

# **How Software Learns: What happens after software is shipped**

Chris Miyachi

SDM Entering Class 2000

# Presentation Outline

- Software Architecture versus Building Architecture
- Definitions of Architecture
- Process and Flow
- Building for Change
- Conclusion

# Models

- “All models are wrong, some are useful”
  - generally attributed to the statistician George Box
- “All *buildings* are predictions, and all *buildings* are wrong”
  - Stewart Brand, “*How Buildings Learn*” video  
(  
[http://video.google.com/videoplay?  
docid=2283224496826631552&emb=1&hl=en](http://video.google.com/videoplay?docid=2283224496826631552&emb=1&hl=en) )

# Building Architecture

- When studying building architecture, three things come to the surface:
  - Software designers can learn a lot from building designers
  - Software designers and building designers make the same mistakes
  - Building designers can learn a lot from software designers

# 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)

# System Architecture Model

## What is in it?

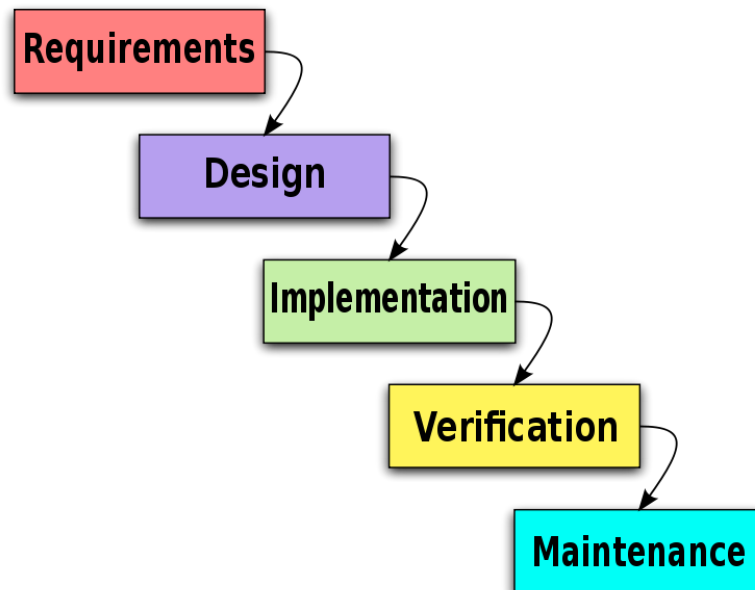
- System Architecture should contain *goals /requirements* artifacts, and *structure* and *behavior* artifacts based on those goals [\[2\]](#)
- Unified Modeling Language has been chosen by the team to create these artifacts:
  - Goals/Requirements: Use Cases
  - Structure: Package and Class Diagrams
  - 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

# Building Architecture

- In relation to buildings, architecture is defined as:
- *The planning, designing and constructing form, space and ambience that reflect functional, technical, social, environmental, and aesthetic considerations. It requires the creative manipulation and coordination of material, technology, light and shadow.*
- *Architecture also encompasses the pragmatic aspects of realizing buildings and structures, including scheduling, cost estimating and construction administration.*
- *As documentation produced by architects, typically drawings, plans and technical specifications, architecture defines the **structure and/or behavior** of a building or any other kind of system that is to be or has been constructed.*

# The Software Development Process

- Waterfall Lifecycle



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



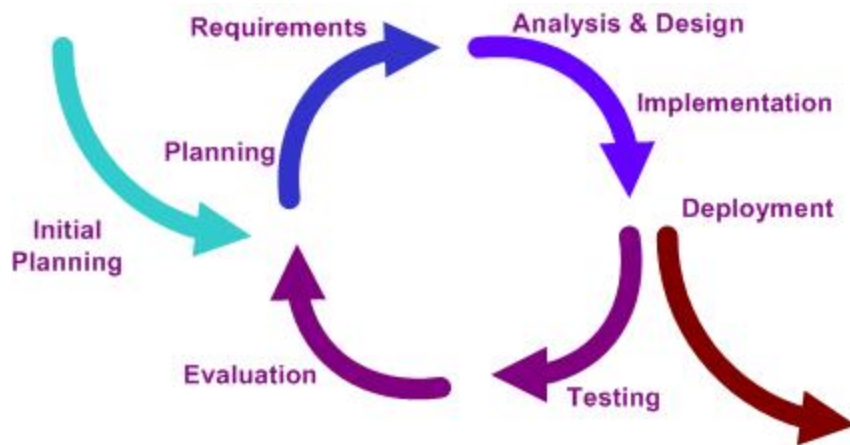
# The Software Development Process

- 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.

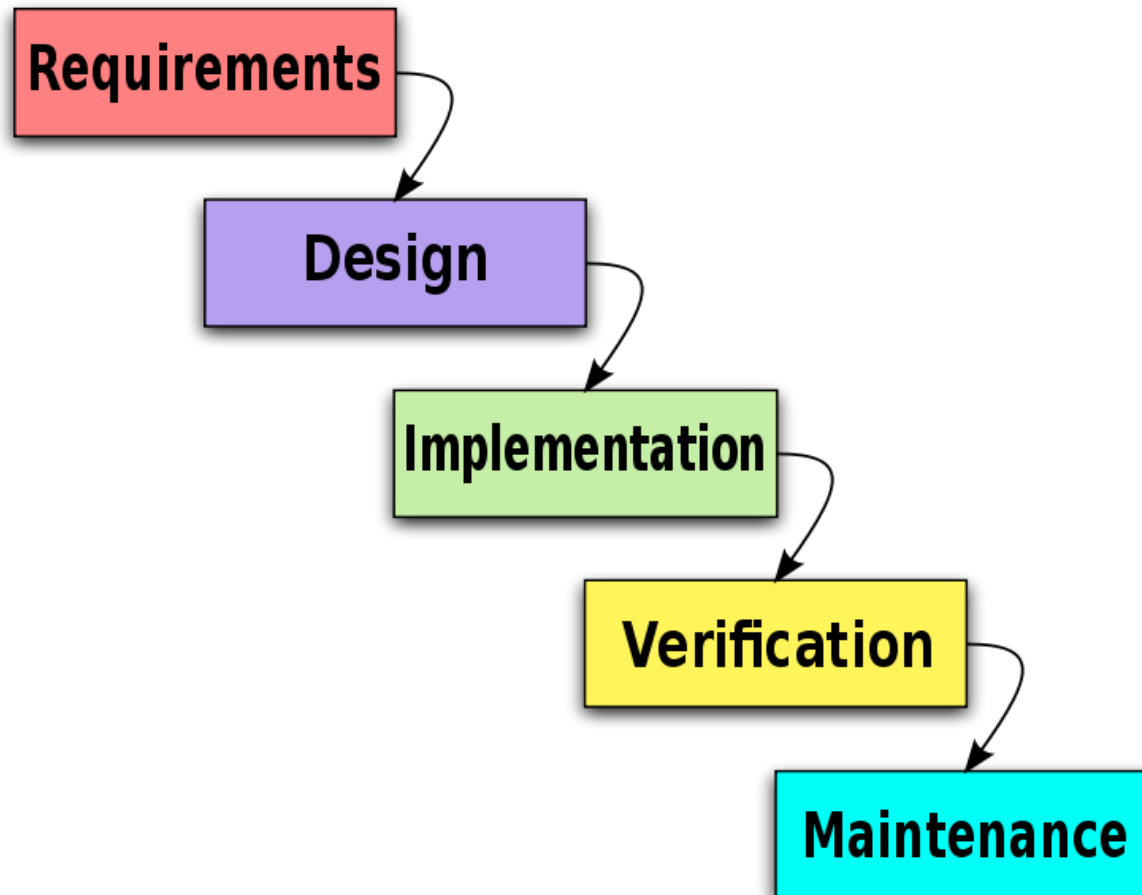
This methodology is typically part of lean or agile software methodologies, like SCRUM.



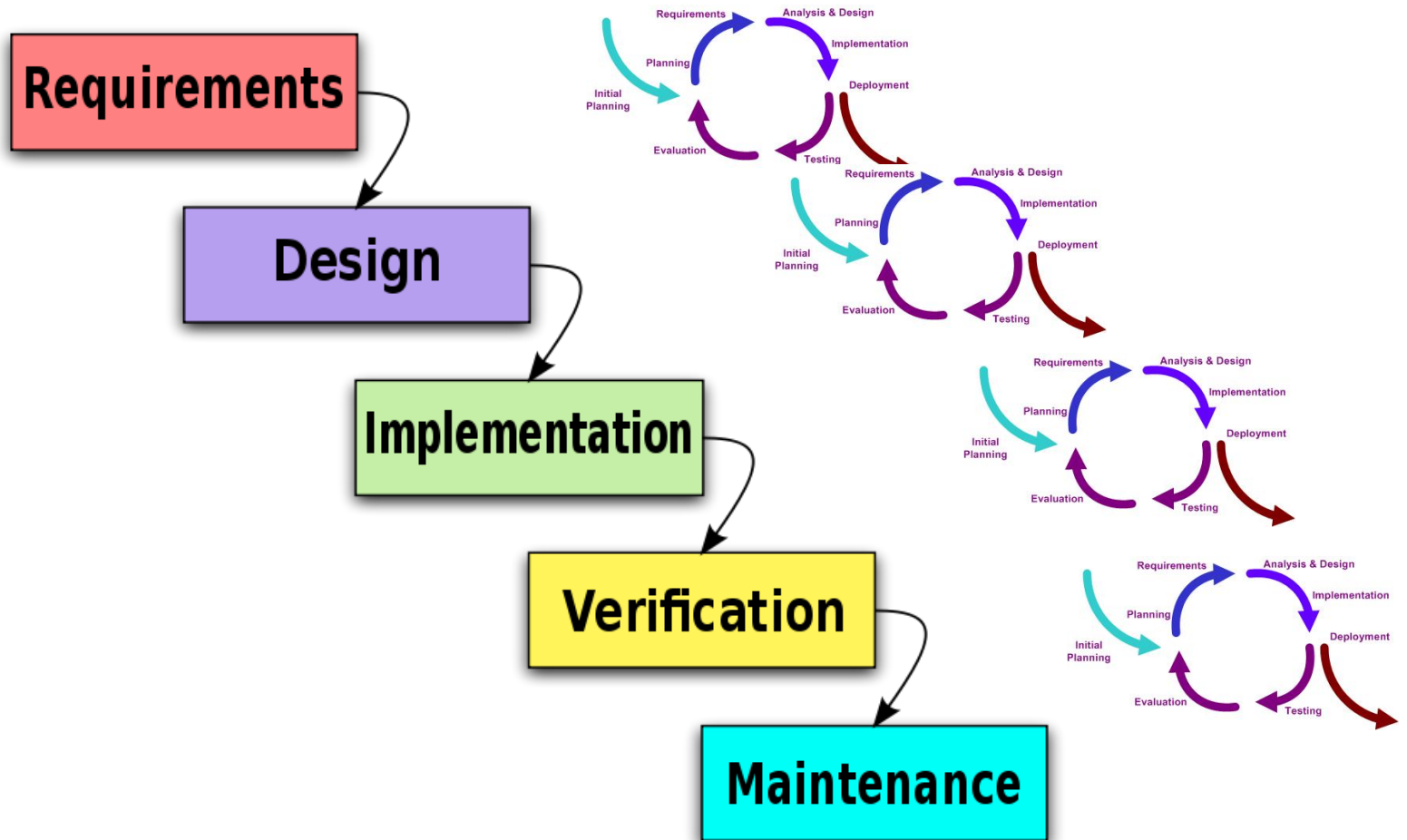
# What is Agile Development / Lean Software Development?

- 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

# The Building Architecture Process



# The New Software Process



# Waterfall Versus Agile

- Agile: no decisions need to be made up front
- Waterfall: 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.

# Flow

- “Flow, continual flow, continual change, continual transformation”
  - Rina Swentzel, Pueblo Indian architectural historian
- “Responding to change over following a plan - focus on quick responses to change and continuous development.”
  - The Agile Manifesto

# Example: The Media Lab



# Example: Stata Center at MIT



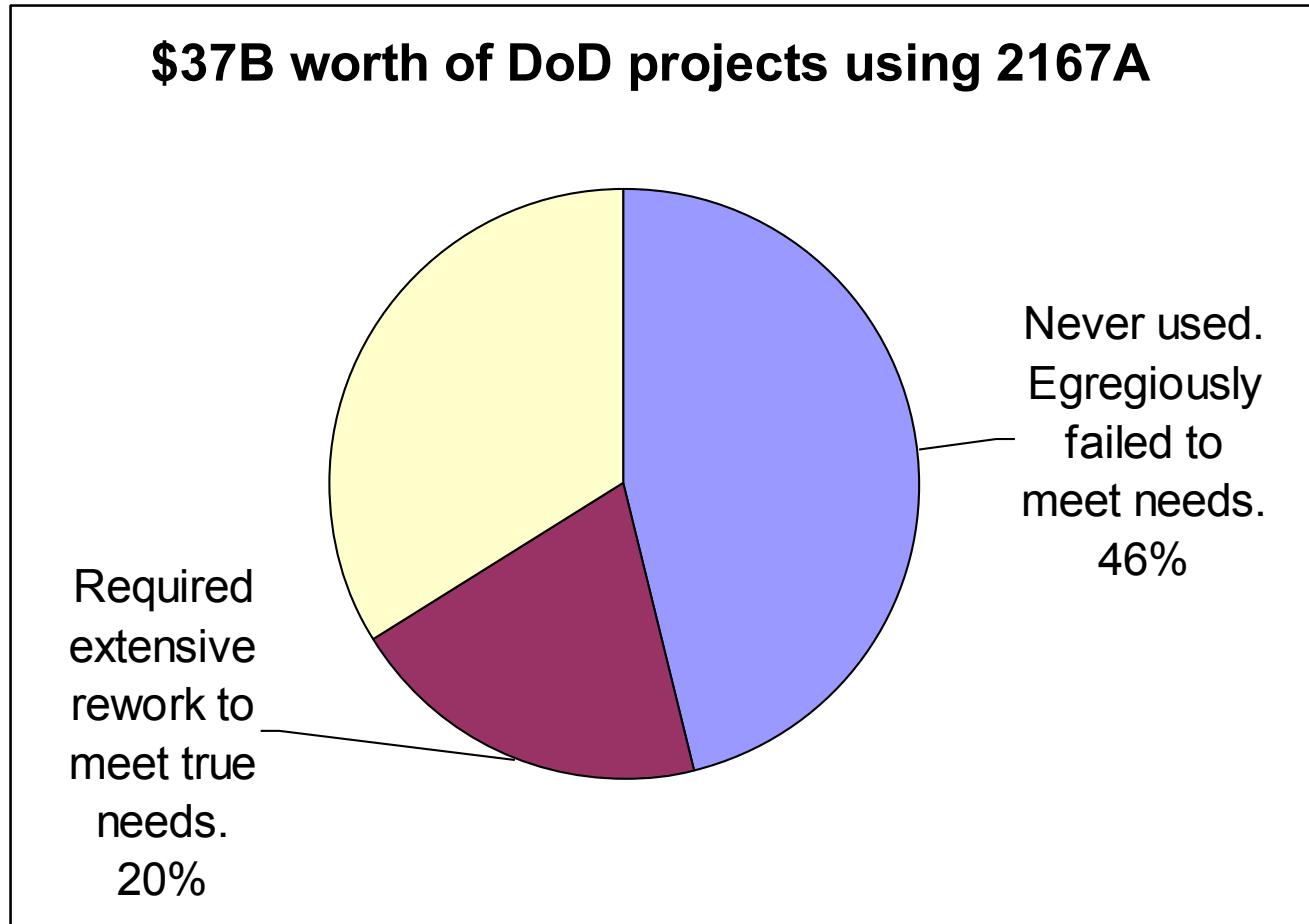


# Form Follows Function

- Form ever follows function
  - 1896, Louis Sullivan, a Chicago high rise designer
- Three of the principles of Agile development are focused on change
  - Customer satisfaction by rapid delivery of useful software
  - Welcome changing requirements, even late in development
  - Working software is delivered frequently (weeks rather than months)

# Requirements and Failure

**\$37B worth of DoD projects using 2167A**



Jarzombek Study.

Failure attributed to use of waterfall.

# Building For Change

- Agile and Future Maintenance
  - Often not considered
  - Off-shoring
- We shape our buildings, and afterwards our buildings shape us (Winston Churchill )
  - First we shape our buildings, then they shape us, then we shape then again, ad infinitum

# Tradeoff

- Waterfall is easier but the world isn't static
  - Change is always occurring
- Look at what areas rapidly change and design an architecture that can handle high change in that area of the system

# Façade

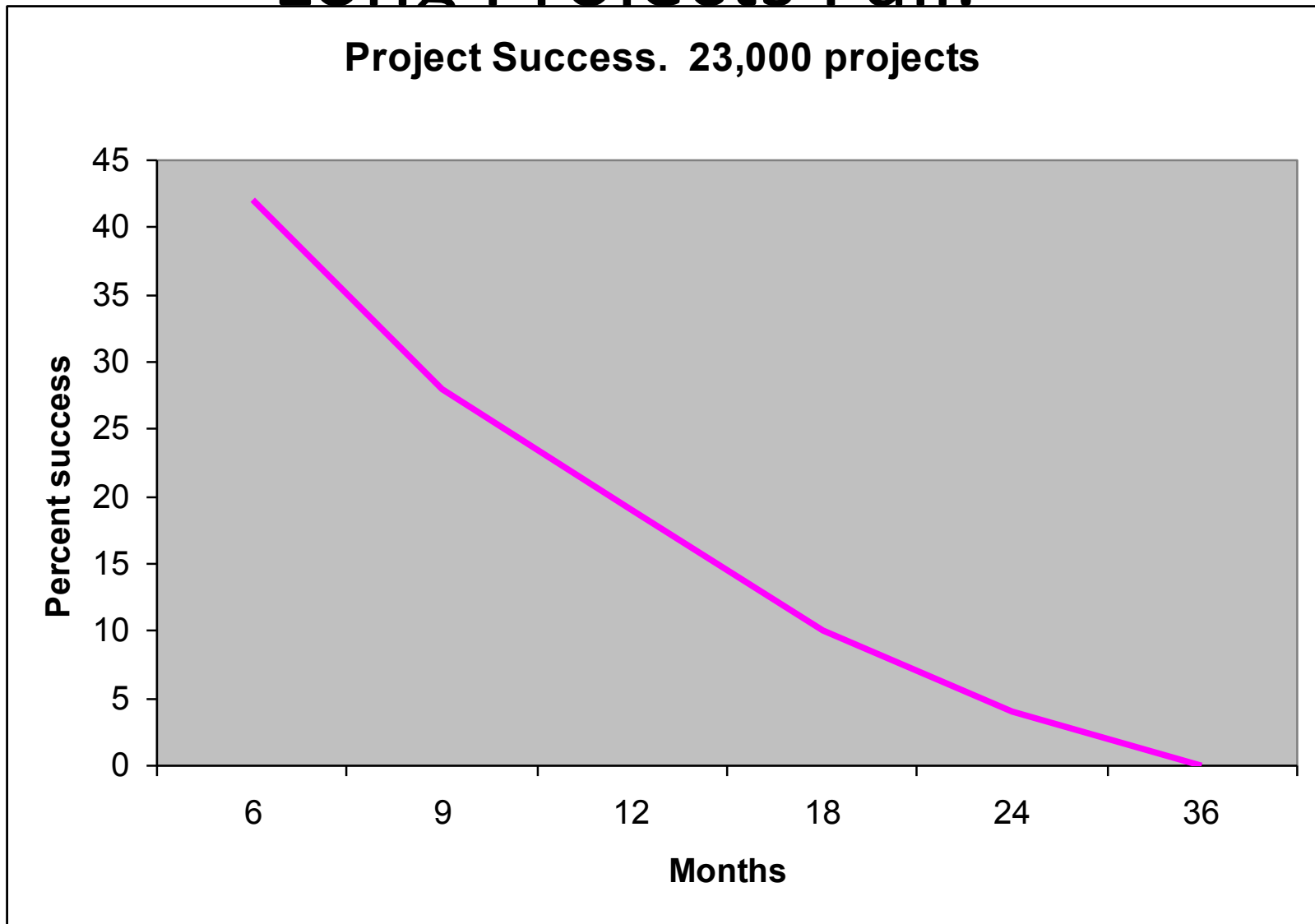


# Software Example: Interfaces

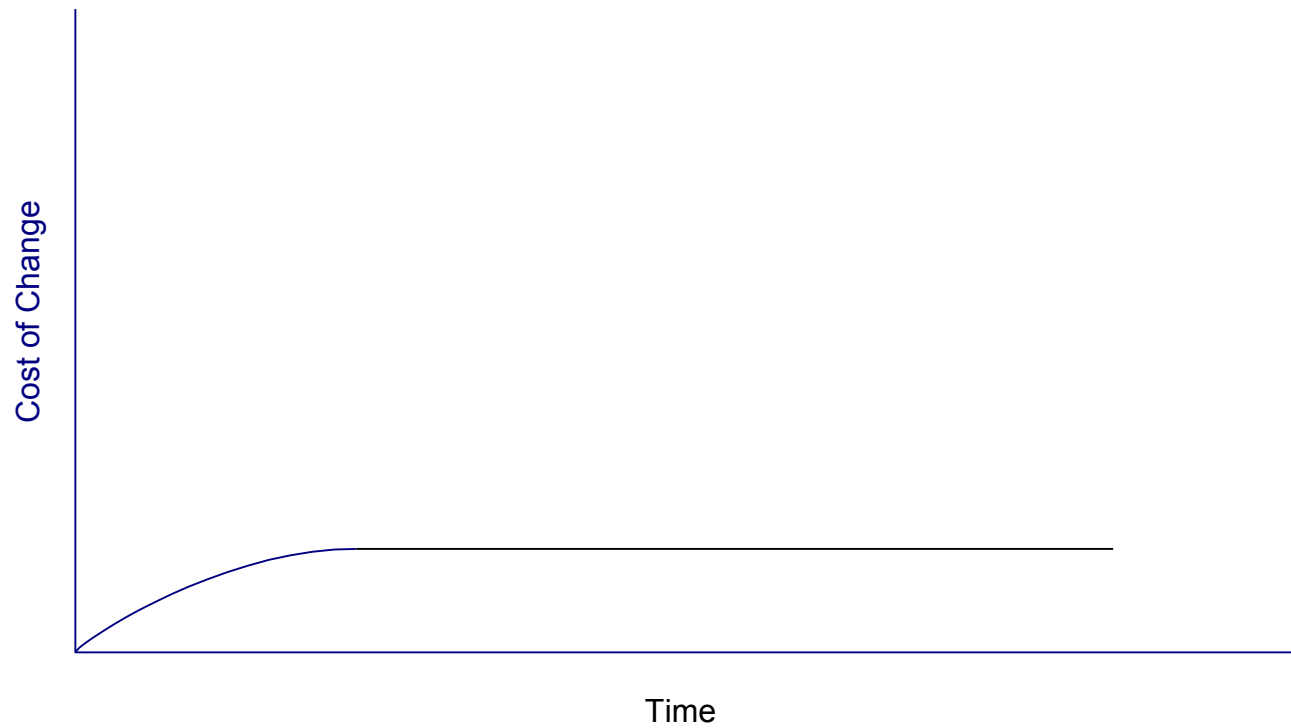
<b>Configure Digital Certificates</b>
+UpdateCertificate() +DeleteCertificate()

<b>Configure Security</b>
+UpdateCertificate() +DeleteCertificate() +ConfigureProtection()

# Long Projects Fail.

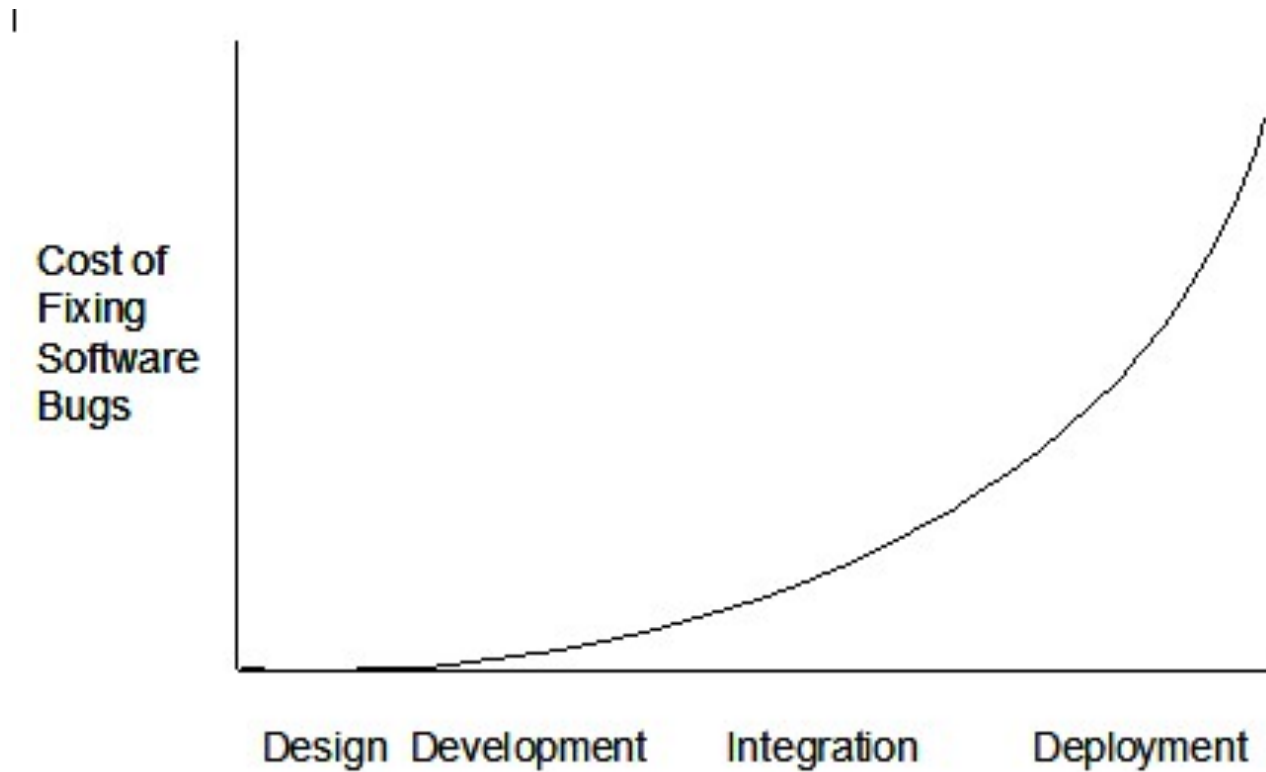


# What Agile Claims





# Cost of Fixing Software Bugs



# Conclusion

- Buildings live in time. In time we learn about how to use them.
- Software also evolves in time.
  - We need to experience our software and our buildings to know what we want to change.
- We need long term thinking and short term thinking
  - Seminars in long term thinking:  
<http://longnow.org/seminars/>

# 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/>