

¿ qué es la A.S.?

Existen muchas definiciones (más adelante), pero empezaremos definiendo : A.S. es un conjunto de estructuras necesarias para razonar acerca de un sistema, el cual comprende elementos de software, relaciones entre ellos y propiedades.

AS → conjunto de estructuras de Software

- ¿ qué es una estructura ? → conjunto de elementos unidos bajo una relación.
- 3 categorías de estructuras arquitectónicas:

1) Algunas estructuras particionan los sistemas en unidades de implementación → **module**. Cada módulo tiene una responsabilidad computacional.

2) Otras estructuras son dinámicas (run-time) y se conocen como **C&C**.

3) Otras estructuras describen el “mapeo” de las estructuras de sw con la organización del sistema o los ambientes de dev,qa,prod, etc. Se conocen como **allocation**

AS → abstracción

- La arquitectura OMITE cierta información sobre los elementos que no son útiles para razonar acerca del sistema.
- La arquitectura es una abstracción del sistema que selecciona ciertos detalles y suprime otros.



AS → todos los sistemas tienen una

- Aunque cada sistema tiene una arquitectura, es probable que no todos la **conozcan ... (motivo ?)**
- Esto genera el problema de la arquitectura versus la **representación** de la arquitectura → la arquitectura existe independientemente de su descripción ... por eso es importante DOCUMENTAR !!



AS → incluye comportamiento

- El comportamiento permite razonar acerca del sistema porque indica como interactúan los elementos
- Notar que esto no describe el comportamiento exacto o la performance de cada elemento



AS → no todas son buenas

- La arquitectura existe independientemente de si es buena o no.

Necesidad de una Arquitectura de Software

- o La ausencia de documentación de la arquitectura de software causa problemas.
- o Una buena arquitectura tiene beneficios
- o Cada proyecto de software necesita su propio equipo de arquitectura

Unas definiciones

- La arquitectura de software de un sistema es un conjunto de decisiones de diseño importantes o principales sobre el sistema.
- La arquitectura de software es una plantilla para la construcción y evolución del software.
- Son decisiones de diseño que acompañan cada faceta del sistema bajo desarrollo:
 - Estructura
 - Comportamiento
 - Interacción
 - Requerimientos no funcionales

Otra definición

"La arquitectura de software de un sistema son el conjunto de estructuras que nos permiten razonar sobre el sistema. Una estructura involucra elementos de software, las propiedades de estos elementos y las relaciones entre ellos"¹.

¹ Bass, L., Clements, P. & Kazman, R. Software Architecture in Practice, Third Edition. 2013. Addison-Wesley.



Más definiciones

- Perry y Wolf

- Software Architecture = { Elements, Form, Rationale }
what how why

- <http://users.ece.utexas.edu/~perry/work/papers/swa-sen.pdf>

- Shaw y Garlan

- Software architecture [is a level of design that] involves
 - the description of elements from which systems are built,
 - interactions among those elements,
 - patterns that guide their composition, and
 - constraints on these patterns.

- Kruchten

- Software architecture deals with the design and implementation of the high-level structure of software.
 - Architecture deals with abstraction, decomposition, composition, style, and *aesthetics*.

Una aproximación más...



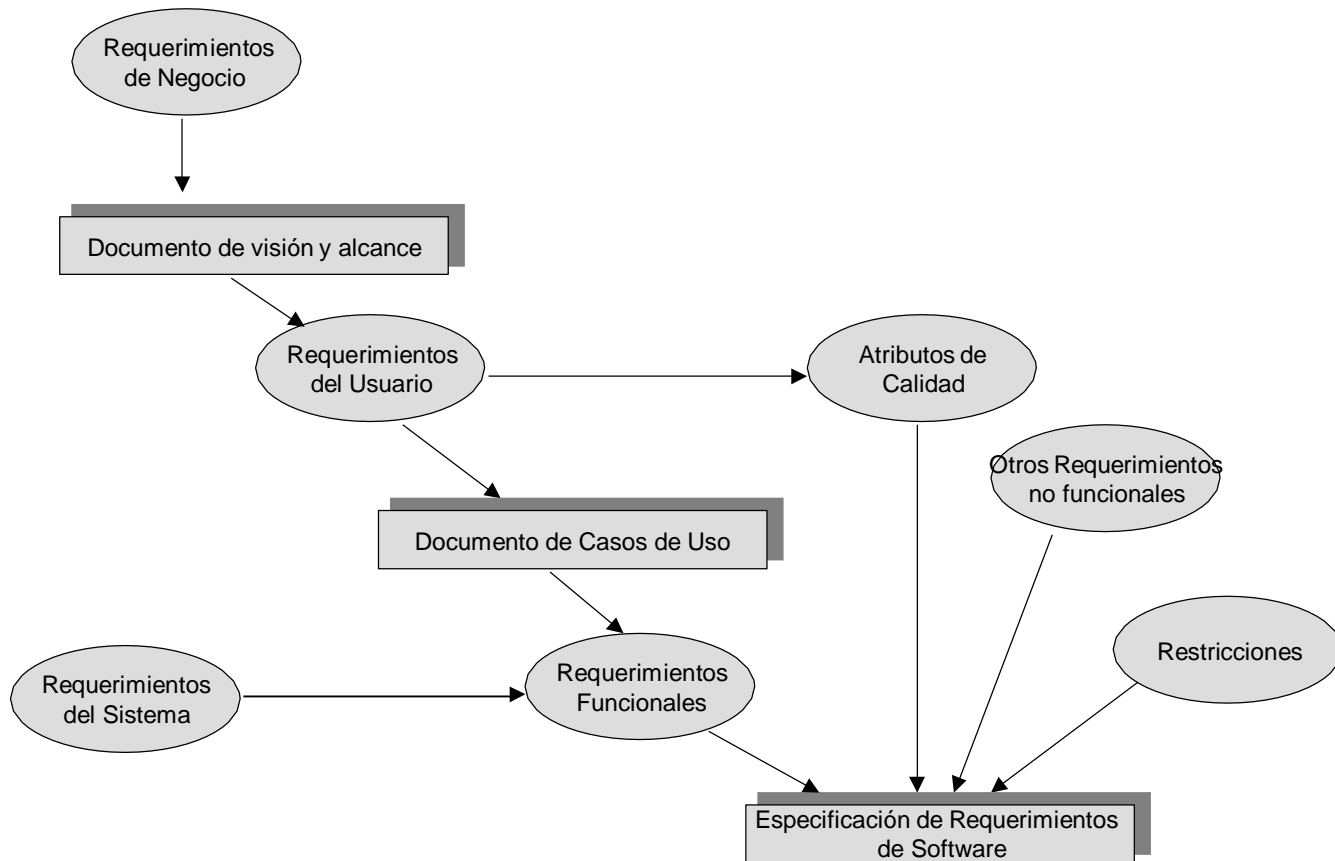
Estructura



Entender lo que se
necesita construir

Todo se basa en los requerimientos

Relación entre los requerimientos de Software



Fuente: WIEGERS, Karl. "Software Requirements". Microsoft Press, 1997. ISBN 0-7356-0631-5.

Tipos de arquitectura

Existen muchos tipos de arquitectura:

Infraestructura, network,
de soluciones, técnica, de
aplicaciones, empresarial,
etc.

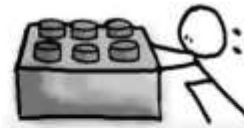


¿Qué tienen en común ?

ESTRUCTURA



VISIÓN



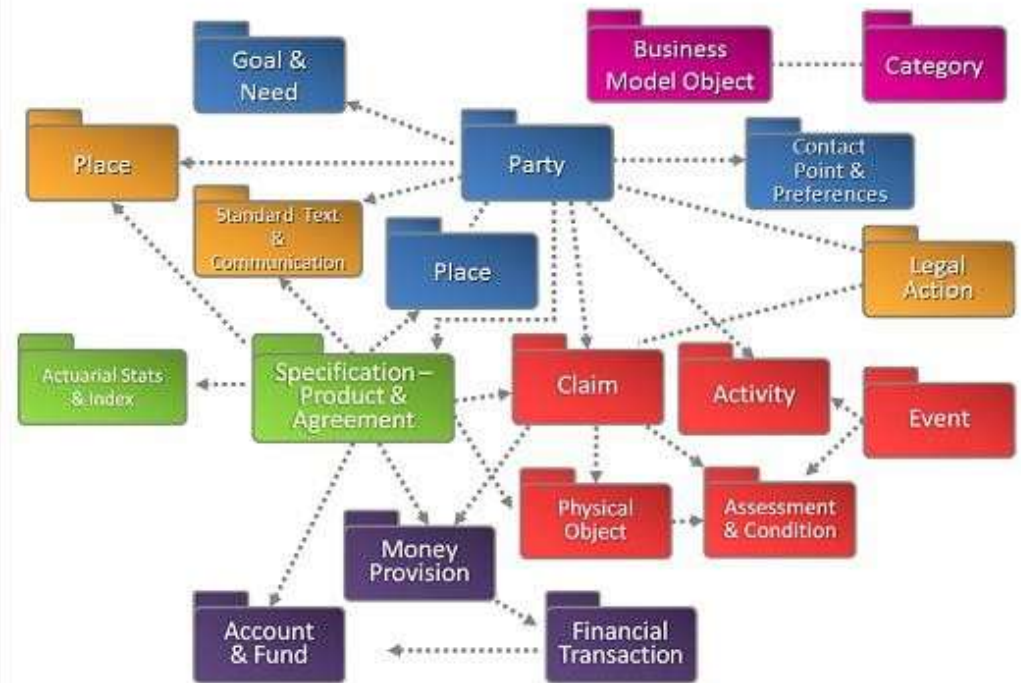
Entender el problema y crear una visión que pueda ser comunicada a todos los involucrados en la construcción del producto final

Arquitectura de Aplicaciones

- Bajo nivel
- Generalmente vinculado a un conjunto de tecnologías (java,net, etc.)
- Comprende:
 - ✓ Lenguajes de programación
 - ✓ Librerías
 - ✓ Frameworks
 - ✓ API's



Software y organización del código

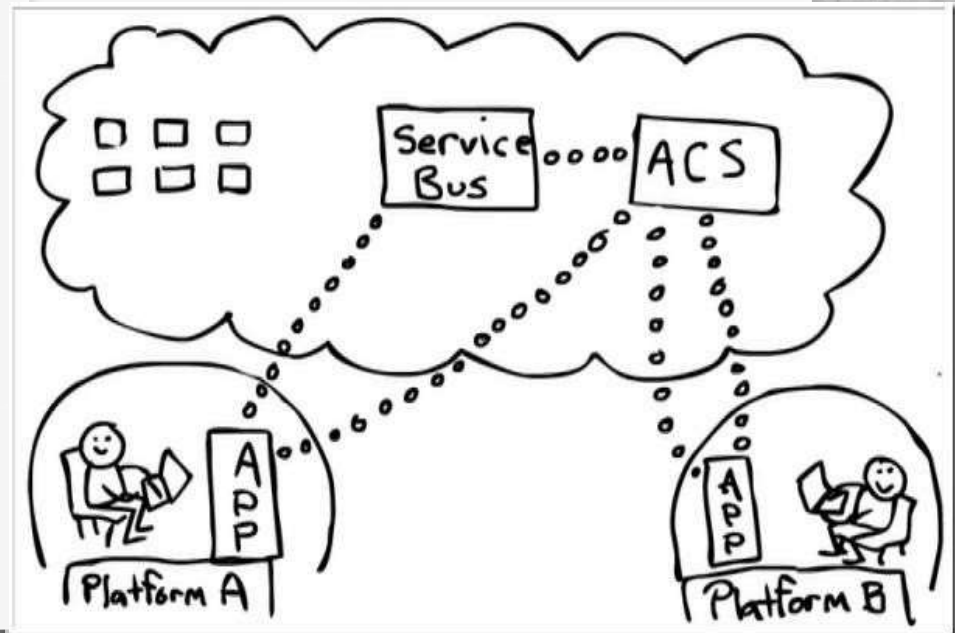


Arquitectura de Sistemas

- Estructura completa de una aplicación a alto nivel
- Comprende:
 - ✓ Mix de HW + SW + Humanos
 - ✓ Alto nivel de abstracción
 - ✓ Componentes y Servicios a traves de subsistemas
- Negociación entre Arq de Sistemas y Arq de SW para ver la distribución de funcionalidad → restricciones a la arquitectura



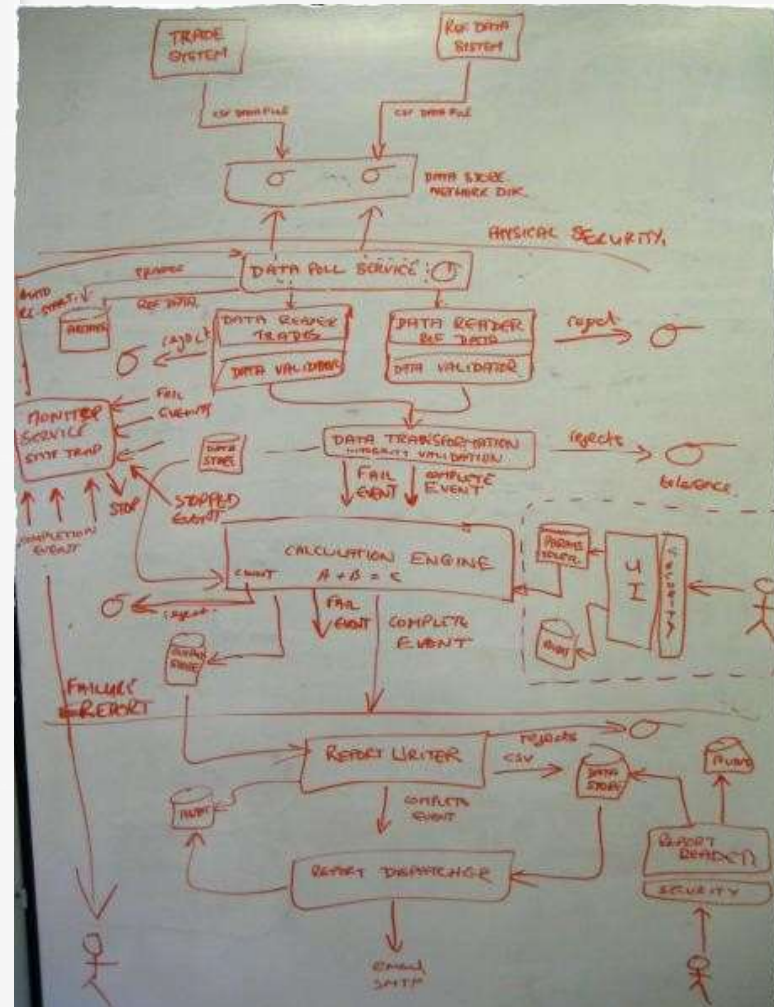
Relacionado a la interoperación e integración con otros sistemas



Arquitectura de Software

- Muchos significados: depende de las personas
- En general va desde la concepción hasta el despliegue del código.
- Si todos son "developers" ¿quien se ocupa de los RNF's ?

Combinación de arquitectura de Aplicaciones con Arquitectura de Sistemas

































Arquitectura Empresarial



Orientada a la Estrategia

- Analiza la forma en que se organizan la gente, los procesos y la tecnología para que le empresa logre sus objetivos.
- No necesariamente ve la tecnología en detalle.
- Se prioriza la estrategia antes que el código : QUE sistemas, con QUE funcionalidad deberían soportar a los objetivos de negocio

ENTERPRISE ARCHITECTURE - A FRAMEWORK™

	DATA	Proc	FUNCTION	Net	NETWORK	ITime	PEOPLE	TIME	MOTIVATION	
SCOPE (CONTEXTUAL)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events/Cycles Significant to the Business 	List of Business Goals/Strategies 	SCOPE (CONTEXTUAL)			
Power	Entity = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization	Time = Major Business Event/Cycle	End/Mean = Major Business Goal/Strategy	Power			
BUSINESS MODEL (CONCEPTUAL)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Business Location System 	e.g. Work Flow Model 	e.g. Master Schedule 	e.g. Business Plan 	BUSINESS MODEL (CONCEPTUAL)			
Owner	Ent = Business Entity Rel = Business Relationship	Proc = Business Process FID = Business Relationship	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	Owner			
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model 	e.g. Application Architecture 	e.g. Distributed System Architecture 	e.g. Human Interface Architecture 	e.g. Processing Structure 	e.g. Business Rule Model 	SYSTEM MODEL (LOGICAL)			
Designer	Ent = Data Entity Rel = Data Relationship	Proc = Application/Function FID = User Usage	Node = IS Function (Processor, Storage, etc.) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = System Goal/Strategy Means = Rule Assertion	Designer			
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model 	e.g. System Design 	e.g. Technology Architecture 	e.g. Presentation Architecture 	e.g. Control Structure 	e.g. Rule Design 	TECHNOLOGY MODEL (PHYSICAL)			
Builder	Ent = Segment/Table/View Rel = Data Relationship	Proc = Computer Function FID = Data Relationship	Node = Hardware/Software Link = Line Characteristics	People = User Work = Screen Format	Time = Global Cycle Cycle = Consistent Cycle	End = Condition Means = Action	Builder			
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)			
Sub-Component	Ent = Field Rel = Attribute	Proc = Language Statement FID = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interval Cycle = Machine Cycle	End = Sub-condition Means = Test	Sub-Component			
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE			

© 1986 - 2005 John A. Zachman, Zachman International

See www.ZachmanInternational.com for 2005 Zachman Framework Standards

Arquitectura vs. Diseño

"Toda arquitectura es diseño, pero no todo diseño es Arquitectura"

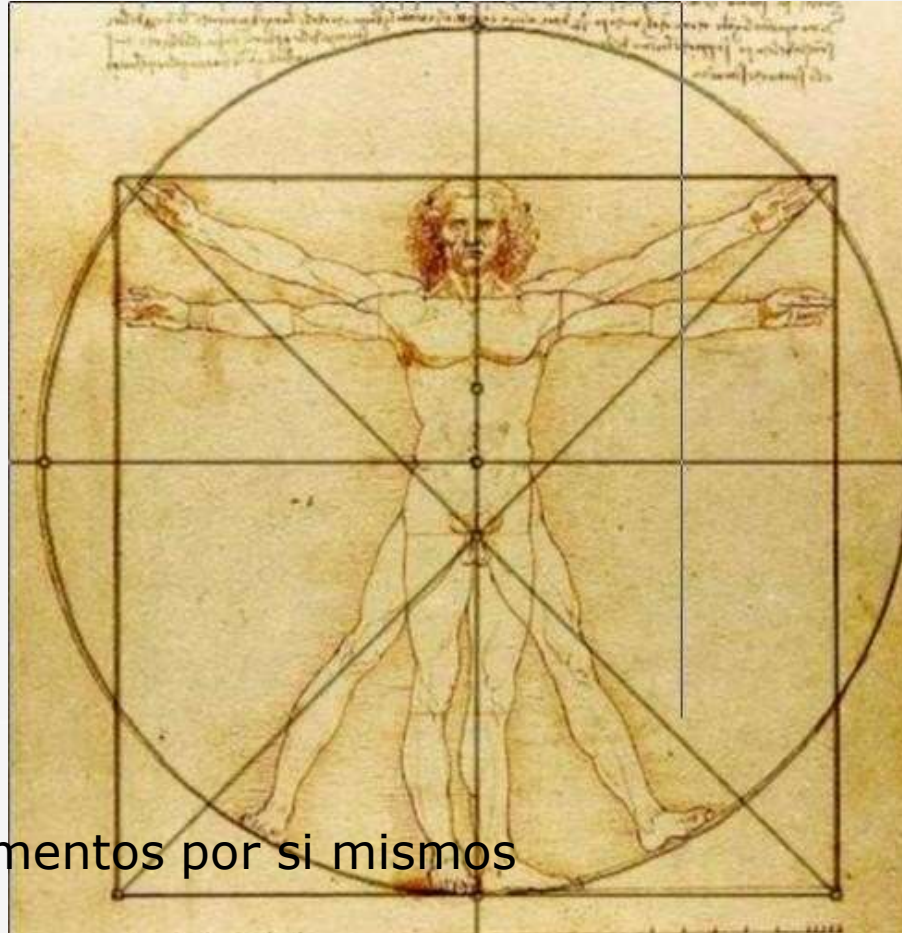
Grady Booch



*La arquitectura representa decisiones **importantes** de diseño que dan forma a un sistema y en donde la importancia se mide en función a los **costos** que representa cada cambio.*



Estructura



conjunto elementos por si mismos

Vistas



Representación de un conjunto coherente de elementos arquitecturales: elementos y relaciones

Vistas y Estructuras Arquitectónicas



A.S. : diseñar estructuras y documentar vistas de esas estructuras

¿ Cuales eran las
estructuras
arquitectónicas ?



Algunas estructuras útiles

Modules

- ✓ Decomposition → "is-a-submodule-of"
- ✓ Uses → es una forma de dependencia
- ✓ Layer → es una VM abstracta
- ✓ Class → es una forma de generalización :
hereda de o instanceof
- ✓ Data Model → entidades y relaciones

C&C

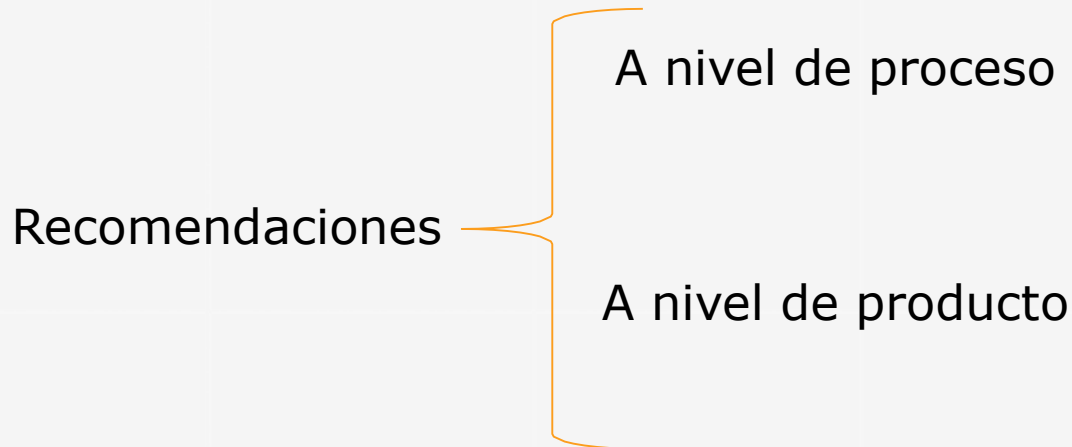
- ✓ Service → servicios que interoperar mediante un mecanismo de coordinación (SOAP/REST/RPC/RMI/IIOP/CORBA)
- ✓ Concurrency → oportunidad para determinar paralelismo y bottlenecks

Allocation

- ✓ Deployment → relación de "allocated-to" o "migrates-to"
- ✓ Implementation → normalmente módulos que mapean a estructuras de archivos en el sistema
- ✓ Work assignment → quien hace qué

¿ Buena o Mala Arquitectura ?

La A.S. debe ser evaluada pero dentro del contexto de objetivos específicos del sistema/negocio



Importancia de una AS

1. La comunicación entre los interesados.
2. Las decisiones de diseño temprano.
3. Permite estimar el costo y esfuerzo
4. Permite la creatividad ... reduciendo la complejidad

¿ más razones ?

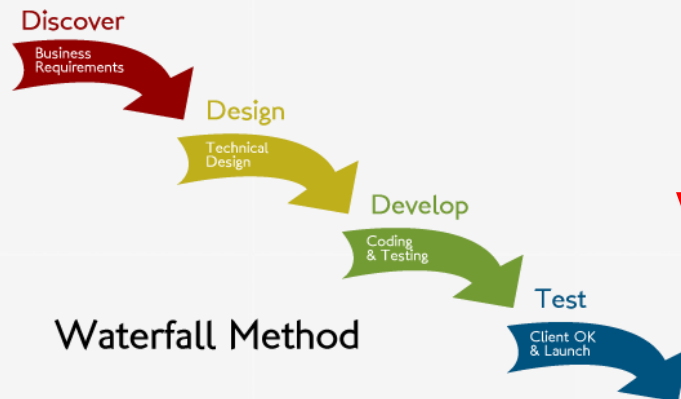
Contextos de una AS

- ☐ Técnico : inhibe o estimula el cumplimiento de los atributos de calidad
 - La funcionalidad NO ES parte de la arquitectura
 - La Arquitectura es el contenedor donde se coloca la funcionalidad
- ☐ Ciclo de vida del Proyecto (SDLC)
- ☐ Objetivos del Negocio
- ☐ Entorno Profesional

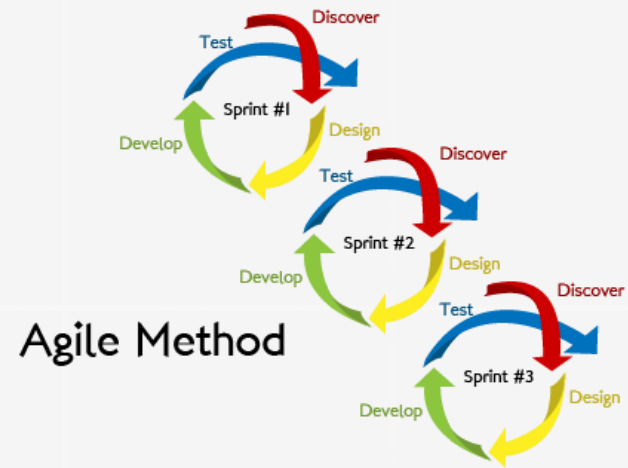


Contextos de una AS

- ☐ Requerimientos
- ☐ Diseño
- ☐ Implementación
- ☐ Pruebas
- ☐ Evolución



VS.

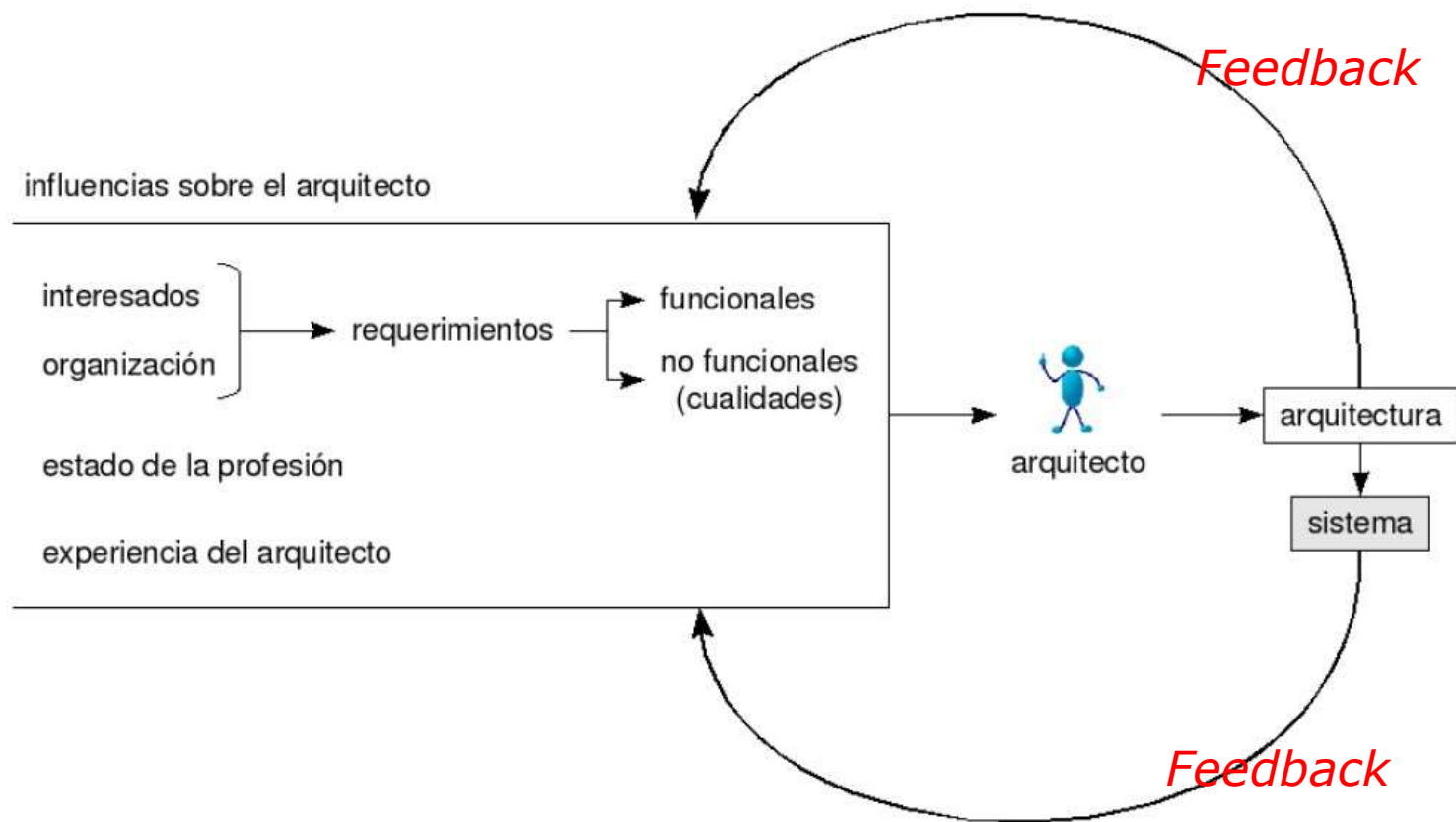


Stakeholders

- ✓ Arquitectos
- ✓ Programadores
- ✓ SQA
- ✓ Gerentes
- ✓ Clientes
- ✓ Usuarios
- ✓ Proveedores



Influencias



El arquitecto de software



... pero soy developer !!

