

Agile Software Architecture: Why Your Project Needs It

Business meets Implementation

Chris Miyachi

SDM Entering Class 2000

Presentation Outline

- What is Software Architecture
- Architecture within the Software Lifecycle
- Software Architecture and Lean Software Development
- Software Architecture Decisions
- Documenting Software Architectures

Software Architecture Definition

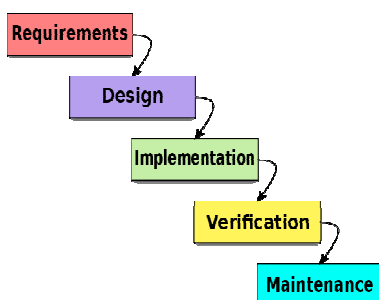
- Software architecture is the **fundamental organization** of a system, embodied in its **components**, their **relationships** to each other and the environment, and the principles governing its design and evolution.
(IEEE 1471-2000)

Software Architecture Views

- Functional view
- Code view
- Structural view
- Runtime view
- Physical/ installation view
- User Interaction
- Data view

Where is Software Architecture Used During the Software Lifecycle?

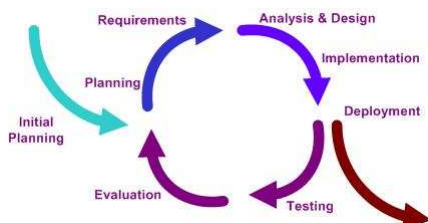
- It depends on the lifecycle
- Waterfall Lifecycle



In this methodology, all requirement analysis, design, and architecture are done up front. This is often called BDUF (Big Documentation Up Front)

Where is Software Architecture Used During the Software Lifecycle?

- Iterative/Agile



In this methodology, requirements analysis, design, and architecture are done each iteration.

A final system is complete after many iterations. Requirements, design, and architecture are added each iteration.

This methodology is typically part of lean or agile software methodologies, like SCRUM.

What is Agile Development / Lean Software Development?

- In 2001, a group of developers met to develop the Agile Manifesto:
 - Individuals and interactions over processes and tools
 - Working software over comprehensive documentation
 - Customer collaboration over contract negotiation
 - Responding to change over following a plan

How Does Architecture Work with Lean Software Development?

- Some architectural decisions must be made up front, even with software.
 - For example, decisions about hardware, underlying support tools, etc.
 - Major architectural framework decisions
- Most need to be made iteratively.
 - The feedback and knowledge gained from the previous iteration will drive the current architectural decisions.

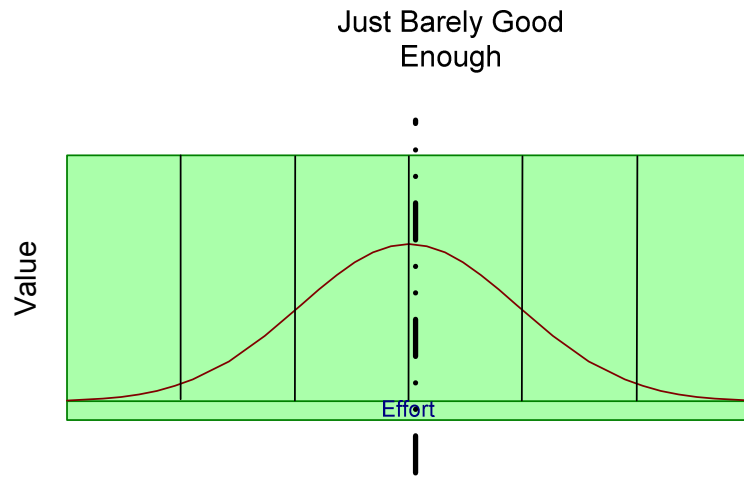
Waterfall Versus Agile

- Pure Agile proponents say nothing needs to be made up front
- Pure Waterfall proponents say all architectural decisions need to be made up front
- Most agree that there is a combination of upfront decisions and iterative decisions that will contribute to the success of the software architecture.

Can Architecture Be Agile?

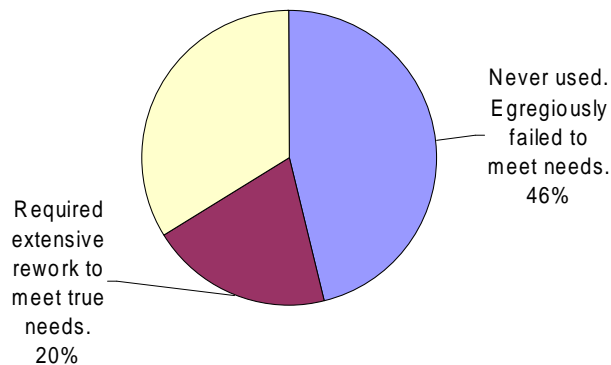
- Architecture can be done with Agile and Lean Processes
 - Create architecture in iterations
 - Deliver working systems with each iteration
 - Create a cross functional team to resolve architectural issues

Architect Just Enough For What You Need Now



Requirements and Failure

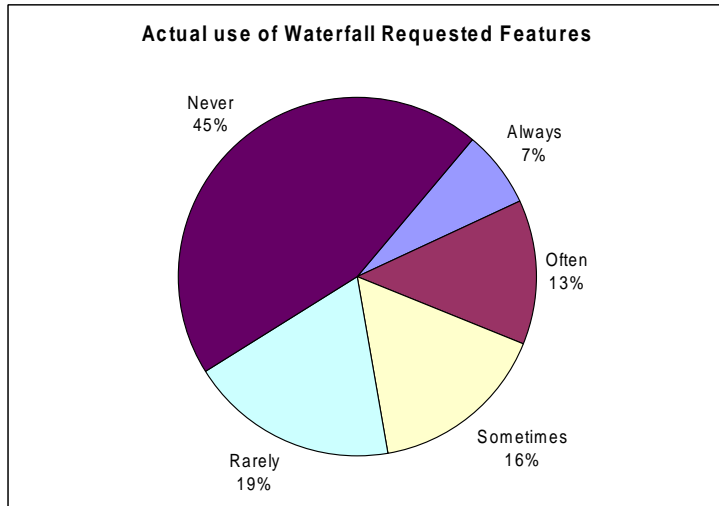
\$37B worth of DoD projects using 2167A



Jarzombek Study.

Failure attributed to use of waterfall.

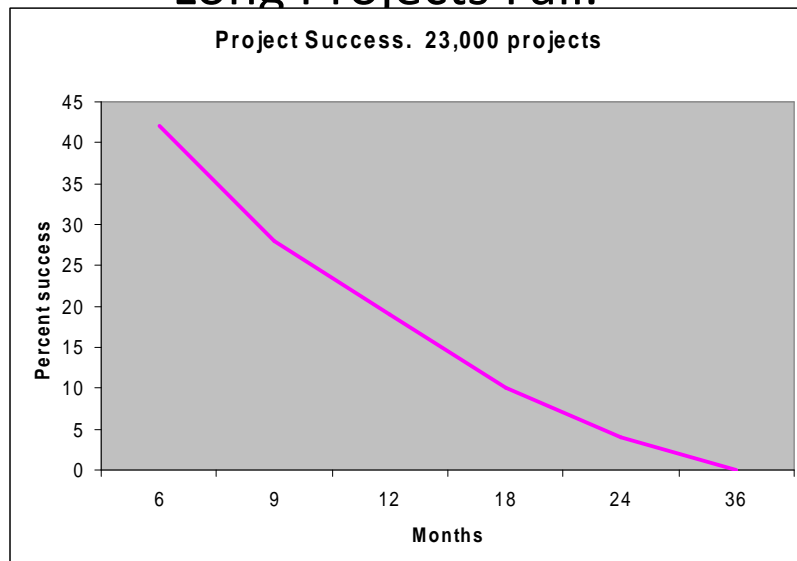
Over specification.



2/12/2011

13

Long Projects Fail.

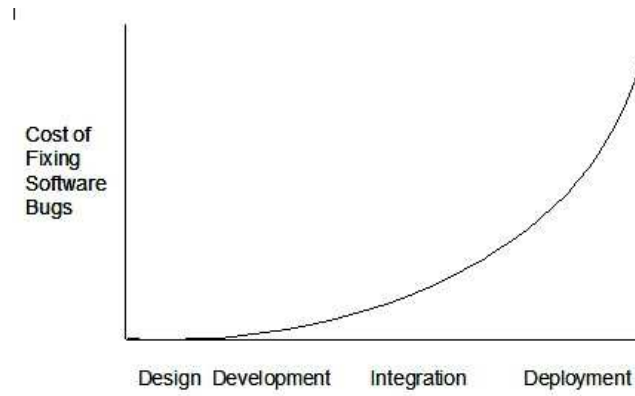


2/12/2011

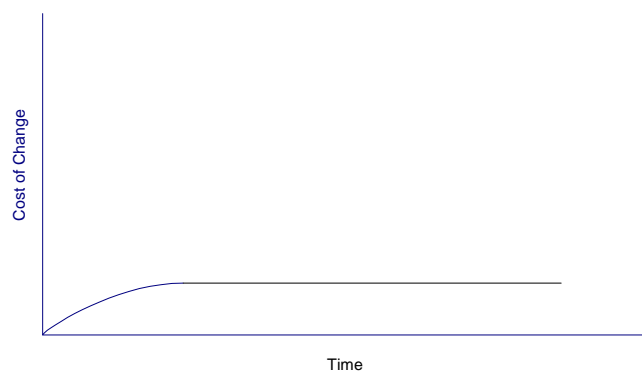
Source: www.objectmentor.com

14

Cost of Fixing Software Bugs



What Agile Claims

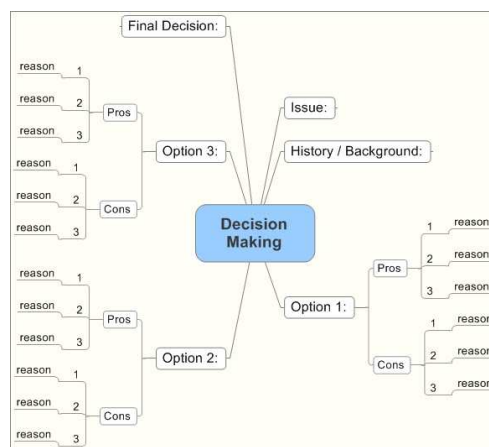


How are Architectural Decisions Made?

- While major decisions are made up front, lower level architecture and design are often made with daily decisions.
- Grady Booch calls this Accidental Architecture (<http://www.informit.com/articles/article.aspx?p=471929>).
- So how to make the best decision:
 - Get stakeholders together
 - Decide on requirements and their importance
 - Brainstorm solutions
 - Rate solutions against requirements.
 - Make the decision before everyone leaves (if possible)

Key Tools – Mind Maps

- I first learned about Mind Maps in the SDM program



Key Tools – Decision Matrices

Reference: *Getting Started in Project Management, by Karen Tate*

Steps:

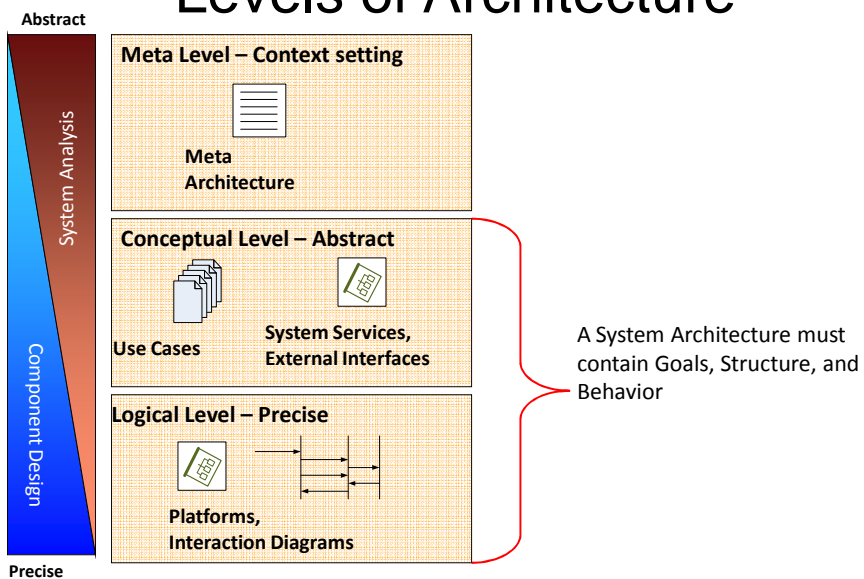
1. Define the decision goal.
2. Identify and list the decision options. (One option is to brainstorm these in a team meeting.)
3. Identify and list the decision criteria. Try to limit the number of criteria to about six. It is important that the criteria be clearly defined and that they help differentiate the decision options.
4. Establish a weight for each decision criteria.¹ The process works best when each criterion has a different weight. One approach is to pool and average the weights given by individual team members.
5. Rate each decision option against each decision criterion.² (Try to ignore the results of Step 4!) One approach is to pool and average the ratings given by individual team members.
6. Calculate the weighted value of the decision option.³
7. Calculate the decision score for each decision option.⁴ The decision option with the highest number is the solution that best satisfies the decision criteria.
8. Evaluate the results of Step 7 against the decision goal defined in Step 1. If the results make sense, document the decision reached; otherwise, review the decision criteria, their weights, and the decision ratings assigned, and try again. Do not change the numbers to affect the selection of a "favored" solution.

Decision Goal: Define the goal to be achieved by the decision. What is the problem? Why is a solution required?											
Decision Participants: X, Name, Y, Name		DECISION OPTIONS									
DECISION CRITERIA	Criteria Weight	A – Describe decision option		B – Describe decision option		C – Describe decision option		D – Describe decision option		E – Describe decision option	
		Rating	Weighted Value	Rating	Weighted Value	Rating	Weighted Value	Rating	Weighted Value	Rating	Weighted Value
1											
2											
3											
DECISION SCORE											

Decision: Identify and describe the decision reached.

- ¹ Each decision criteria has a weight, as follows: 1-3 = nice to have; 4-6 = highly desirable; 7-9 = essential (must have)
- ² Rating scale: 1 = the decision option does not satisfy the criterion; 2 = the decision option satisfies the criterion moderately well; 9 = the decision option satisfies the criterion extremely well.
- ³ Each decision has a weighted value equal to the criteria weight multiplied by the decision rating.
- ⁴ Each decision has a decision score equal to the summation of the weighted values given for each decision criteria.

Levels of Architecture



Documentation Software Architectures

- 5 ways to screw up:
 - Overload a view
 - Explain too much
 - Be over precise
 - Be inconsistent among views
 - Select the wrong view for your audience.
- Source ([Rebecca Wirfs-Brock](#) Documenting Software Architectures)

System Architecture Model

What is in it?

- System Architecture should contain *goals /requirements* artifacts, and *structure* and *behavior* artifacts based on those goals [\[2\]](#)
- Unified Modeling Language has been chosen by the team to create these artifacts:
 - Goals/Requirements: Use Cases
 - Structure: Package and Class Diagrams
 - Behavior: Sequence, State, and Collaboration Diagrams
- [\[2\] A Taxonomy of Decomposition Based on Structures, Behaviors, and Goals](#), Koopman, Design Theory and Methodology '95
 - Paper from System Architecture Course in SDM Program at MIT

Objectives of Agile Architecture

- Deliver working solutions
- Maximize stakeholder value
- Find solutions which meet the goals of all stakeholders
- Enable the next effort
- Manage change and complexity

<http://www.agilearchitect.org/agile/principles.htm>

Conclusion

- Agile Development is enormously effective in the delivery better quality software faster
- Architecture must fit into the Agile Development lifecycle, otherwise you will have an accidental architecture

About Me

- Principal Systems Engineer / Architect at Xerox Corporation
- MIT SDM graduate (entering class of 2000)
- Web site:
<http://home.comcast.net/~cmiyachi/default.html>
- Blog on Software Architecture:
<http://abstractsoftware.blogspot.com/>