A 3D rendering of a warehouse conveyor belt system. Several cardboard boxes are shown on the belt, moving from left to right. The floor is dark blue with a red grid pattern. The lighting is dramatic, with strong highlights and shadows, creating a futuristic and industrial atmosphere. The text is overlaid on the left side of the image.

On the role of a Software architect

EL PAPEL DEL ARQUITECTO DE SOFTWARE -
ENTREVISTA SE RADIO 616

ARQUITECTURA DEL SOFTWARE

GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE
- UNIVERSIDAD DE OVIEDO

Integrantes

Jorge Puente García

U0294228@uniovi.es

Manuel García Baldó

U0295497@uniovi.es

Ignacio Llaneza Aller

U0294914@uniovi.es



Sobre la Entrevista

- Entrevista *SE Radio 616: Ori Saporta on the Role of the Software Architect*

*Presentador: Jeff Doolittle | Invitado: Ori Saporta
(Cofundador y arquitecto de sistemas en vFunction)*

- Introducción

- **Ori Saporta:** Cofundador de **vFunction**, experto en arquitectura de sistemas, y experiencia previa en **WatchDox** (adquirido por BlackBerry).
- **Tema central:** El papel fundamental del arquitecto de software en el diseño, desarrollo y mantenimiento de sistemas complejos.



MAY 15, 2024

SE Radio 616: Ori Saporta on the Role of the Software Architect

LISTEN ON:



Evolución para ser un arquitecto del software.

No se comienza siendo un arquitecto

- El camino empieza desde abajo

El arquitecto de software es una decisión

- Pasión por el trabajo en conjunto
- Tomador de decisiones
- Visión más amplia del aspecto técnico

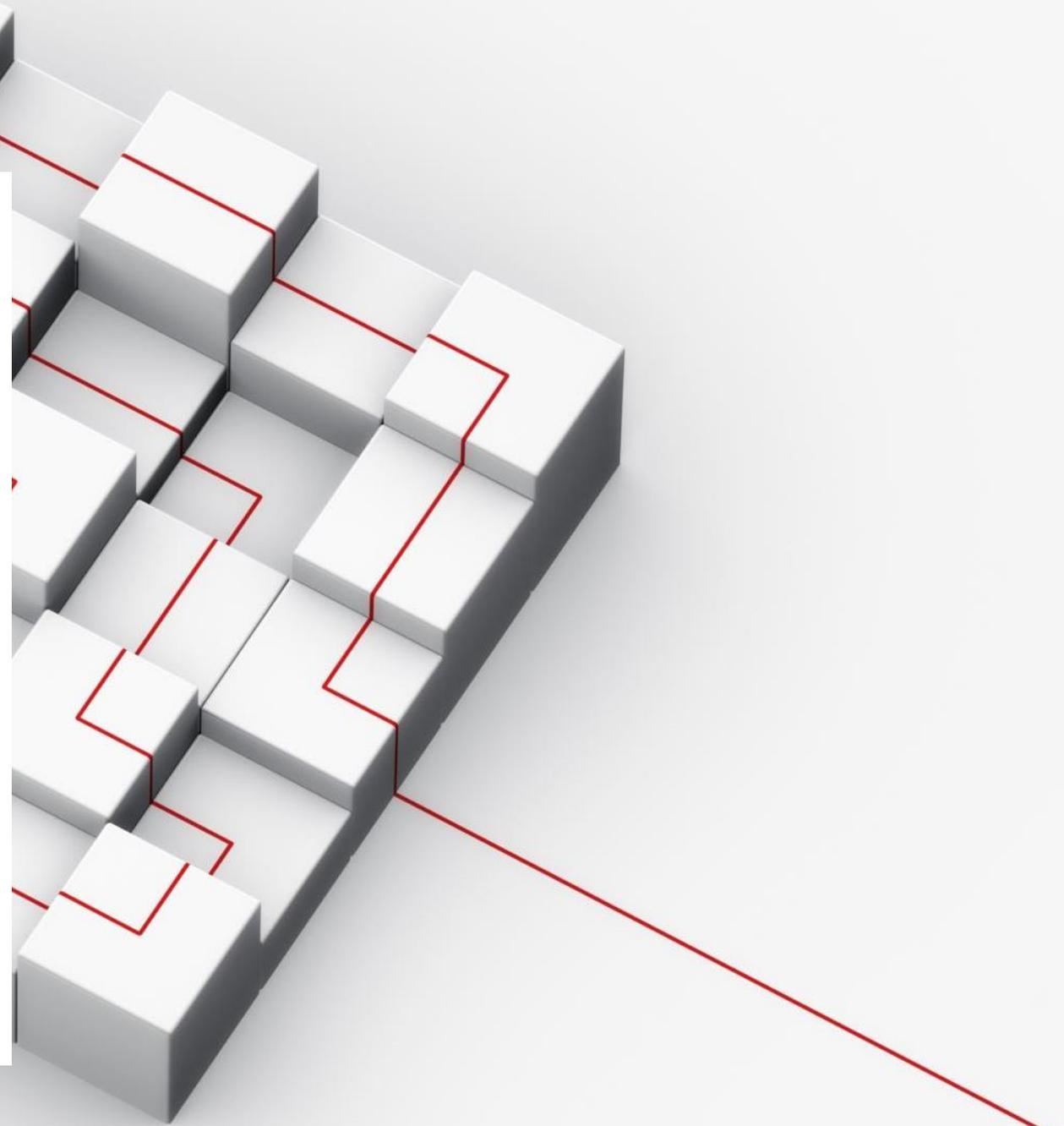
Lado técnico y lado humano

Habilidades esenciales del arquitecto de software



Habilidades técnicas (Hard Skills)

- Capacidad y habilidad técnica.
- Punto de partida para muchos arquitectos.



Habilidades Interpersonales: Soft Skills



Escucha activa y traducción:

- Buen oyente: entender los problemas del lado técnico y del lado comercial.
- Traduce necesidades técnicas y de negocio → comprensión entre stakeholders y desarrolladores.

Ambos mundos

- Debe comprender ambos mundos para hacerse las preguntas correctas.

Facilitador del cambio

- Ayuda a los equipos a aceptar y adaptarse a nuevas estrategias tecnológicas.

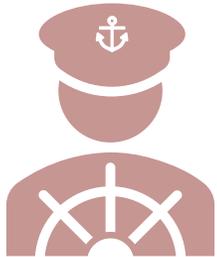
Reconducir

- Todos deben reconocerse por su trabajo.

Confianza

- Requiere más confianza escuchar que pisar otra idea.

Mediador vs. Guardián



Ori hace una distinción importante entre ser:

Un "guardián" - proteger el diseño de los malos requisitos

Un "mediador" - ayudar a los interesados a entender y llegar a los requisitos correctos



El arquitecto no debe imponer sus opiniones, sino facilitar el entendimiento mutuo.

El inicio de Saporta

Llegada
sin habilidades
técnicas

- Las habilidades blandas vienen del interés
- Compartir tu pasión entre personas técnicas y no técnicas

Persona a la que
todos acueden

- No necesitas don de gentes

Papel del
arquitecto del
software a lo
largo del tiempo.



De waterfall a Agile

- Modelo Waterfall
 - Había un **diseño exhaustivo** al principio.
 - El arquitecto era considerado el "rey" del proyecto.
- Agile:
 - Busca flexibilidad
 - Parece reducir el diseño a un rol consultivo, lo que puede llevar a una falta de visión global del sistema.
 - El software como **algo vivo**.

El cambio

El cambio en el contexto del software:

- El **cambio** es una constante en el desarrollo de software.
 - *Requisitos* que **evolucionan**.
 - *Tecnologías* que **avanzan**.
 - **Cambios** en el *negocio*.

El arquitecto tiene que estar **preparado**.

Arquitecto del software y el cambio

<<Dicen que el puente que se derrumba es el puente que no se mueve con el viento.>>

El arquitecto diseña para el cambio.

- Diseño dinámico y adaptable.
- Arquitectura flexible que permita evitar conflictos.

Facilita el cambio:

- Es un "puente" entre los diferentes roles.
- Ori Saporta destaca que los arquitectos deben ser abiertos de mente.
- Se hacen analogías como la **del puente y el viento** o la de los edificios y los terremotos.

Revisión continua:

- El arquitecto acompaña al proyecto y modifica el diseño como si fuera un ser vivo.

Diseño continuo y evolutivo

El diseño es evolutivo.

- Diseño \neq Evento único.
- Arquitecto **siempre presente** el proceso.
- Evita **deuda técnica**

Debe adaptarse a cambios en requisitos y tecnologías

- **Evolución organizada**
- Flexible ante los cambios

El diseño de la arquitectura **afecta**:

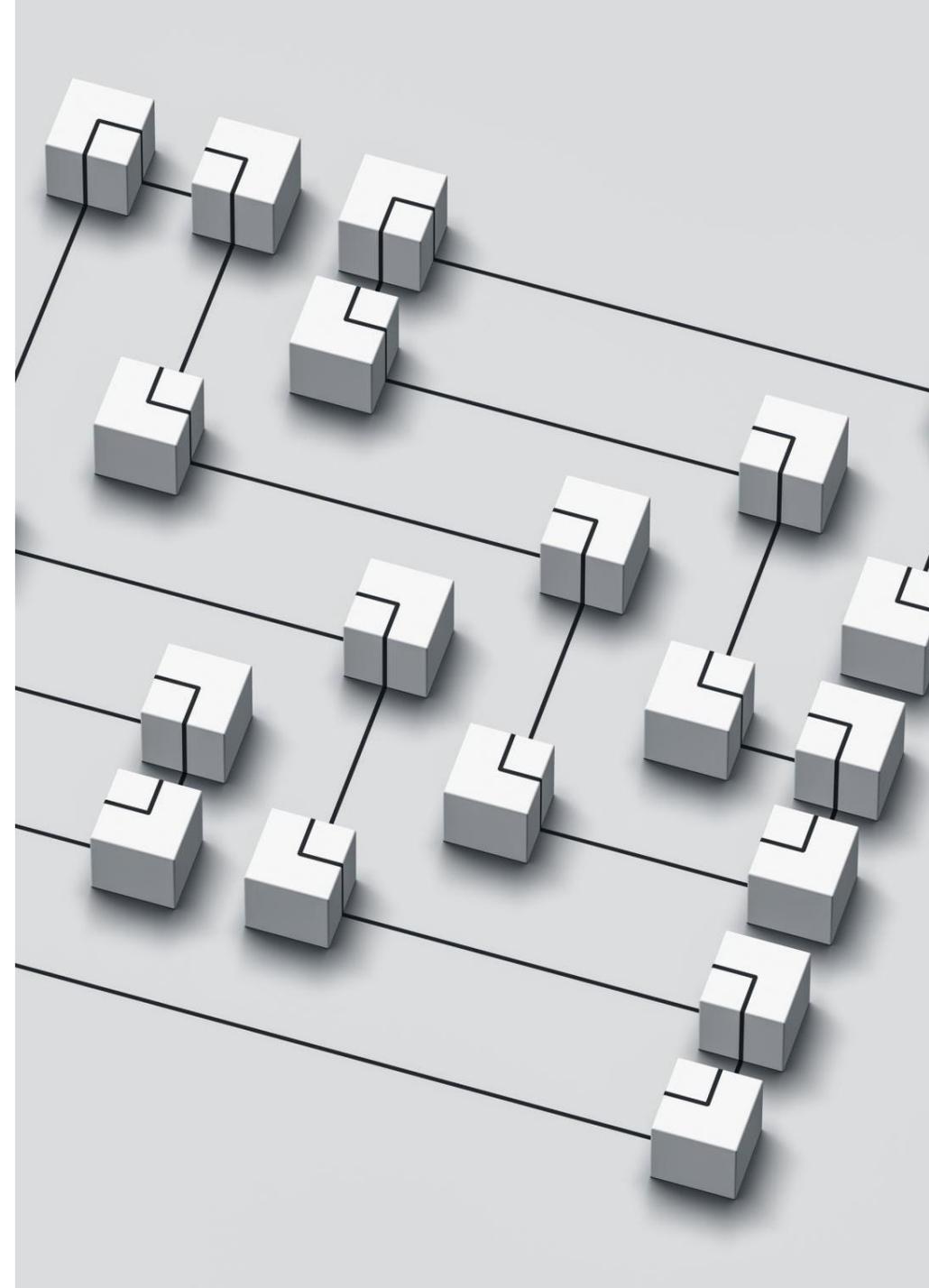
- **Agilidad** del proyecto
 - Evolución del proyecto.
-

Mala gestión del arquitecto: Diseña y desaparece

Pérdida de control y visibilidad: *No se puede validar* si el diseño fue implementado correctamente.

Falta de iteración: Un diseño puede *necesitar ajustes* debido a nuevas características o cambios en los requisitos.

Dificultades en la escalabilidad: Un *mal diseño inicial* que *no evoluciona adecuadamente* puede convertirse en un obstáculo técnico.



Calidad

Deber del arquitecto

La base sobre la que se construye.
No es algo binario

La calidad es responsabilidad de todos.

Se consigue con buenas practicas,
pudiendo mejorarlo siempre.

La calidad de tu sistema te limita.

Observabilidad Arquitectónica

Visualización y gestión de dependencias en el software.

1

Permite identificar relaciones entre clases, bases de datos y servicios externos.

2

Localiza dependencias problemáticas y promueve modularidad (dependencias localizadas).

3

Facilita la priorización de problemas para optimizar el desarrollo.

Gestión de dependencias

01

Los arquitectos deben gestionar dependencias entre módulos y servicios.

02

Evitar acoplamientos innecesarios que dificulten cambios futuros.

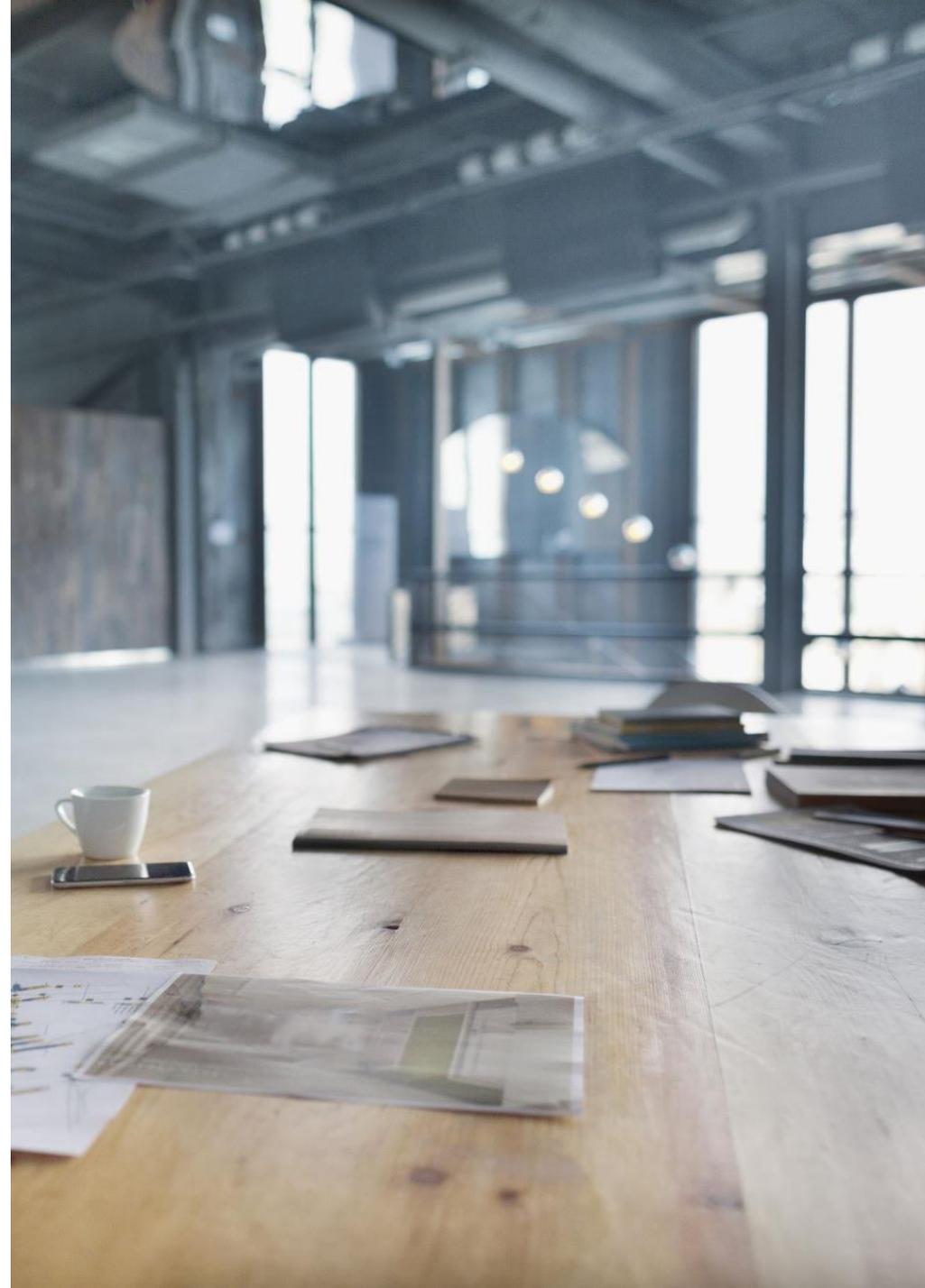
03

Diseñar con flexibilidad para permitir iteraciones y mejoras continuas.



vFunction

- Empresa que proporciona observabilidad arquitectónica.
- Ayuda a arquitectos a comprender software complejo.
- Facilita la modernización continua.
- Ayuda a la gestión de deuda técnica.
- Gestión de dependencias, consiguiendo modularidad
- Mantiene el diseño del software actualizado



Conclusión

Arquitecto de Software es en última instancia un “Decision maker”.



Comprensión del problema +
Visión del sistema.



Genera valor de negocio:
Transforma los problemas en
soluciones técnicas.



Participación continua: Clave
en todo el ciclo de vida de la
aplicación.



Influye en decisiones
estratégicas:

Diseño, Escalabilidad y Mantenimiento.



Preguntas

MUCHAS GRACIAS

Fin

