The Role of the Architect in Agile

Dror Helper | @dhelper | http://helpercode.com

Practical Agile

About.ME

Consultant and Software Architect

Developing software since 2002
Clean Coder & Test Driven Developer

Pluralsight author
https://www.pluralsight.com/authors/dror-helper

B: http://helpercode.com
T: @dhelper

“Architects make high-level design choices much more often than low-level choices. In addition, the architect may sometimes dictate technical standards, including coding standards, tools, or platforms.”

Wikipedia: Software architect

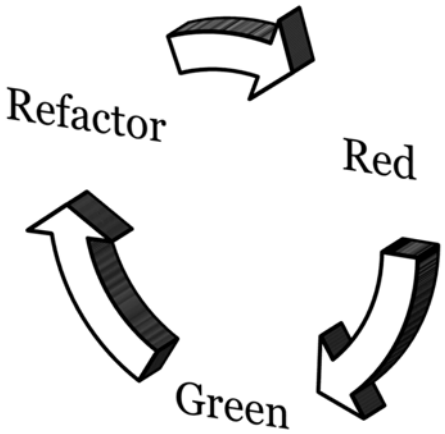
The Agile Manifesto

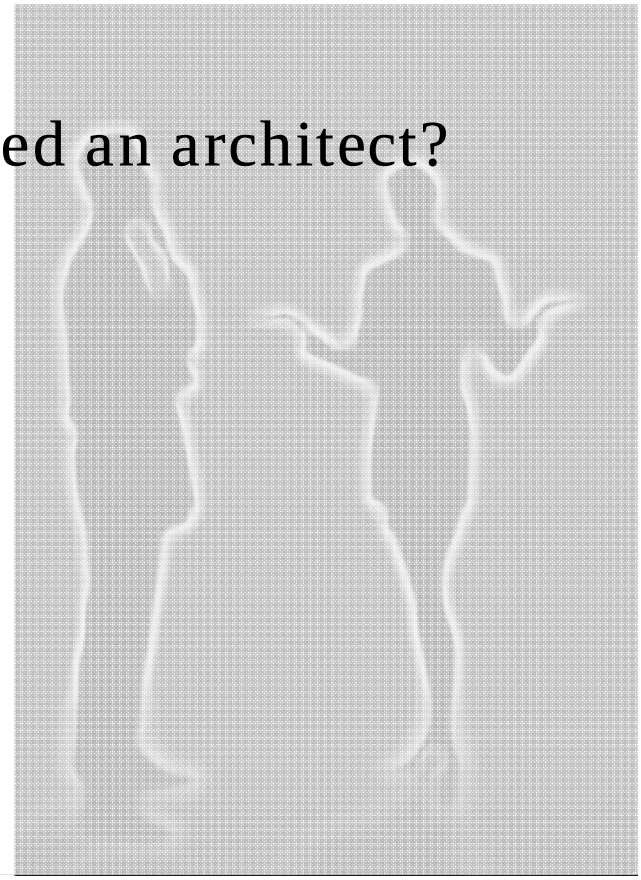
Individuals and interactions	Over	processes and tools
Working software		comprehensive documentation
Customer collaboration		contract negotiation
Responding to change		following a plan

Extreme Programming (XP) - Do we still need an architect?

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour week
- On-site Customer
- Coding Standard

In (A)TDD - do we still need an architect?





Gather functional requirements

Decide on quality attributes

Set constraints

Create principles

Architecture is needed

“The point is that you now have a framework and some boundaries to work within, which provide some often needed vision and guidance for the team.”

Simon Brown

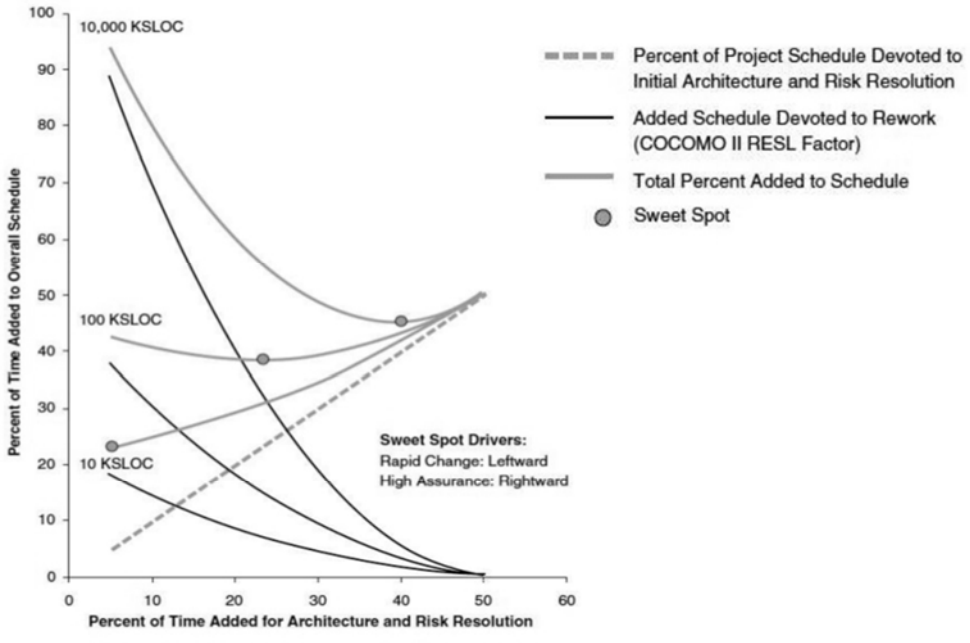


http://www.codingthearchitecture.com/2014/02/19/software_architecture_provides_boundaries_for_tdd_bdd_ddd_rdd_and_clean_code.html

Agile Architecture



How Much Architecture?



© Len Bass, Paul Clements, Rick Kazman, distributed under Creative Commons Attribution License

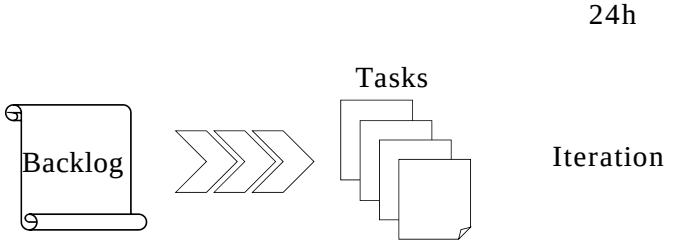
Garden shed
or
Skyscraper?

“While we must acknowledge emergence in design and system development, a little planning can avoid much waste.”

James O. Coplien, Lean Architecture

Agile Architecture

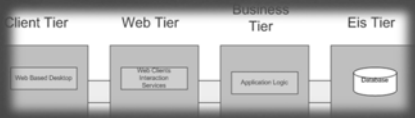
- Facilitate architectural decisions
- Explore new technologies
- Evolve architecture based on learning
- JIT Modeling

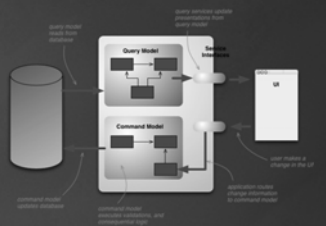


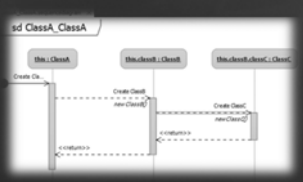
What About Documentation?

Documentation and Diagrams





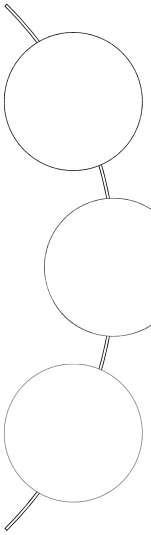




IEEE software life cycle	
SQA	Software quality assurance IEEE 730
SCM	Software configuration management IEEE 828
STD	Software test documentation IEEE 829
SRS	Software requirements specification IEEE 830
V&V	Software verification and validation IEEE 1012
SDD	Software design description IEEE 1016
SDD	Software project management IEEE 1058
SUD	Software user documentation IEEE 1063



Agile Documentation → YAGNI!



Only document what you need

- Consider the reader

Provide guidance using code

- Describe using documentation what you can't get from the code

Document what you must due to regulations, contracts etc.

- But make sure you have to

Architecture by example

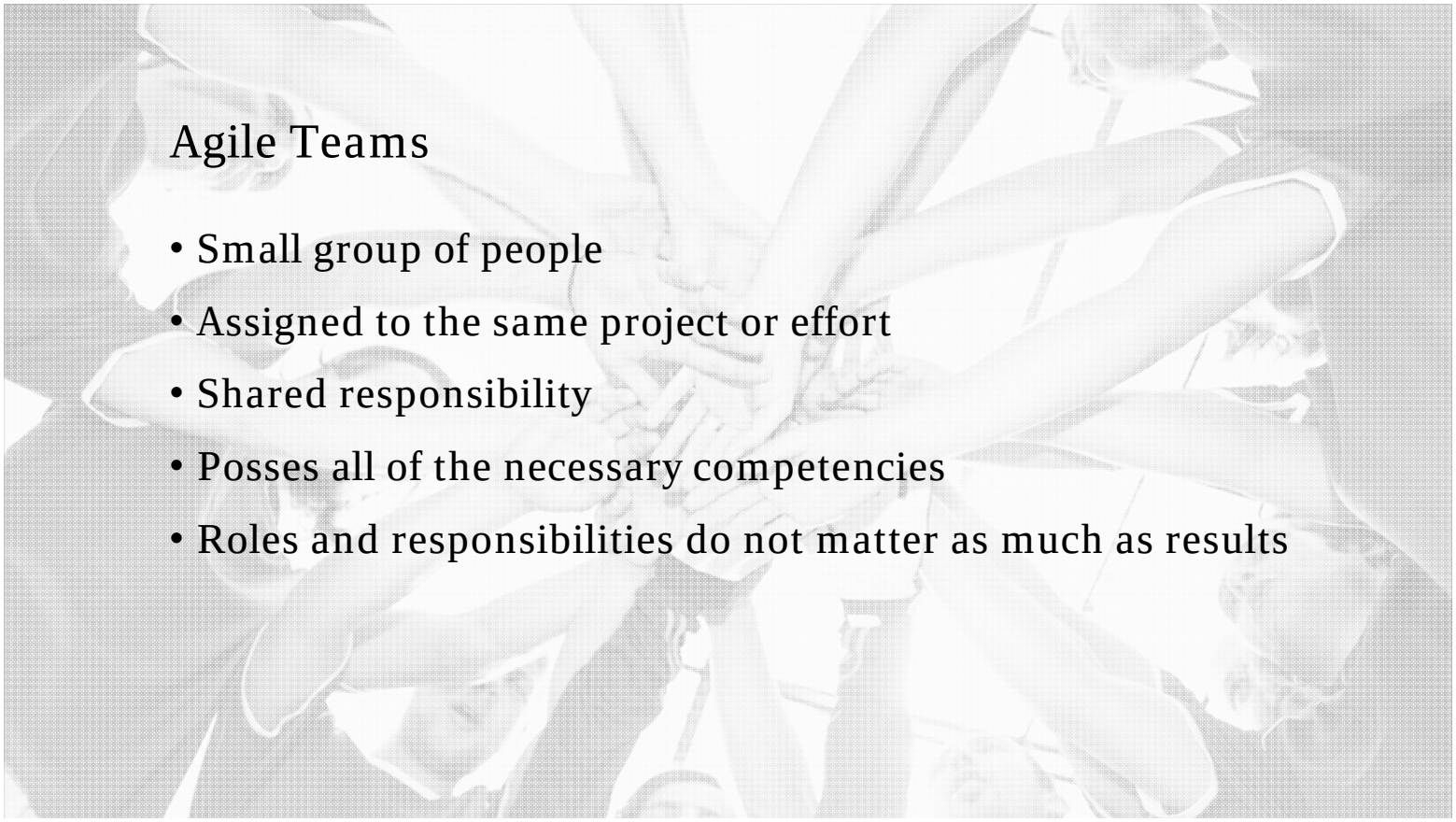
Create vertical slices

Show use of technology

Provide coding standards

Only develop enough code to solve a problem or provide guidance

The Architect Place in the Agile Team

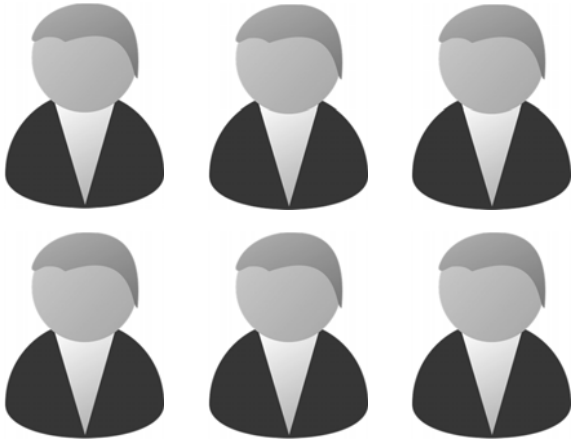


Agile Teams

- Small group of people
- Assigned to the same project or effort
- Shared responsibility
- Posses all of the necessary competencies
- Roles and responsibilities do not matter as much as results

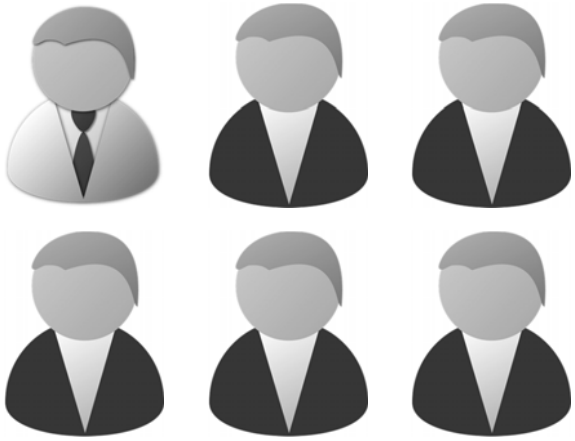
Option 1 - Everybody an architect

- Small company
- Experienced developers
- Mature developers
- Ability to self govern



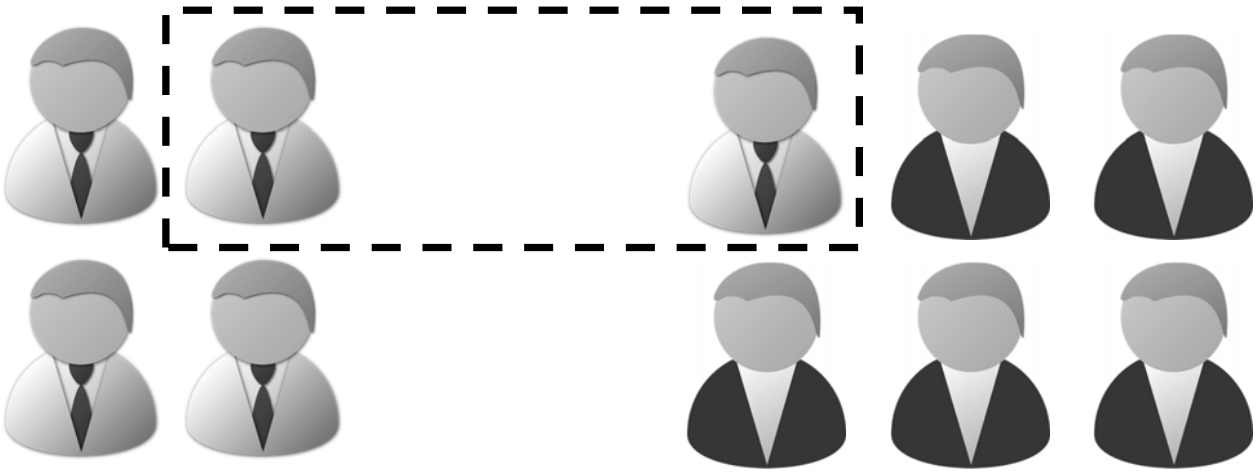
Option 2 – Architecture Owner

- Small company
- Clear architecture
- Different skill levels/experience
- Align development within the team



Option 3 – Architecture Owner Team

- Centralized standards
- Align architecture for every team



Top 10 list of things for a solution architect to be successful in an agile team

Tom Hollander

- "Just enough" upfront design
- Start with a vertical slice
- Just-in-time design each iteration
- Trust your team… but be there for them
- Write code!
- Be involved in everything
- Drive a culture of quality
- Know when changes are required
- Shield the team from external randomization
- Write docs … but only if someone needs to read them