

Copyright (unless noted otherwise): Olaf Zimmermann, 2017. All rights reserved.

## Topic: Agile Architecting

### Context

Agile methods and practices introduce new roles (for instance, the Scrum master and the product owner in Scrum). Furthermore, they emphasize decentralization and self-responsibility (if individuals and teams). Hence, it is not always clear where the architect role goes in such settings (See Figure “Architects on Agile Projects”):

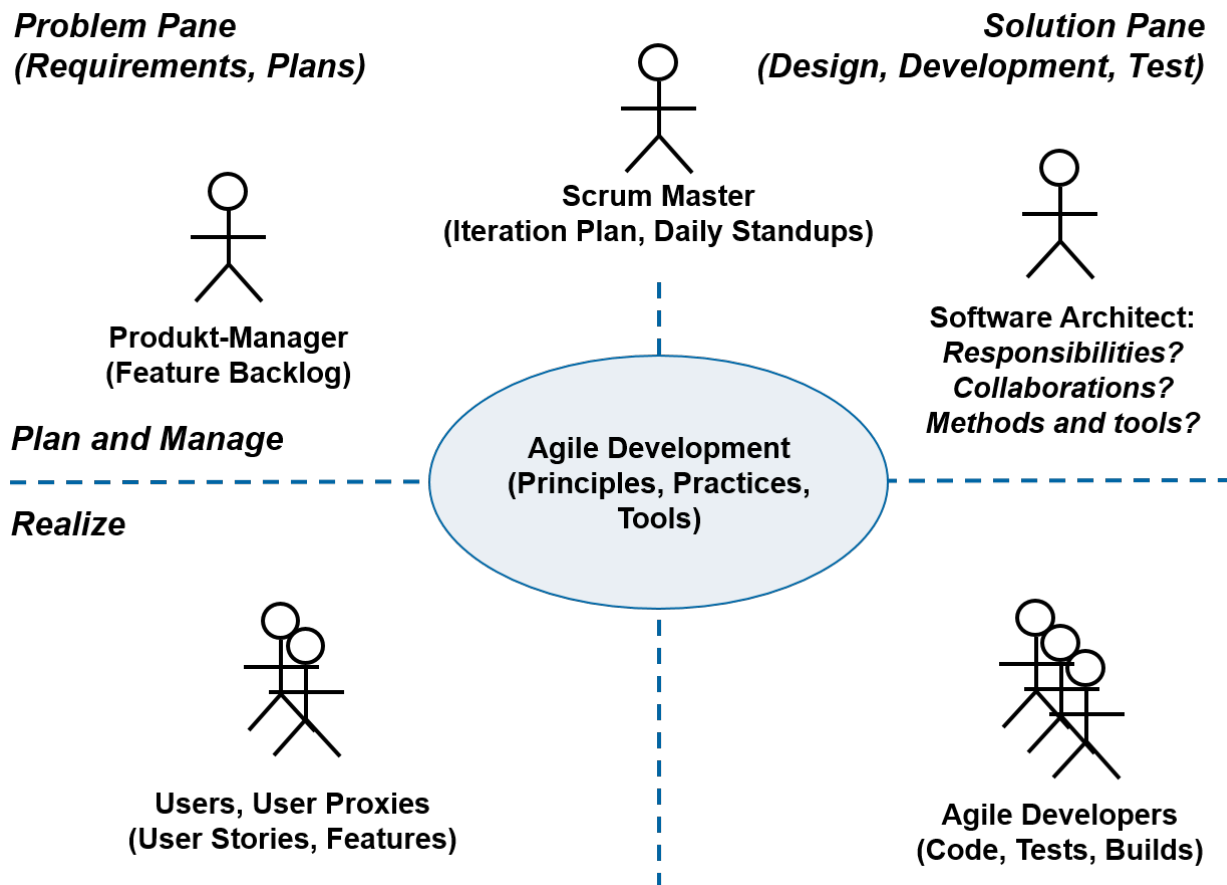


Figure 1: Architects on Agile Projects: Dilemma? Dying Breed?

A closer look unveils that the two topics and communities do not alienate, but actually cross-fertilize and depend on each other.

### Definition(s)

Agile architecting is “an agile way to define an architecture, using an iterative lifecycle, allowing the architectural design to tactically evolve gradually, as the problem and the constraints are better understood” according to P. Kruchten’s blog post on Agile Architecture and agile architecting<sup>1</sup>.

Agile architecting should comply with values and principles such as being customer- and product-centric and avoiding waste; however, agile projects should resist the temptation to oversimplify the picture; “code first” does not mean “code only”, and valuing “working software over documentation” does not imply a “no documentation”

<sup>1</sup><https://philippe.kruchten.com/2013/12/11/agile-architecture/>

mantra. Design works, and that includes architecture design work (see fact sheet on “Software Architecture Fundamentals” for role and responsibilities), remains an important task whenever something is built that goes beyond “yet another fature (of the same kind)”. The more complex the problem domain and the larger the team, the more architecture work is required, both upfront and ongoing.

## Examples

An ECSA SAGRA 2016 keynote presentation<sup>2</sup> lists seven elements of agile architecting with examples to craft architectures and systems of longevity and endurance:

- Specify ASRs in a SMART way, e.g., in the form of quality stories.
- Make conscious decisions and provide rationale, e.g., in Y-statements.
- Model context, containers, components, classes (C4) – but not more.
- Apply an Architecturally Evident Coding Style (AECS).
- Architect the time dimension, e.g. in the form of event tables and roadmaps.
- Practice architectural refactoring, revisiting the team’s decisions along the way.
- Consider the SOA style and its microservices implementation approach.

The emerging BizDevOps<sup>3</sup> website collect related/supporting practices in its DevOps compendium.

## Application in Products and Projects

### Usage of Concept/Topic

The Scaled Agile Framework (SAFe), at present standing at Version 4.5, emphasizes the notion of emergent architecture<sup>4</sup> but also has the notion of an Architectural Runway<sup>5</sup>.

S. Ambler lists 17 strategies for sclaing agile under the term Agile Architecture<sup>6</sup>.

Another architecture design method that has (or at least claims to have) an agile mindset is Risk- and Cost-Driven Architecting (RCDA). While public presentations<sup>7</sup> and other publications on it exist (up to a PhD thesis), it also is a commercial training offering by CGI, a globally operating consulting firm.

### Tips and Tricks

Letting the architecture of the system under construction emerge silently during refactoring is not advisable on most projects. If you think that you do not need any architecture design because you apply agile practices and t please read through some of the following references:

- K. Schwaber’s original Scrum paper<sup>8</sup> from OOPSLA ’97 (hint: have a look at pages 10, 12, and 13).
- ThoughtWorks presentation Agilists and Architects: Allies not Adversaries<sup>9</sup> (R. Parsons, M. Fowler).
- IEEE Software, Special Issue on Agility and Architecture<sup>10</sup> (March/April 2010).

Additional advice is (source: ZIO):

---

<sup>2</sup><https://sagra2016.files.wordpress.com/2016/10/zio-towardsopenleanarchitectureframework-sagranov2016v10p.pdf>

<sup>3</sup><http://bizdevops.uk/terminology/>

<sup>4</sup><http://www.scaledagileframework.com/agile-architecture/>

<sup>5</sup><http://www.scaledagileframework.com/architectural-runway/>

<sup>6</sup><http://agilemodeling.com/essays/agileArchitecture.htm>

<sup>7</sup><http://www.sei.cmu.edu/library/assets/presentations/poort-saturn2013.pdf>

<sup>8</sup><http://www.jeffsutherland.org/oopsla/schwapub.pdf>

<sup>9</sup><http://www.infoq.com/presentations/agilists-and-architects>

<sup>10</sup><http://www.computer.org/portal/web/computingnow/archive/april2010>

- Don't follow any hype without thinking just because a topic is popular; have the courage to challenge hidden assumptions and fit for your project.
- Translate agile folklore into terms that you are familiar with.
- Conduct a *sprint zero* (or *architectural spike* or *pregame* including high-level design work) and maintain a decision backlog in addition to a product backlog (to pick up some of the agile practice terminology).
- Stay focused in your modeling and documentation efforts (but do *not* go down to zero specification/documentation); see IFS web page Method Selection and Tailoring Guide<sup>11</sup> for related advice.
- Assess the architectural significance of each backlog item (both functional and technical ones) at the beginning of each sprint/iteration; the importance and urgency of an item is defined by both business value and technical risk (see lesson 1 and supporting exercise for significance criteria).
- Think about the most (not necessarily the last) responsible moment for your architectural decisions; capture them in a lightweight form, for instance a Y-Statement or an Architectural Decision Record (ADR) following the M. Nygard template (more on this later in the lecture).
- Make and revisit architectural decisions at the start and end of each iteration. Follow a “good enough” approach to architectural decision making and documentation (later in the lecture, we will investigate different decision capturing templates).

Do not follow these recommendations blindly; apply common sense and take your organization's culture and project context into account.

**Reference:** Inspired by Aligning Architecture Knowledge Management with Scrum, V.-P. Eloranta and K. Koskimies, Proc. IEEE/IFIP WICSA SHARK 2012, ACM Digital Library, 2012.

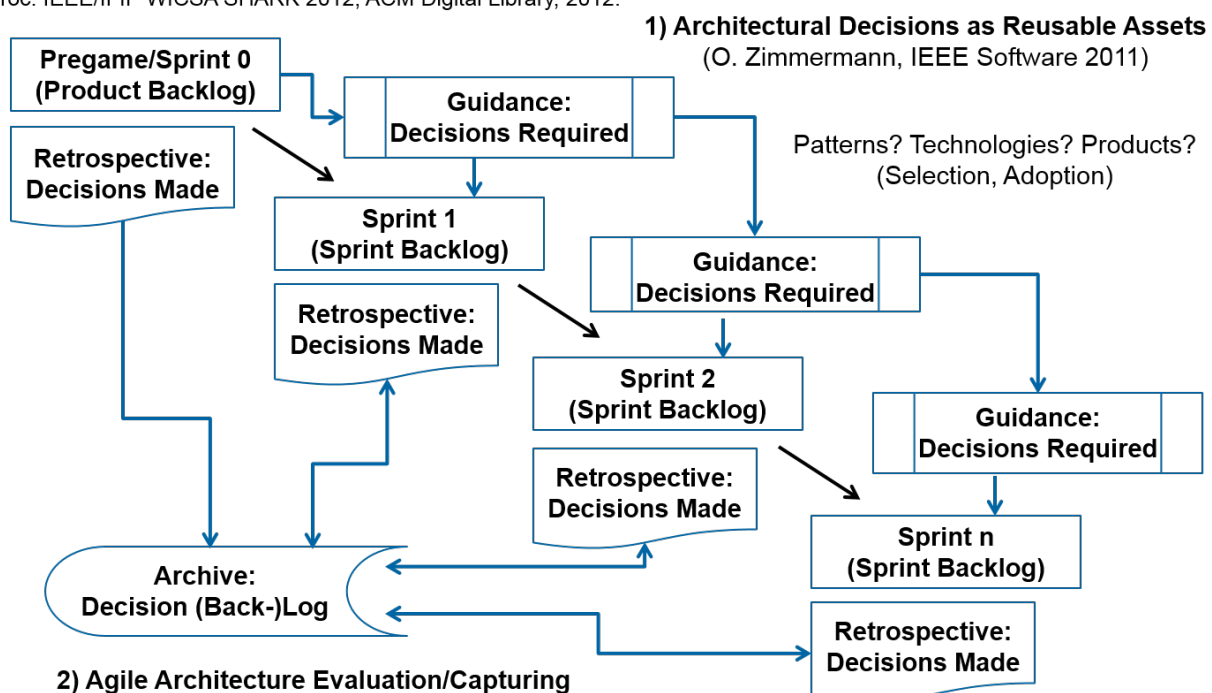


Figure 2: Architectural Decisions on Agile Projects

## Related Topics and Concepts

- SMART NFRs (sibling topic in lesson 2)
- The C4 Model, a simplified viewpoint model and diagramming approach (lessons 3 and 4)
- Lightweight Architectural Decision (AD) capturing (through lecture, lesson 12)

<sup>11</sup><https://www.ifs.hsr.ch/index.php?id=13195&L=4>

## More Information

See how the Agile Alliance documents agile practices such as simple simple design<sup>12</sup> and review the agile glossary<sup>13</sup>.

- B. Meyer's, "Agile! The Good, the Hype and the Ugly" reviews popular agile methods and distills and assesses their essence (Meyer (2014)).
- A video conversation between G. Fairbanks and M. Keeling discusses what agile practitioners need to know about architecture<sup>14</sup>.
- The Agile Architecting Library<sup>15</sup> at the Software Engineering Institute (SEI).
- The book "Vorgehensmuster für Software-Architektur" by S. Toth presents architecture in the context of agile in the form of process patterns (in German).
- The guest editorial of the IEEE Software special theme issue "The Role of the Software Architect in the Digital Age"<sup>16</sup> also discusses agile and architecture. This is a related presentation<sup>17</sup> by one of the authors.

More links are available at the bottom of the IFS website Architectural Knowledge Hubs<sup>18</sup>.

## References

Meyer, Bertrand. 2014. *Agile!: The Good, the Hype and the Ugly*. Springer Publishing Company, Incorporated.

---

<sup>12</sup><https://www.agilealliance.org/glossary/simple-design/>

<sup>13</sup><https://www.agilealliance.org/agile101/agile-glossary/>

<sup>14</sup><https://www.youtube.com/watch?v=uF9dIMlyy44>

<sup>15</sup>[http://www.sei.cmu.edu/architecture/research/agile-architecting/agile\\_architecting\\_library.cfm](http://www.sei.cmu.edu/architecture/research/agile-architecting/agile_architecting_library.cfm)

<sup>16</sup><http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7725214>

<sup>17</sup>[https://malibabar.files.wordpress.com/2010/11/turkish\\_sa\\_conf\\_keynote\\_041110-ppt.pdf](https://malibabar.files.wordpress.com/2010/11/turkish_sa_conf_keynote_041110-ppt.pdf)

<sup>18</sup><https://www.ifs.hsr.ch/index.php?id=13193&L=4>