. La arquitectura implica manejar extrema ambigüedad y contradicciones.
 - La experiencia del arquitecto es importante.
- La arquitectura es un arte
 - Incluso los mejores arquitectos copian soluciones probadas, las mejoran y adaptan a su contexto.
 - La parte “artística” de la arquitectura de software es mínima.
 - Un proceso bien definido de diseño arquitectónico puede ser establecido.

Qué es arquitectura de software?

- Definiciones:
 - La organización fundamental de un sistema representado por sus componentes, las relaciones entre ellos y con su contexto, y los principios que guían su diseño y evolución (IEEE 2000a, p. 9).
 - La arquitectura de software abarca el conjunto de decisiones significativas sobre la organización de un sistema de software incluyendo:
 - Elección de elementos estructurales y sus interfaces, los cuales componen al sistema.
 - Comportamiento tal como se especifica en las colaboraciones entre dichos elementos (*continua*)

Qué es arquitectura de software?

- Definiciones:
 - *(continua)*
 - Composición de estos elementos estructurales y su comportamiento en sub-sistemas mas grandes.
 - Estilo arquitectónico que guía esta organización.
 - La arquitectura de software también implica funcionalidad, usabilidad, flexibilidad, eficiencia, reutilización, comprensibilidad, limitantes económicos y tecnológicos, *trade-offs* y cuestiones estéticas (Booch, Kruchten, Bittner, Reitman and Shaw).

Qué es arquitectura de software?

- Definiciones:

- La arquitectura de software de un programa o sistema de computación es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos (Bass, Clements, Kazman 2003, p. 21)
- La estructura de los componentes de un programa/sistema, sus interrelaciones, y principios y guías que gobiernan su diseño y evolución en el tiempo (Garlan and Perry 1995, p. 269).

Por qué la arquitectura es importante?

Existen 3 razones fundamentales que determinan la importancia de la arquitectura de software:

- Comunicación entre los accionistas.
- Decisiones tempranas de diseño.
- Representación transferible de un sistema.

Por qué la arquitectura es importante?

Arquitectura es el vehículo para la comunicación entre los accionistas.

- Accionistas: Clientes, Usuarios, Gerente de proyecto, Desarrollador, Tester, etc.
- Cada uno de los accionistas están interesados en características diferentes del sistema que son afectadas por la arquitectura.
- La arquitectura provee un lenguaje común en el cual cada asunto o problema puede ser expresado, negociado y resuelto.

Por qué la arquitectura es importante?

Arquitectura manifiesta las decisiones prematuras de diseño.

- Define restricciones sobre la implementación.
- Inhibe o posibilita un atributo de calidad del sistema.
- Permite predecir la calidad del sistema.
- Facilita el razonamiento y gestión del cambio.
- Permite estimaciones de costo y presupuesto mas precisos

Por qué la arquitectura es importante?

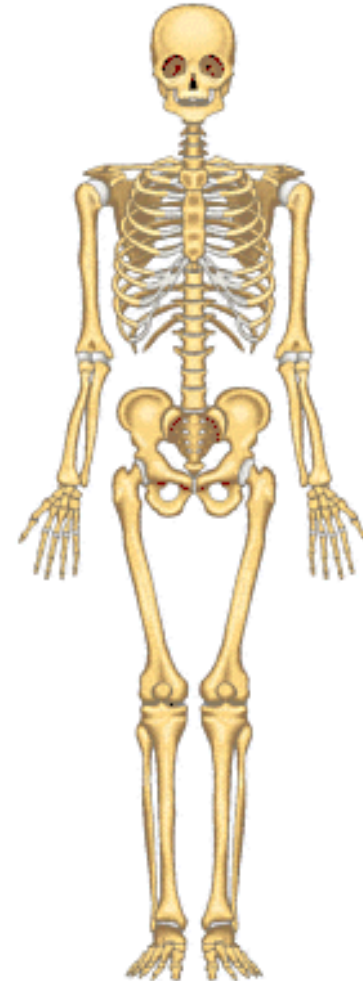
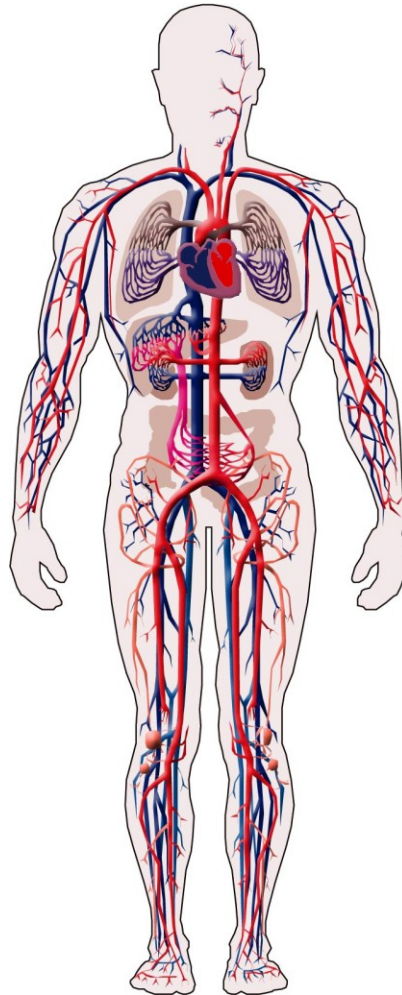
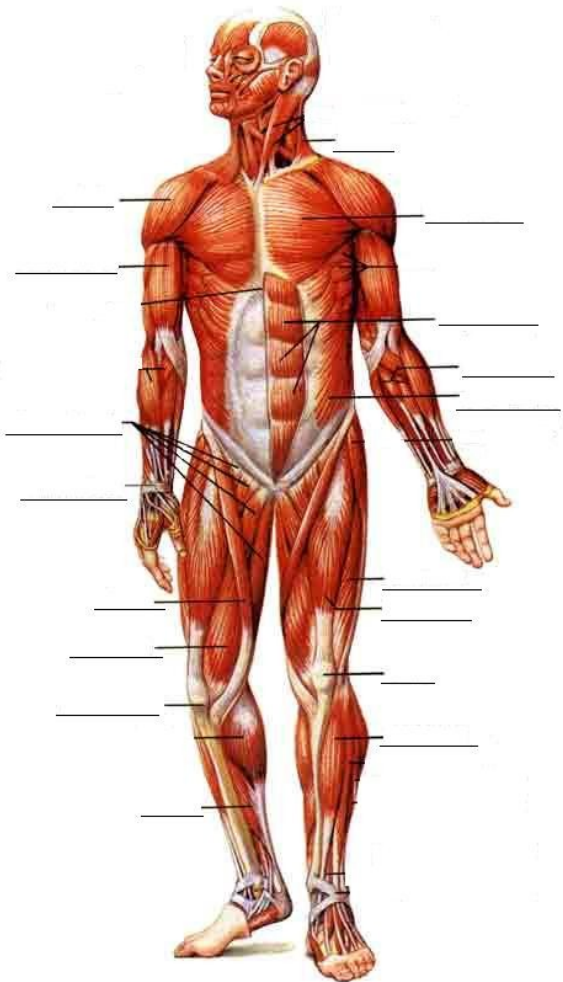
Arquitectura como un modelo transferible y reutilizable.

- Líneas de productos comparten una misma arquitectura base.
- Facilita la construcción de software a partir de grandes elementos desarrollados externamente.
- Permite el desarrollo basado en “templates”.
- Puede ser la base para el entrenamiento de nuevos miembros de un proyecto.

Estructuras y vistas arquitectónicas.

- Los sistemas de software modernos son complejos, de modo que no podemos entenderlos a simple vista, ni podemos describirlos de forma unidimensional.
- Para entender un sistema debemos estudiar cada **estructura** del sistema de forma aislada.
- Una **estructura** es un conjunto de elementos, tal como existen en el software o hardware. Ej. Modulo o Componente.
- Una **vista** es la **representación** de un conjunto coherente de elementos arquitectónicos y sus relaciones. Una vista es una representación de una estructura.

Estructuras y vistas arquitectónicas.



Estructuras y vistas arquitectónicas.

- Elementos
 - Músculos
 - Arterias
 - Venas
 - Huesos
- Estructuras
 - Sistema muscular
 - Sistema circulatorio
 - Esqueleto
- Vistas
 - Diferentes representaciones

Estructuras y vistas arquitectónicas.

Las estructuras arquitectónicas pueden ser divididas en 3 grupos fundamentales, dependiendo de la naturaleza de sus elementos:

- Estructuras de **Módulos**
 - Son unidades de implementación.
 - Se les asigna responsabilidades funcionales.
 - No interesa como se manifiestan en tiempo de ejecución.
 - Que funcionalidad se asigna a que módulo?
 - Que otros elementos de software pueden ser utilizados por un módulo?
 - Que elementos de software son realmente utilizados por un módulo?
 - Que módulos se relacionan a partir de una generalización?

Estructuras y vistas arquitectónicas.

- Estructuras de **Componentes y conectores**
 - Aquí los elementos son **Componentes** (unidades principales de computo) y **Conectores** (medios de comunicación entre los componentes).
 - Los componentes y conectores se manifiestan en tiempo de ejecución.
 - Cuales son los principales componentes ejecutables y como interactúan?
 - Cuales son los principales repositorios de datos?
 - Que partes del sistema están replicadas?
 - Como fluyen los datos en el sistema?
 - Que partes del sistema se ejecutan en paralelo?

Estructuras y vistas arquitectónicas.

- Estructuras de **Asignación**

- Las estructuras de **asignación** muestran la relación entre los elementos de software y los elementos del ambiente o entorno en el cual el software es creado o ejecutado.
- En que procesador o servidor ejecuta que elemento de software?
- Como se organizan los elementos de software en el *file system*? En que archivos de código o proyectos residen los elementos de software durante el desarrollo o el testing?
- Como se asignan los elementos de software a equipos de desarrollo?

Referencias

- Software Architecture in Practice (2nd Edition)
 - Len Bass, Paul Clements, Rich Kazman
- Misconceptions about architecture (Booch)
 - *www.booch.com/architecture/blog/artifacts/Software%20Architecture.ppt*