

I don't need an Architect
I'm Agile



Un ringraziamento speciale ai nostri sponsor



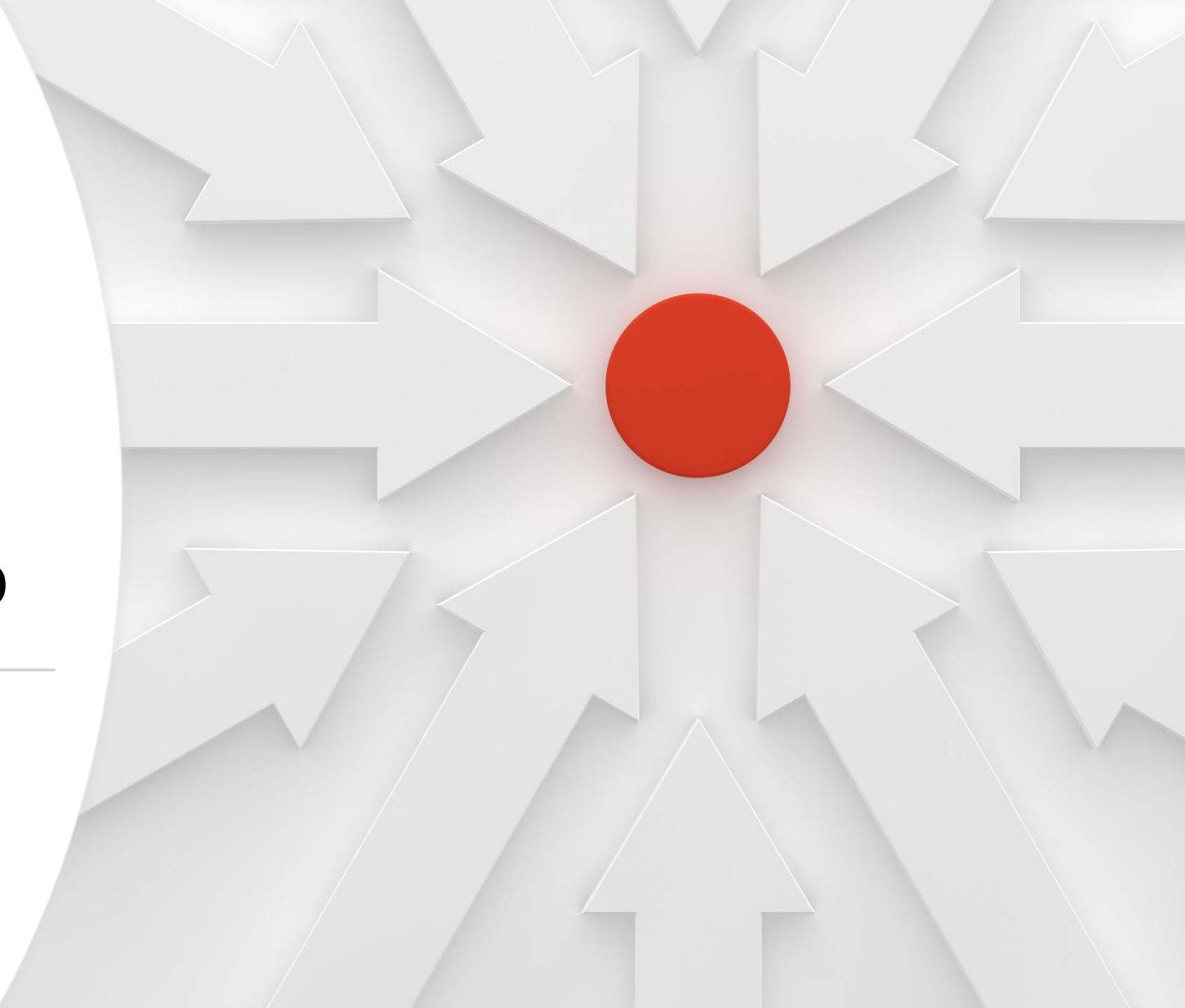
A Software Architect in an Agile world

In origine era il Waterfall

Tutti vivevano felici per
mesi o talvolta anni

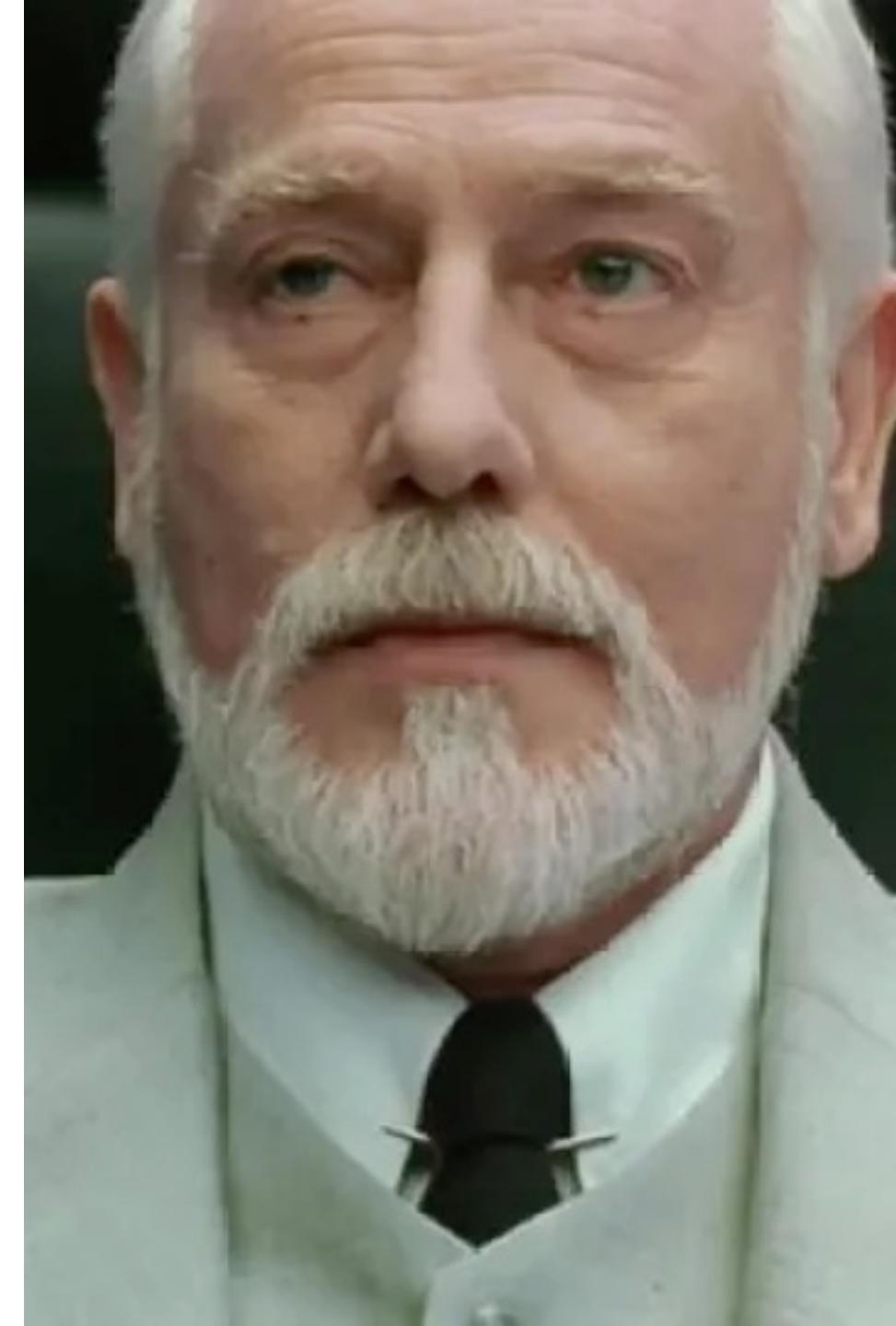


L'avvento del
movimento
Agile ha
cambiato tutto



No Software Architects!

- Per l'XP, se un ruolo o un'attività non contribuiscono direttamente allo sviluppo di codice eseguibile, sono controproducenti e vanno eliminati.
- L'approccio è quello di avere la cosiddetta Architettura Emergente, pertanto i Software Architects sono inutili
 - “The best architectures, requirements, and design emerge from self-organizing teams”



Ma è realmente così?

Leggiamo alcuni punti del manifesto



Gli individui e le interazioni più che i processi e gli strumenti



Il software funzionante più che la documentazione esaustiva



La collaborazione col cliente più che la negoziazione dei contratti



Rispondere al cambiamento più che seguire un piano

Rispondere al cambiamento



Accogliamo i cambiamenti nei
requisiti, anche a stadi avanzati
dello sviluppo.



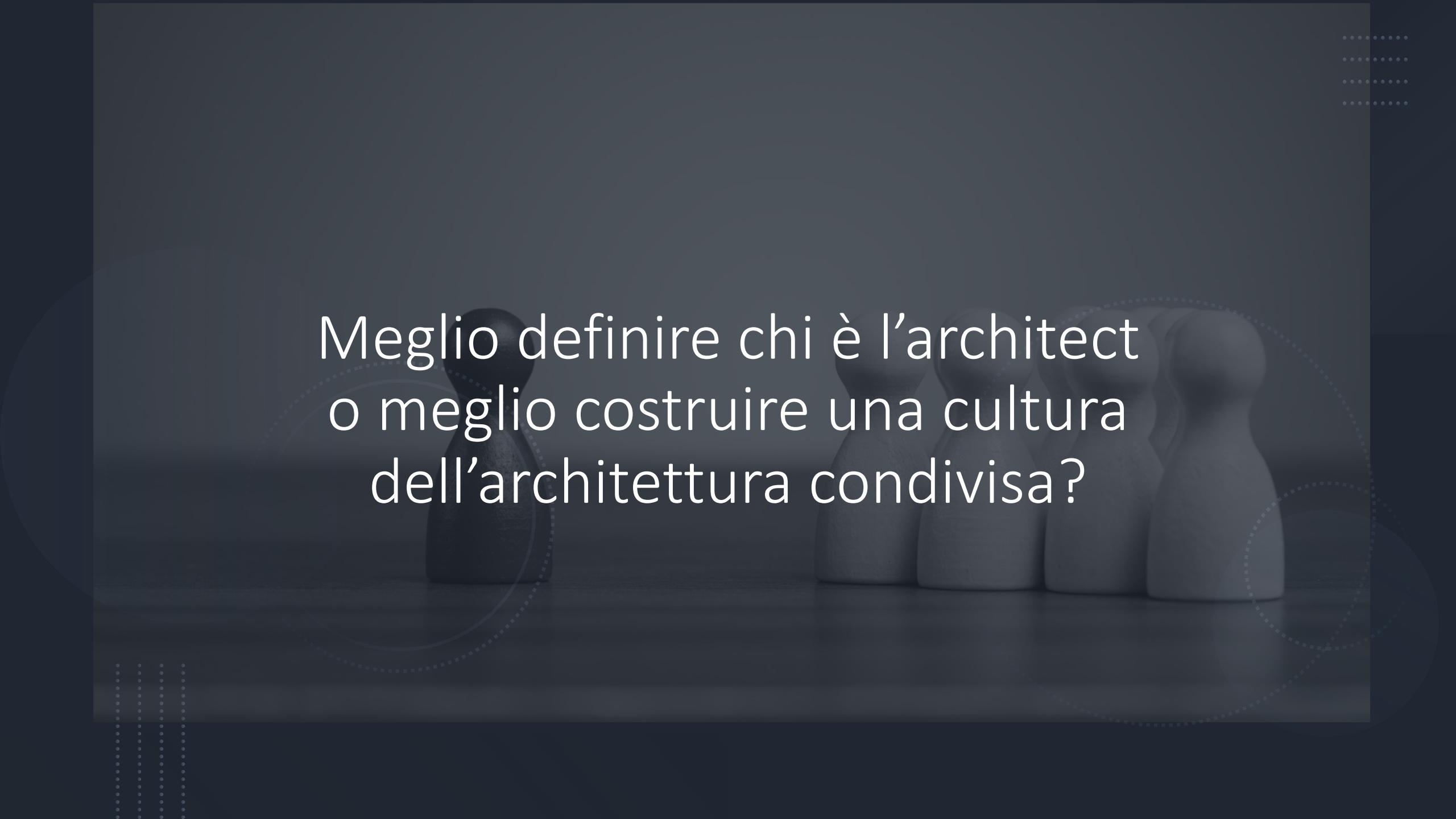
Le architetture, i requisiti e la
progettazione migliori emergono
da team che si auto-organizzano.

Ok, ma quindi l'architetto?

Qualcuno inizia a pensarci seriamente

- SAFe introduce l'Architecture Runway
 - Architettura Intenzionale
 - Design Emergente
- Large-Scale Scrum (LeSS) prova a combinare agile e architecture insieme
 - Technical Excellence: design patterns, dependency injection, test-driven development, refactoring, etc...
 - Cultura del coaching-while-coding da parte dei technical leaders
 - Gli architetti fanno parte del team
- Disciplined Agile Delivery (DaD) include il ruolo di Architecture Owner
 - Facilita la creazione dell'architettura, senza forzarla
 - Coaching
 - Technical priorities



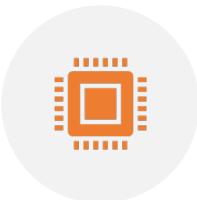


Meglio definire chi è l'architect
o meglio costruire una cultura
dell'architettura condivisa?

Possiamo pensare
che l'architettura sia
sempre in Continua
Evoluzione?



Principi della Continuous Architecture



Architect products;
evolve from project
to products



Focus on quality
attributes, not on
functional
requirements



Delay design
decisions until they
are absolutely
necessary



Architect for change
– leverage the
“power of small”



Architect for build,
test, deploy, and
operate



Model the
organization of your
teams after the
design of the system
you are working on

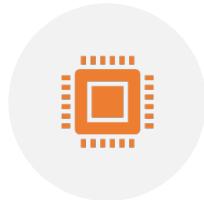
Architect for Change: alcuni approcci

Hexagonal
Architecture (Clean
Architecture)

Evolutionary
Architecture

Microservices

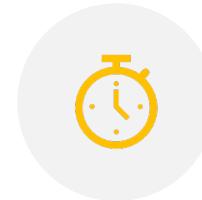
Principi della Continuous Architecture



Architect products;
evolve from project
to products



Focus on quality
attributes, not on
functional
requirements



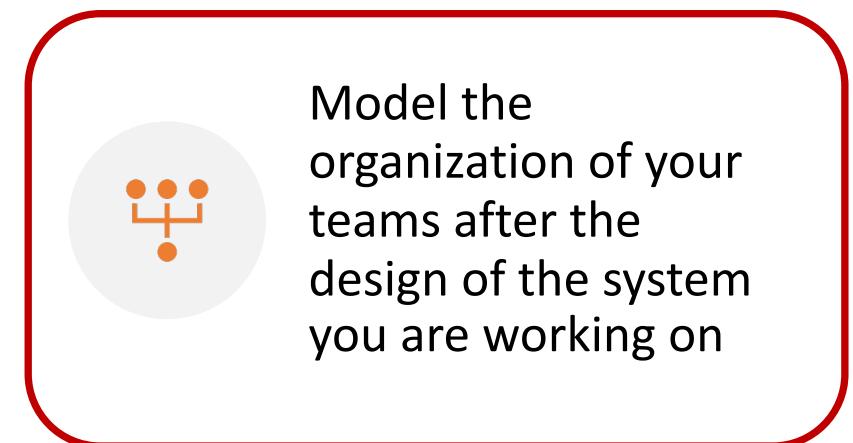
Delay design
decisions until they
are absolutely
necessary



Architect for change
– leverage the
“power of small”

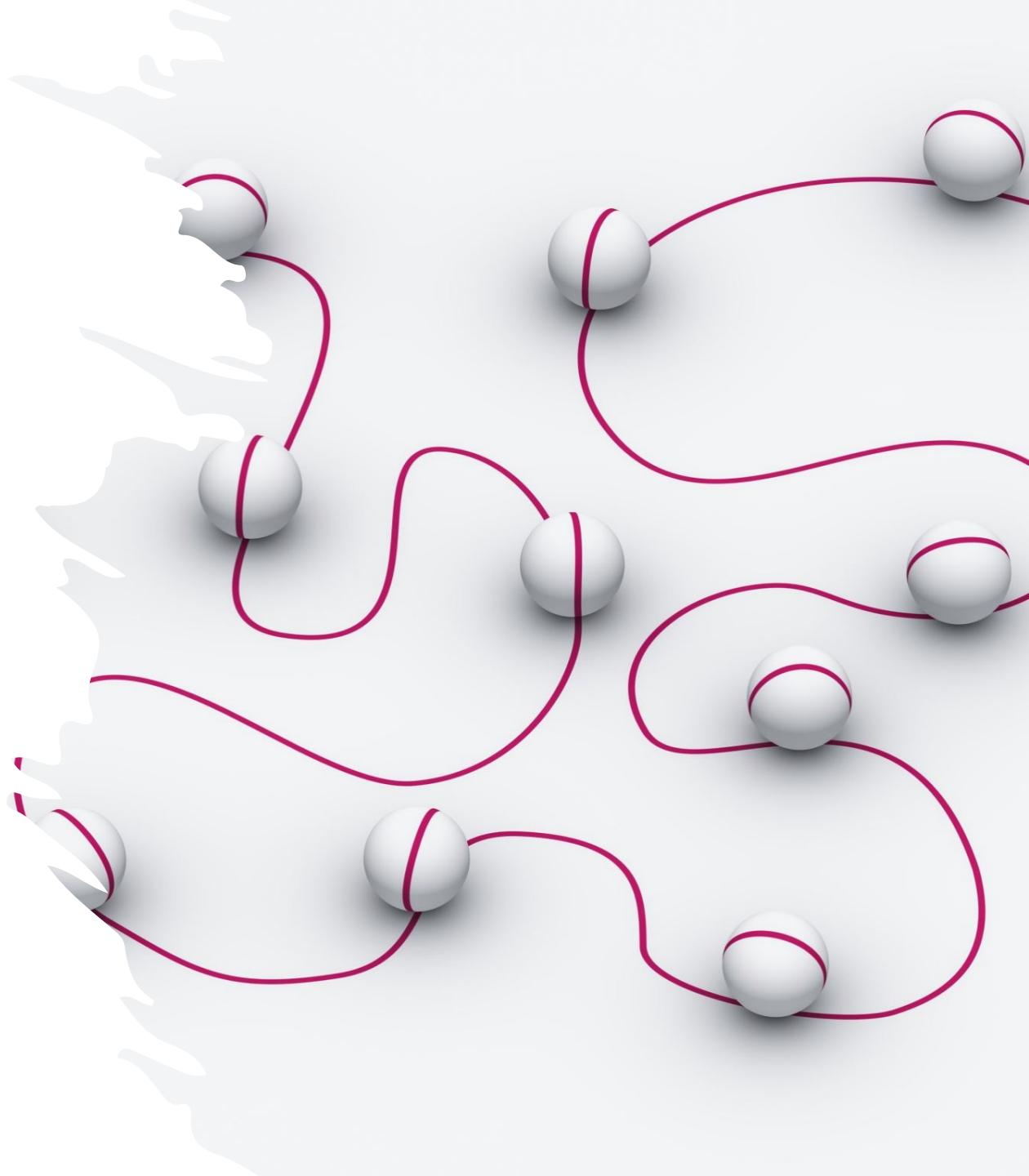


Architect for build,
test, deploy, and
operate



“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

Conway's Law



L'Architect e la topologia dei teams



GARANTIRE L'AUTONOMIA
DEI TEAM



CONTINUOUS DEPLOYMENT



MODELLO COMUNICATIVO
(TRA TEAM / COMPONENTI)

Key takeaways

- L'architect guarda anche al modello organizzativo e come facilitatore dei lavori architetturali, una guida più che un comandante
- L'architettura come un organismo in continua evoluzione / adattamento
- Adozione di pratiche e strumenti come Architectural Decision Records, RFC, etc...
- Quality attributes e feedback loop



Contatti e riferimenti

- LinkedIn
<https://www.linkedin.com/in/cozzolinofabio/>
- Continuous Architecture
<https://continuousarchitecture.com/>



Grazie ☺

