



Arquitectura de Software

MSc. Juan José Quesada Sánchez
ajuanjo@gmail.com

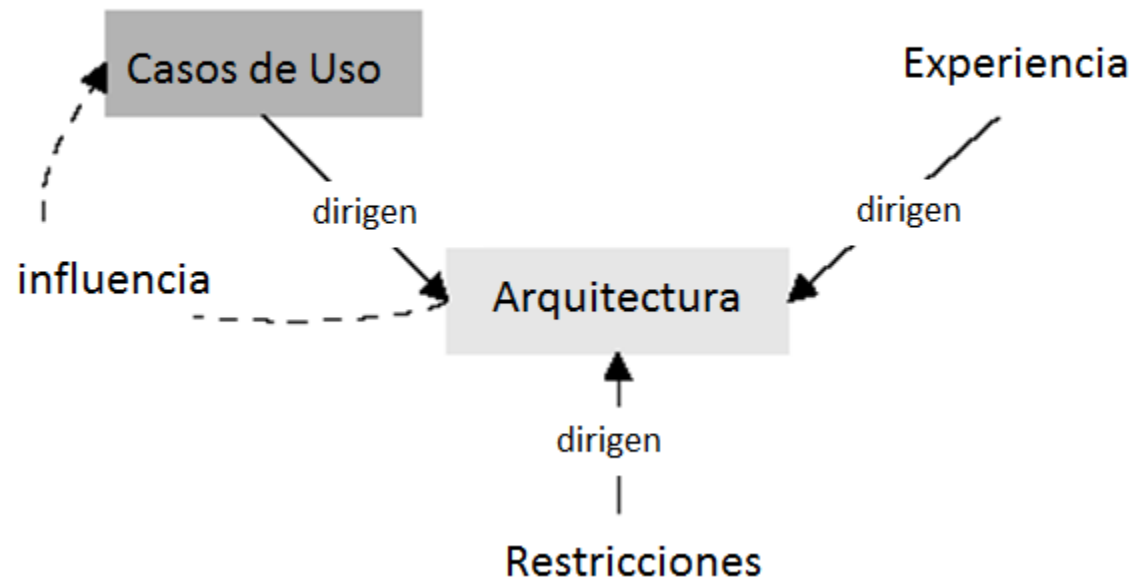
UNIVERSIDAD DE COSTA RICA
SEDE DEL ATLÁNTICO



Agenda

- ▶ ¿Por qué tener una arquitectura?
- ▶ Razones por las cuales es necesaria
- ▶ La arquitectura del software
- ▶ Características de una buena arquitectura de software
- ▶ Influencias sobre la arquitectura de software

Reflexionemos



¿Por qué tener una arquitectura?

Los requerimientos
funcionales
identifican lo que el
sistema debe hacer
(su funcionalidad)



Los requerimientos
no-funcionales
imponen
restricciones sobre
la realización del
sistema

Ambos tipos no dicen nada de cómo las funcionalidades debe ser implementadas o cómo el sistema debe ser estructurado o diseñado

¿Por qué tener una arquitectura?

- ▶ Los requerimientos del sistema han sido utilizados como punto de entrada para diseñar e implementar el sistema
 - ▶ Es una buena alternativa si se conocen bien
 - ▶ Mala si el sistema es bastante complejo
- ▶ Construcción de una casa partiendo de sus funciones(I)
 - ▶ Ofrecer una cochera para parquear dos carros
 - ▶ Tener un área para lavar la ropa, cocinar, y comer
 - ▶ Proporcionar 3 habitaciones para dormir
 - ▶ Proporcionar un área para recreación (tele y radio)

¿Por qué tener una arquitectura?

- ▶ **Construcción de una casa partiendo de sus funciones(2)**
 - ▶ Las funciones no determinan las relaciones que existen entre ellas
 - ▶ Los planos muestran las distintas vistas arquitectónicas de la casa (cableado eléctrico, aguas negras, elementos estructurales, etc)
 - ▶ ¿Será posible visualizar toda la arquitectura de la casa en nuestras mentes si nunca antes hubiésemos construido una?

Razones por las cuales es necesaria

▶ *Entender el sistema*

- ▶ El software a desarrollar puede ser grande y complejo
- ▶ La arquitectura provee un modelo o plano del sistema
- ▶ También, se abstrae de los detalles de implementación
- ▶ Se concentra en los elementos (cajas-negras) que brindan las funcionalidades

▶ *Organizar el desarrollo*

- ▶ Ayuda a separar los elementos del sistema
- ▶ Identifica las relaciones entre los elementos: las intersecciones son bien documentadas y especificadas

Razones por las cuales es necesaria

- ▶ ***Promueve el reuso***

- ▶ Ayuda a identificar los sistemas y subsistemas críticos de forma temprana: los subsistemas comunes puede ser reutilizados

- ▶ ***Favorece el desarrollo continuo***

- ▶ *Los sistemas evolucionan en el tiempo: nuevos requerimientos surgen o cambios son solicitados*
 - ▶ *La arquitectura original puede colaborar en la evolución: una buena arquitectura requerirá pequeños ajustes en el ciclo de vida*

Razones por las cuales es necesaria

- ▶ *Favorece la comunicación entre los interesados*
 - ▶ Los interesados mantienen distintas perspectivas sobre el software siendo desarrollado
 - ▶ Sus intereses deben ser reflejados en la arquitectura

La arquitectura de software

- ▶ **Arquitectura de sistemas de información**
 - ▶ “La arquitectura de un sistema de información, por lo general grande, que puede contener subsistemas. La arquitectura se refiere a la estructura o estructuras del sistema, que comprenden los elementos del sistema, las propiedades visibles externamente de esos elementos y las relaciones entre ellos” [Kurbel 2008]

La arquitectura de software

- ▶ Hunt [2003] delimita los aspectos que constituyen la arquitectura de software
- ▶ **El plan total para la estructura del sistema**
 - ▶ Un modelo del sistema
 - ▶ Debe poder leerse y entenderse como el sistema está estructurado sin conocer los detalles de los elementos
- ▶ **Los elementos estructurales claves y sus interfaces**
 - ▶ Qué elementos constituyen el sistema, sus interfaces y cómo se conectan entre ellos

La arquitectura de software

- ▶ **Cómo estos elementos forman sistemas y subsistemas**
 - ▶ Aspecto clave
 - ▶ La identificación temprana de los sistemas y subsistemas centrales permite organizar el diseño y promueve el reuso y la comprensibilidad
- ▶ **El estilo arquitectónico que guía la organización**

La arquitectura de software

- ▶ “The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution”.

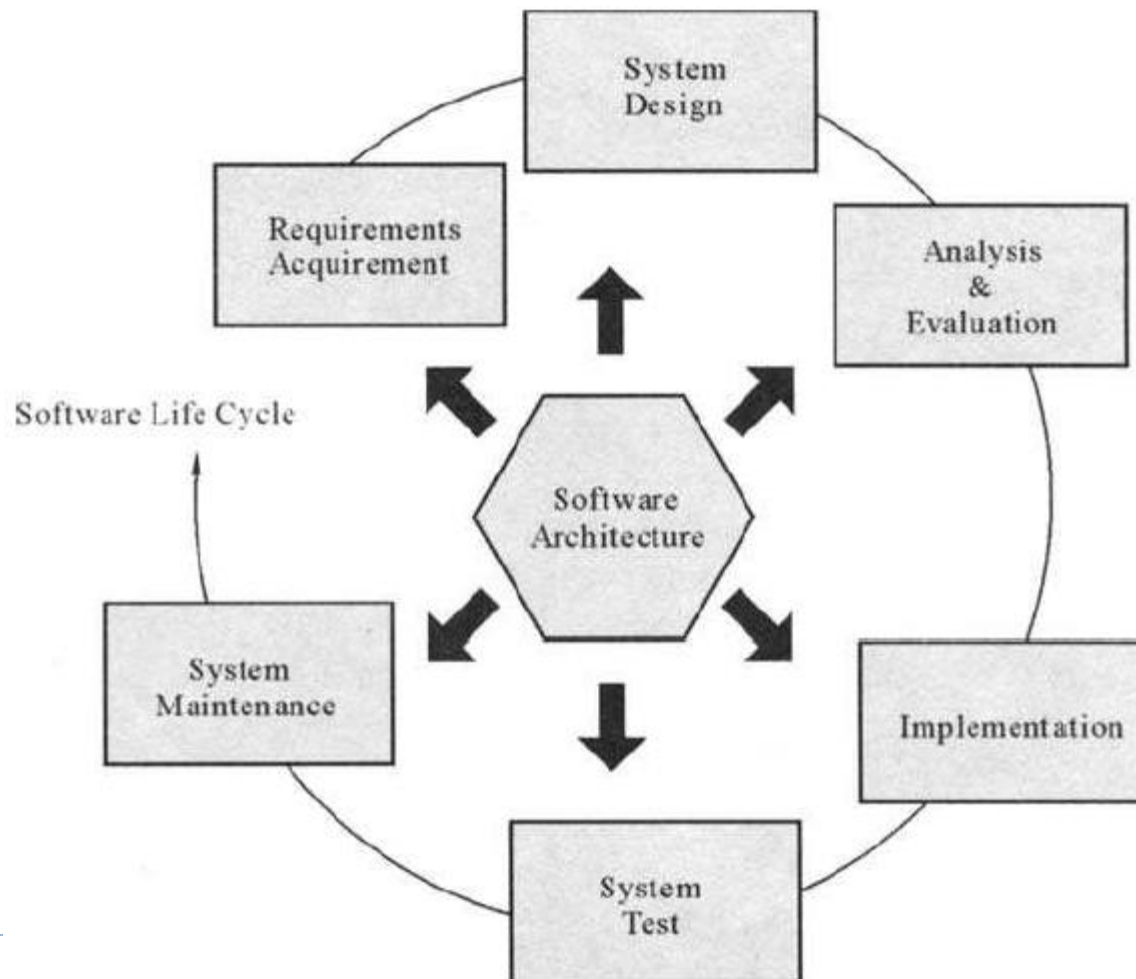
IEEE 1471--2000

La arquitectura de software

- ▶ SA = {Componentes, Conectores, Restricciones}
- ▶ La arquitectura de software se *avoca a las interacciones entre las unidades* en un sistema (estructura tiempo-ejecución)
 - ▶ **Unidades (*componentes*)**: cualquier unidad que ejecute servicios predefinidos y se comunique con otros componentes
 - ▶ **Conectores**: define las estrategias y protocolos de comunicación
 - ▶ **Restricciones**: define las reglas sobre las cuales el sistema es constituido

La arquitectura de software

► Ubicación en el ciclo de desarrollo del software



Características de una buena arquitectura de software

▶ *Resistencia*

- ▶ *Debe ser resistente al cambio*
- ▶ *Los cambios en la funcionalidad o funcionalidades adicionales deberían tener un efecto mínimo en la arquitectura*
- ▶ *Los subistemas, por ejemplo, deberían tener claras las interfaces*

▶ *Simplicidad*

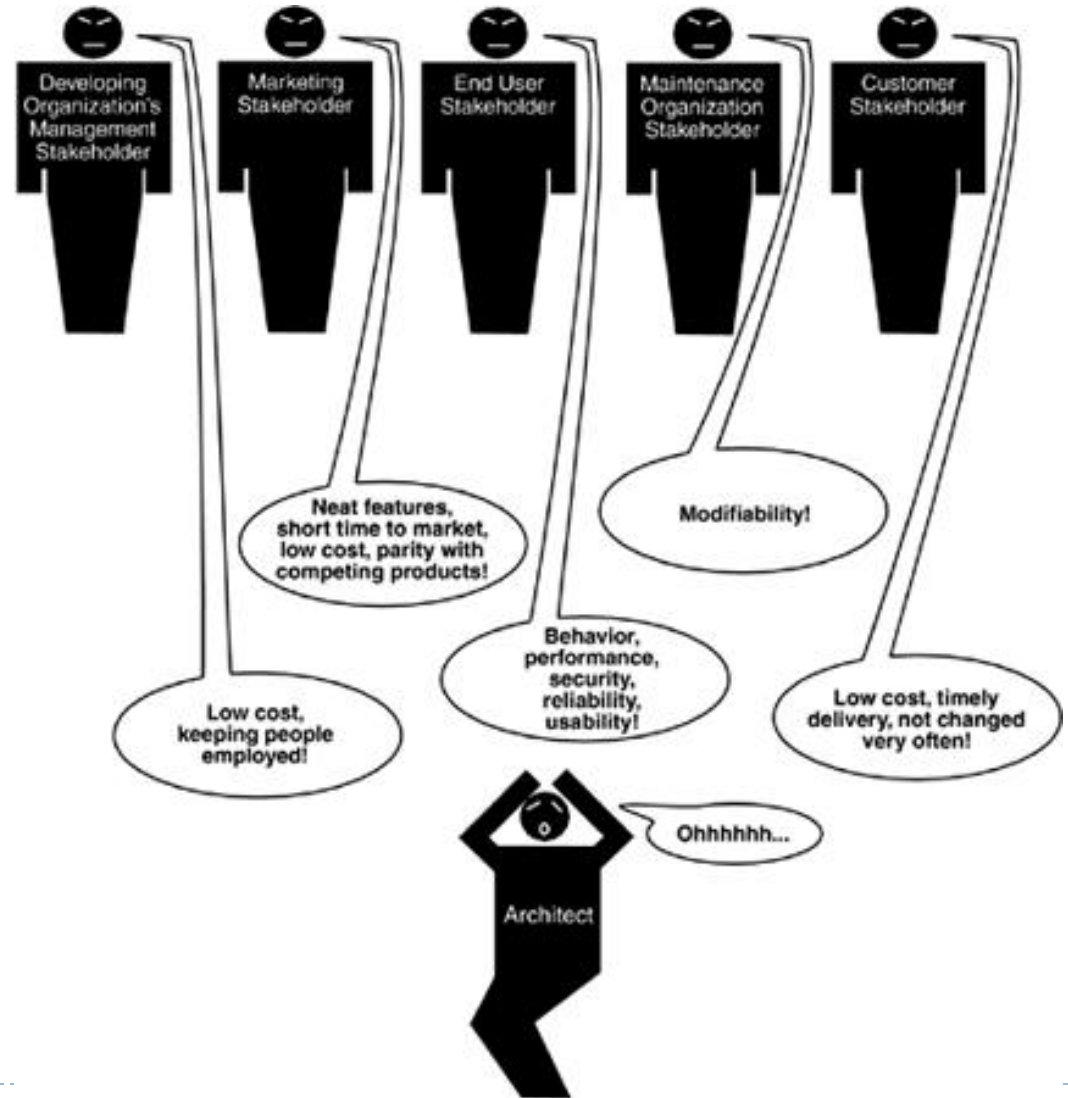
- ▶ *Una regla empírica dice que la arquitectura debería representar sólo el 10% de todo el diseño*
- ▶ *Debe ser comprensible*

Características de una buena arquitectura de software

- ▶ **Claridad de presentación**
 - ▶ *Dado que es accedida en el resto del diseño y otras iteraciones debe ser carente de ambigüedad y ser accesible*
- ▶ **Clara separación de intereses**
 - ▶ *Debe existir una separación clara de los distintos aspectos del sistema*
- ▶ **Balancea las restricciones económicas y las tecnológicas**

Influencias sobre la arquitectura de software

- ▶ Los afectados tienen distintos intereses y objetivos acerca del sistema las cuales pueden ser contradictorias
- ▶ Estos intereses pueden ser contradictorios



Influencias sobre la arquitectura de software

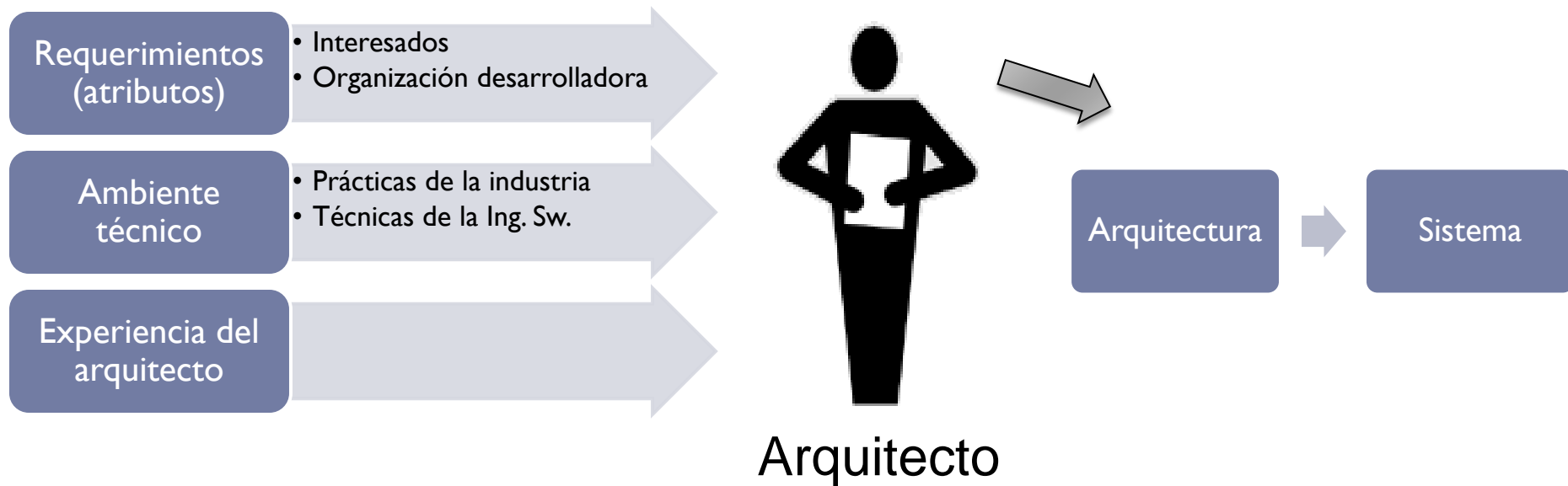
- ▶ **Influencia de la organización que la desarrolla**
 - ▶ Ejs. Conocimientos que los empleados tengan, el cronograma de desarrollo y el presupuesto
 - ▶ Negocios inmediatos: reutilizar arquitecturas definidas en sistemas anteriores para proponer un nuevo producto
 - ▶ Estructura organizacional: externalización (outsourcing) de partes de la arquitectura

Influencias sobre la arquitectura de software

- ▶ **Influencia del bagaje y experiencias del arquitecto**
- ▶ **Influencia del ambiente técnico actual**
 - ▶ Incluye: prácticas de la industria, técnicas de la ingeniería de software usadas por el arquitecto
 - ▶ ¿Cuál arquitecto no consideraría una solución basada en web, orientada a objetos o separación lógica en capas?

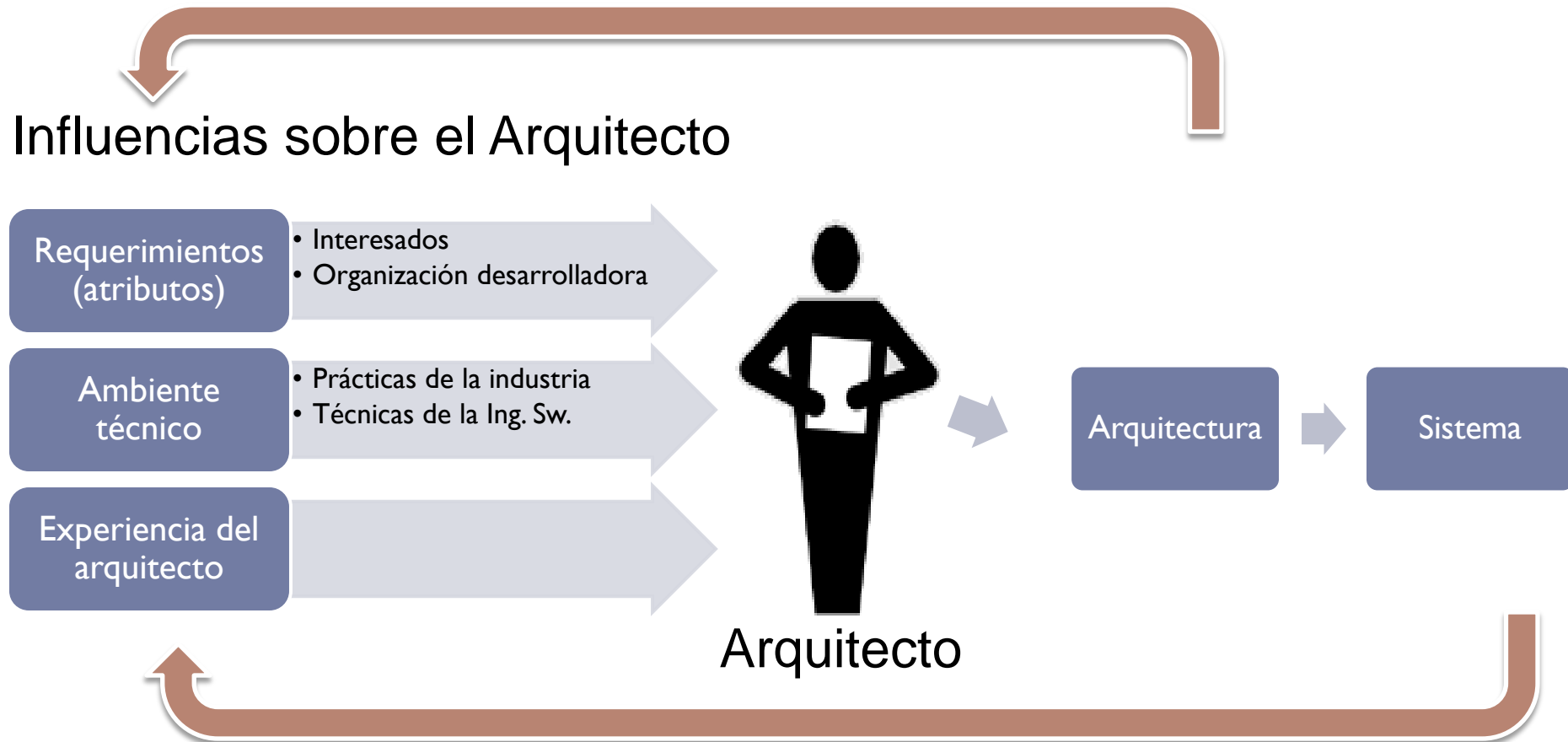
Influencias sobre la arquitectura de software

Influencias sobre el Arquitecto



Influencias sobre la arquitectura de software

La arquitectura afecta los factores que la influyen



Referencias

- ▶ Mena, A. (2011). **Los Procesos: Documentación, Modelados alternativos**. Curso Ingeniería de Software, Universidad de Costa Rica
- ▶ Bass, L.; Clements, P.; Kazman, R. (2003). **Software Architecture in Practice**. Segunda edición. Addison Wesley.
- ▶ Hunt, J. (2003). **Guide to the Unified Process featuring UML, Java and Design Patterns**. UK: Springer. {Cap. 4}
- ▶ Kurbel, K. E. (2008). **The Making of Information Systems: Software Engineering and Management in a Globalized World**. Germany:Springer-Verlag Berlin Heidelberg. {Cap. 3}

