What Agile Architects Do and What They Need

Viktor Clerc
Rik Farenhorst
Daan Kalmeijer
Who we are

- Inspearit:
  - Consultancy and Training
  - Architecture, Process Improvement, Software Quality, Security, ...
  - Based in France, the Netherlands, Italy, Singapore, China
- Daan Kalmeijer:
  - Enterprise Architect
  - Defense, Finance, Public Security, Healthcare, ...

2
When Traditional and Agile meet ...
Building very Large and Complex Systems
Agile Architecture as a Paradigm Shift

- Traditional architecture focuses on rules, standards and guidelines, limiting the solution space of development projects.

- Helpful for projects that ‘cash’ on the long term
  - Big design upfront
  - “Grand” projects

- Traditional architecture is not tolerant to ambiguity

- “Embracing change” becomes a challenge. A focus on tangible results equally

- Traditional Architecture is **deterministic, instrumental and reductionistic**
Agile and Architecture

Support and promote Agile projects

Understand and promote Agility on an organizational level

Architecture should play a supportive role in (agile) software development

[Madison 2010, IEEE Software]
Support and promote Agile projects

*Adaptive Architecture, not Emergent Architecture!*
Of course, most Architecture frameworks and methods can be used in an Agile way. Most are incremental and iterative.

As frameworks get larger, implementing them becomes more of an investment. Questioning and changing the way of working isn’t an option any more.

**We don’t need an Agile Architecture Framework or Method**
Architecture Frameworks and Methods

“**Learn the form, but seek the formless.** Hear the soundless. Learn it all, then forget it all. Learn The Way, then find your own way”

- The Silent Monk in The Forbidden Kingdom
Agile Architecting Practices

1. Be part of the development team
2. Never slow down the project
3. Communicate often and early
4. Travel Light, Focus on the Essence
5. Architecture needs testing too!
6. Beware of standards ... 
7. Find Agile Business Metaphors
Practice 1: Be Part of the Development Team

- “Those developers just refuse to do as I tell them! @& #$ We should just fire them and get new ones!”
- Delivering Architecture documents is not good enough
- Spend substantial time within the development team
- If you don’t have enough time to spend with all teams, choose those that are instrumental in reaching your architecture goals
- You adapt to the teams way of working, not the other way around
- Be passionate!

This is your problem. Informing, motivating and influencing the team is your job!
Practice 2: Never Slow Down the Project

- You can change priorities, change requirements, do anything but slow down the project.
- There is no such thing as projects that go too fast – only that architects go too slow.
- Your Architecture probably isn’t finished when the teams start working with it.
  - That is OK ...
  - It might never get finished ... that is OK too.
- Balance architectural requirements with functional requirements.
Practice 3: Communicate Often and Early

- It is never to early to inform people
- Be transparent and open
- Expect team members to have the same worries and doubts that you have
  - It is just that you have more insight in concerns, issues and challenges outside the project
Practice 4: Travel Light, Focus on the Essence

 brewers: Of course, there is requirement management, design decisions, architecture models, issue management, architecture documents, ...

 brewers: ... but those are mostly for the project internals, not for communicating with stakeholders

 brewers: For the stakeholders: use only a storyboard and a roadmap

 brewers: Don’t mind drawing your architecture again and again ...

 brewers: It will get better every time and you will have the option to draw it specifically for your audience
BTW: Claim the Storyboard!
Practice 5: Architecture Needs Testing Too!

- Use proof-of-concepts or architectural ‘spikes’ to test key parts of your architecture
  - Even when these parts seem obvious
  - Test for scalability, performance, robustness, ...

- Use ATAM or ATAM-like techniques to validate your architecture
  - Often and early
Practice 6: Beware of standards

- While there is obvious value in standards ...
  - Reuse, maintainability, uniformity, ...

- There is the danger of suboptimal solutions and frustrating projects

- Standards imply that different projects, systems or different parts of your organization have the same needs

- This is OK in a static world, but in reality, in a world where change is perpetual, these needs differ and standards get in the way

- A standard is only valuable when it is implemented and proven, not when it is only written down

- Projects sometimes choose to ignore a standard. This might be the start of your next generation architecture
Practice 7: Find Agile Business Metaphors

- In every type of business, there is some area where agility is recognizable in the business itself
- Use this to explain and promote Agile Software Development
Agile Architecture on a Larger Scale: Landscape Photographs
The Agile Architect’s Toolkit

- Behold, the architect’s toolkit

- Final word of advice: don’t use this toolkit as a method, but
  - Use the toolkit with creativity!
  - Don’t apply all techniques (out of the box)!
Conclusions

- Agile Architecture is more demanding than traditional Architecture
  - It needs more discipline
  - Architects need to be committed

- It is about what you do when Architecture gets more difficult
  - ... because projects don’t comply to your Architecture
  - ... because there is too much work to be done
  - ... because your systems are too complex to really understand
References


