Agile approaches to architecture : transition to microservices

Brian Wehrle
Software Architect

Overview

- Bio
- Goals and impacts of microservices
- Path forward
- How architects can support the transition
- Decentralized governance
- Summary

Bio

Brian Wehrle

- Software architect with background in ecommerce and data systems
- From US, been living in Olesa de Montserrat (BCN) for last 11 years

Companies work for:

- Vistaprint
- Xerox
- Microsoft

Organizational motivations for microservices

- Motivations for microservices are to reduce team dependencies, accelerate development and facilitate collaboration
- This often implies the following for dev teams:
  - Make (smart) platform decisions (language, deployment)
  - Greater autonomy
  - Increased system complexity for lower software complexity
  - Decentralization of operational support and tech decisions

Technology aligned to business goals

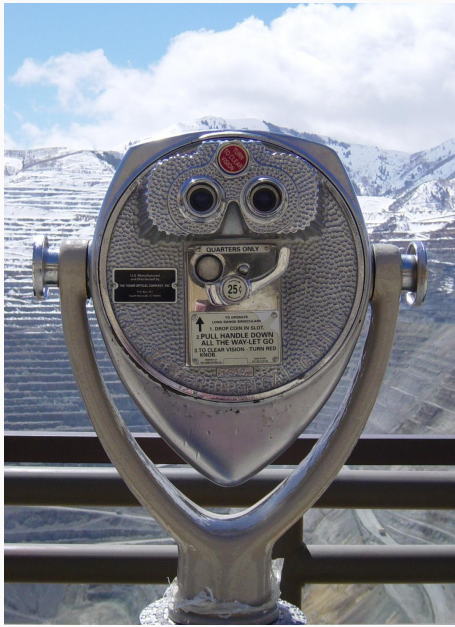
Microservices can be thought of as a way to organizing technology to best support business goals. Technology is an enabler of this vision.

- Faster, flexible releases
- More Agility
- Value can be delivered earlier and more frequently

Looking forward

So, how can we get to this state from an existing stack and centralized way of working?

- Most companies need to go through a technology and organizational evolutionary process lead by architects and motivated teams
- The key for supporting the Agile principles is the strong collaboration between dev and architecture
- Socializing and creating enthusiasm for new org vision



Some impediments & challenges

- Emergent design at team level is not necessarily efficient for large scale changes
  - “Best architectures, requirements, and designs emerge from self-organizing teams” (Agile manifesto)
- Detailed low-level up-front designs produced by architects
- Gaps in technical knowledge
- Team expectations and microservice principles
  - “Organization around business capabilities”
  - “Decentralized governance”

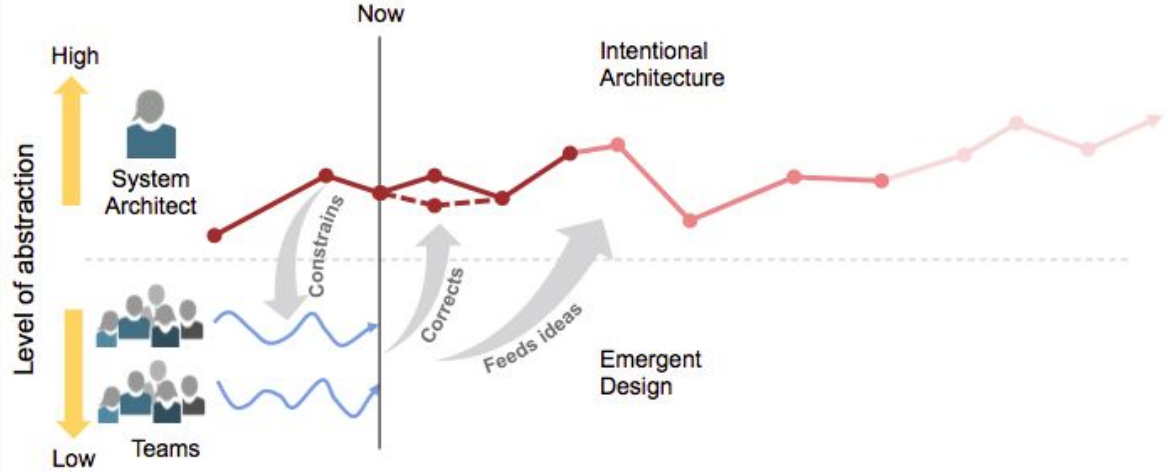
Architect support for this transformation

How do architects / enterprise architects support this change in organization and technology while respecting Agile ideals? Here are five ideas:

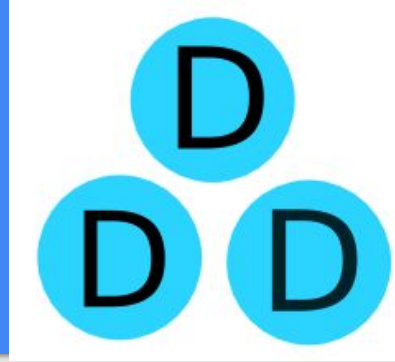
1. “Intentional Architecture” - Scaled Agile Framework
2. Start a domain model
3. Empowering teams through shared responsibility
4. Open guardrails
5. Iterate and improve

“Intentional Architecture”

- Scaled Agile Framework take on Architecture
  - Introduces the concept of “Intentional Architecture”
  - Proposes way to apply Agile at large organizational level
  - Emergent design can work well at team level, but not for major rethink



Model the system



A good microservices design is based on a good understanding of the domain. If you don’t have a domain design or if you have a messy monolith, build a clear map of the services.

- Services should have clear responsibilities
- Getting the granularity right is important
- Review “Domain Driven Design” if your system is complicated or many different functional areas
- https://www.thoughtworks.com/insights/blog/domain-driven-design-services-architecture

Open guardrails

Technology guardrails are tech stack decisions that help limit the diversity of an environment and limit autonomy for efficiency or fungibility. This can provide a good starting point for teams with less experience.

- Define goals that align to ownership model, fungibility and cost/benefit - socialize these goals
- Share the reasoning with teams, create a space for dialogue and feedback
- Pick best of breed tools and make these the baseline picks
- Define an evolutionary path where the team can challenge and experiment to discover better solutions.

Empowering through shared responsibility

- Consistently top-down design decisioning won’t foster ownership
- Bringing the teams into the decisioning flow at appropriately increasing levels of responsibility is a practical approach
- Architecture team should recognize how to support and mentor the teams and learn from them, zooming in and out as needed.

Iterate and improve

- Start small and work with a reduced part of the org
- Experiment with changes, take feedback and iterate
- Pick a space where failure will be acceptable
- Take some interesting risks

“If the project ‘can not fail’, both the success and learnings will be limited.”

- Anonymous

Next station: Decentralization of governance

At a certain point, the teams will need less structure and external support.

- Internalized guard rails as a set of business principles, teams improve these and both initiate and collaborate in future cost / benefit decisions across the organization.
- Take responsibility for tech and arch decisions, using a pull model to reach out for support when there are concerns.
- Participate actively in the “Intentional Architecture”, through Guilds and other structures.

Summary

- Microservices and Cloud create new challenges
- Fully leveraging these ideas implies important org and stack changes
- Self-organization / decentralized governance can arise, but this should be an inclusionary process based on strong collaboration

Questions?

Email: bwolesa@gmail.com

Source: https://github.com/bwehrle/Presentations