

## Lean and Agile Architecture Tips

### Quotes

Posted by:

Posted on : 2011/12/7 8:20:38

The Agile Manifesto and Lean both give us principles. If we translate those principles to concrete practices and techniques for software architecture, we find the following pieces of advice:

1. Focus on the essence of the system form (what the system is) without being unduly influenced by the functionality that the system provides (what the system does).
2. Focus on the form of what the system is, partition it so that each part can be managed as autonomously as possible.
3. The dominant consideration in supporting team autonomy is how locally common changes are handled.
4. Let the human considerations drive the partitioning, with software engineering concerns secondary.
5. Be attentive to domain partitioning. In particular, don't split a domain across geographic locations or across architectural units.
6. Be attentive to the opportunity to use product lines and use this insight to bolster support for domain partitioning where possible.
7. Allow the module partitioning to follow domain knowledge. Think of domain knowledge as a timeless compression of the mental models of end users and other stakeholders—mental models whose patterns are tacitly driven by commonality and variation.
8. In the absence of domain knowledge, allow the module partitioning to follow the end user cognitive model of the domain. For every user "notion" create a corresponding architectural notion.
9. Capture your initial architecture at the most compressed level of expression that covers the scope of your business. Avoid abstraction or the discarding of any priceless information that may give insight on the form.
10. In fact, the initial partitioning is to create decoupled businesses whose concerns can be separated from each other.
11. Most domains in simple, interactive applications lead to modules that are implemented using object-based or object-oriented programming. This is particularly true for entities that the program presents on an interactive interface for direct manipulation by users.
12. The object structures in the what-the-system-is part of the architecture will become part of the Model in a Model-View-Controller(-User) architecture.
13. Provide an API to the code generated from a DSL for use by the rest of the architecture.
14. If you have a trustworthy pattern language for your domain, use it instead of domain analysis or end user input.
15. Leverage standards when dealing with third-party software.
16. Factor changes to third-party software into local parameters of variation or modules that are loosely coupled to the foreign software.
17. The essence of "Lean" in Lean architecture is to take careful, well-considered analysis and distill it into APIs written in everyday programming languages.

Source: <http://www.leansoftwarearchitecture.com/home/lean-and-agile-architecture-techniques>