

# Incorporating Agile Methods in Large-Scale Systems

Trenton Hafterson

University of Minnesota, Morris

April 30, 2011

Why would a large-scale software development company want to be agile?

Agile methods aim to counter the tremendous costs with changes late in the development process of common traditional development methods.

# Outline

- 1 Introduction and Background
  - What is Software Development
  - Agile
  - Why Agile Methodologies
- 2 Challenges of Large-Scale Extreme Programming (XP)
- 3 Hybrid Methods
  - Soft-Structured Agile Framework
- 4 Phasing Methods
  - Agile Software Solution Framework
- 5 Conclusions



# Software Development

Software development is an organized collection of:

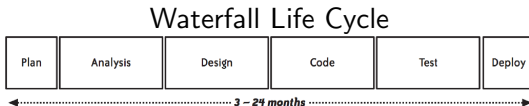
- Research
- Design
- Programming
- Maintenance



# Waterfall Method

Sequential design  
process:

- 1 Planning
- 2 Analysis
- 3 Design
- 4 Code
- 5 Test and  
Debug
- 6 Deploy



Originates from the manufacturing and construction industries



# Agile Development

- As the software development industry evolved from traditional methods
- 17 software developers came together to create Agile Manifesto



# Agile Manifesto's Values

## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.



# Agile Manifesto's Principles

The Agile Manifesto has 12 principles. I will focus on 4 principles.

## Agile Manifesto Principles

- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



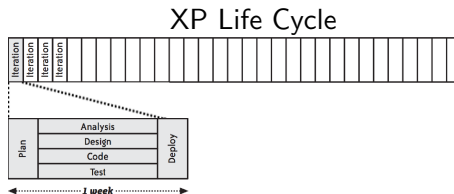
# Extreme Programming (XP) Method

Focus values on:

- Communication
- Simplicity
- Feedback

This is accomplished by:

- Team Communication
- Iterative Design
  - 1-2 weeks
- Customer Involvement
- A set of practices to bring this all together

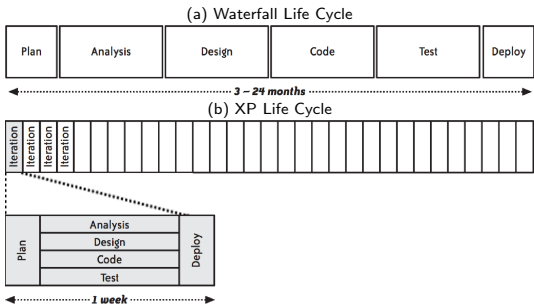




## Why Agile Methodologies?

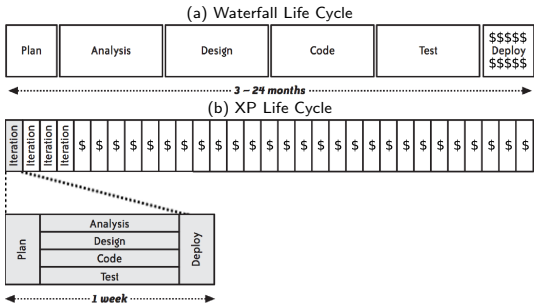
# Agile Benefits

- Ability to handle change
- Continuous customer involvement
- Iteration length



# Agile Benefits

- Ability to handle change
- Continuous customer involvement
- Iteration length



# Challenges of Large-Scale XP

Large-scale projects can be distributed over multiple teams, buildings, and countries.

- Spatial
- Temporal
- Cultural



# Spatial

- Critical part of XP is open communication
- With distributed projects, the development and customer interactions are potentially lost
  - XP practices
    - Deteriorate the effectiveness of the agile process
- A development team would have to rely on electronic communication.
  - Phone
  - Instant Messaging
  - Video or Audio conferencing systems

# Temporal

- Teams working in different time zones
- Asynchronous communication
  - Team moderator
  - Surrogate customer



# Cultural

- Cultural differences may not be directly related to distributed scenarios.
- But the lack of in-person communication and meetings can add to misunderstandings.
- Cultural differences can stem from:
  - Language barriers
  - Cultural references
  - Customs
- Again like in time differences, a moderator role may help to bridge these differences.

# Is XP the Only Option?

- Although XP may be applied to large-scale systems
- There are alternatives to switching to XP all at once:
  - Hybrid agile methods
  - Methods that phase into a agile approach



# Hybrid Methods

- A development team can consider a method that uses the best of both waterfall and XP-like practices
- XP practices that can be applied to large-scale may include:
  - Iterative life-cycle
  - Pair-programming
  - Customer involvement
- And some that may be less feasible:
  - Minimal documentation
  - Code refactoring

# Soft-Structured Agile Framework

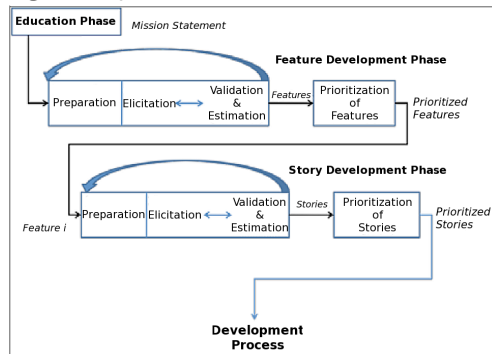
- One hybrid method is Soundararajan's Soft-Structured Agile Framework.
- This method consists of two parts:
  - Agile Requirement Generation Model (Agile RGM)
  - Development Process

# Agile Requirement Generation Model

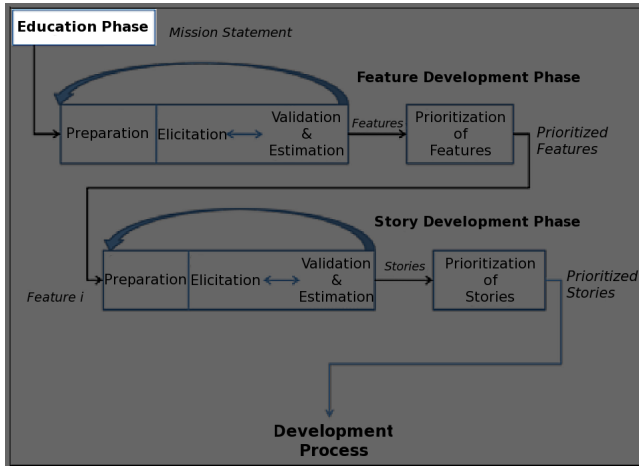
The Agile RGM is built of 3 phases:

- Education Phase
- Feature Development Phase
- Story Development Phase

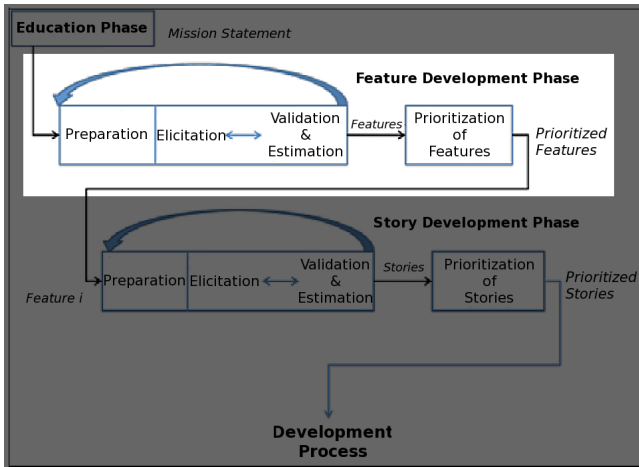
## Agile Requirement Generation Model



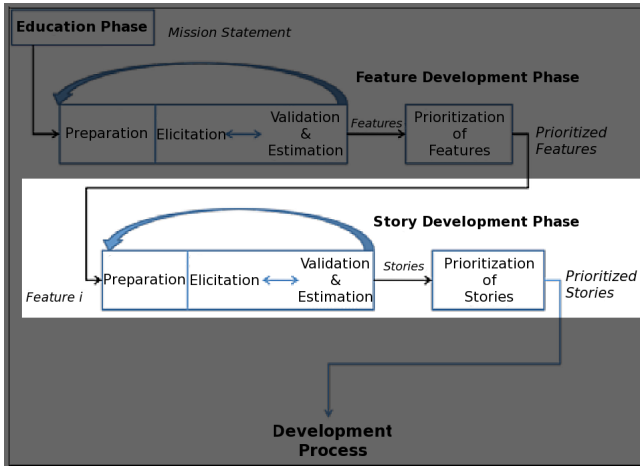
# Education Phase



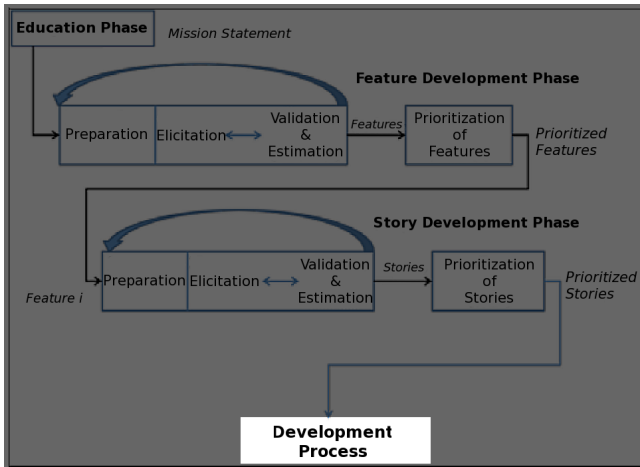
# Feature Development Phase



# Story Development Phase



# Development Process





# Development Process

At this point a development team can take one of two paths, depending on the scale of the project.

## Small- to Medium-Scale

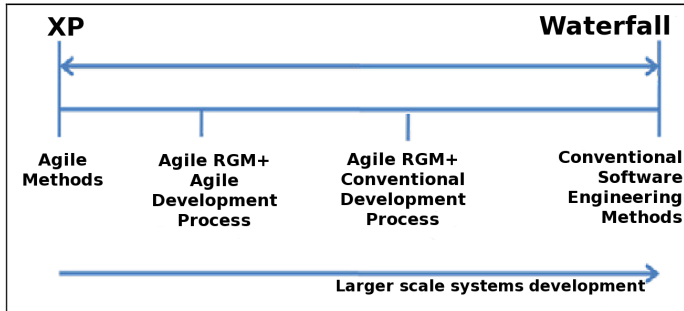
- Code and design from stories
- Small iteration
- XP-like practices

## Large-Scale

- Further transform stories into requirements
- Code and design from requirements
- Waterfall-like practices



# Benefits of the Soft-Structured Agile Framework



- This Method reflects fundamental agile practices and still meets needs for large-scale projects.

# Phasing into Large-Scale Agile Processes

- A development team can consider a method that phases into an agile method.
- This approach may be desirable because:
  - Allow for stable development of agile practices
  - Attractive from a business view-point



# Agile Software Solution Framework

- One phasing method is Qumer's Agile Software Solution Framework.
- This method consists of:
  - Combination of a set of tools to evaluate one's progress.
  - 3 phases that increment one towards their goals.
  - Process may take up to 3-5 years.



# First Phase

During the first phase:

- Focus on applying a subset of specific agile practices to their current development process
- Fine-tune each agile practice
- Incrementally continue to add new agile practices



# Second Phase

During the second phase:

- Introduce agile's key components that differentiate it from traditional methods.
- Example: Open communication, both within development teams, but also customer.



# Third Phase

During the third phase:

- The development team continues to fine-tune the agile practices.
- During this phase the development team should work to master the agile method, and continue to use it.

# Conclusion

- As this field is focused around business and competitive practices, information is scarce.
- Large-scale companies such as British Telecom and BMC Software are currently running agile methods
- With further research, and more companies adopting agile methodologies, information is to become more readily available for large-scale companies.
- It is still possible to scale to agile methods using the methods I've described.

# Questions?

## References:

- K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development. <http://www.agilemanifesto.org/>, 2001.
- T. Hildenbrand, M. Geisser, T. Kude, D. Bruch, and T. Acker. Agile methodologies for distributed collaborative development of enterprise applications. In *Complex, Intelligent and Software Intensive Systems*, 2008. CISIS 2008. International Conference on, pages 540-545, march 2008.
- A. Qumer and B. Henderson-Sellers. A framework to support the evaluation, adoption and improvement of agile methods in practice. *J. Syst. Softw.*, 81:1899-1919, November 2008.
- S. Soundararajan and J. Arthur. A soft-structured agile framework for larger scale systems development. In *Engineering of Computer Based Systems*, 2009. ECBS 2009. 16th Annual IEEE International Conference and Workshop on the, pages 187-195, april 2009.