

# Adopting Agile Practices

Amr Elssamadisy

Gemb

The logo for Gemba Systems, featuring a stylized lowercase 'a' in blue followed by the word 'systems' in a smaller, blue, sans-serif font.

# About Amr

- Technical roots as a C++, then Java, then .NET, then Java again then ...
- First Agile project in 1999 with ThoughtWorks. Caught the disease - have been spreading it since - hence Agile adoption is a main interest.
- Everything is based on context - the answer is “it depends”. Believe that patterns are a great way to communicate successes and failures.
- Author, editor for AgileQ on InfoQ, editor of Agile Journal.

© 2008 Gemba Systems.

All rights reserved.

# Gemb

 a systems

# About Amr contd..

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (LZW) decompressor  
are needed to see this picture.

© 2008 Gemba Systems.

All rights reserved.

# Gemb

**a**systems

# Adopting Practices <> Being Agile

- We all learn differently.
- Real knowledge comes from doing.
- Doing (sometimes) leads to Being

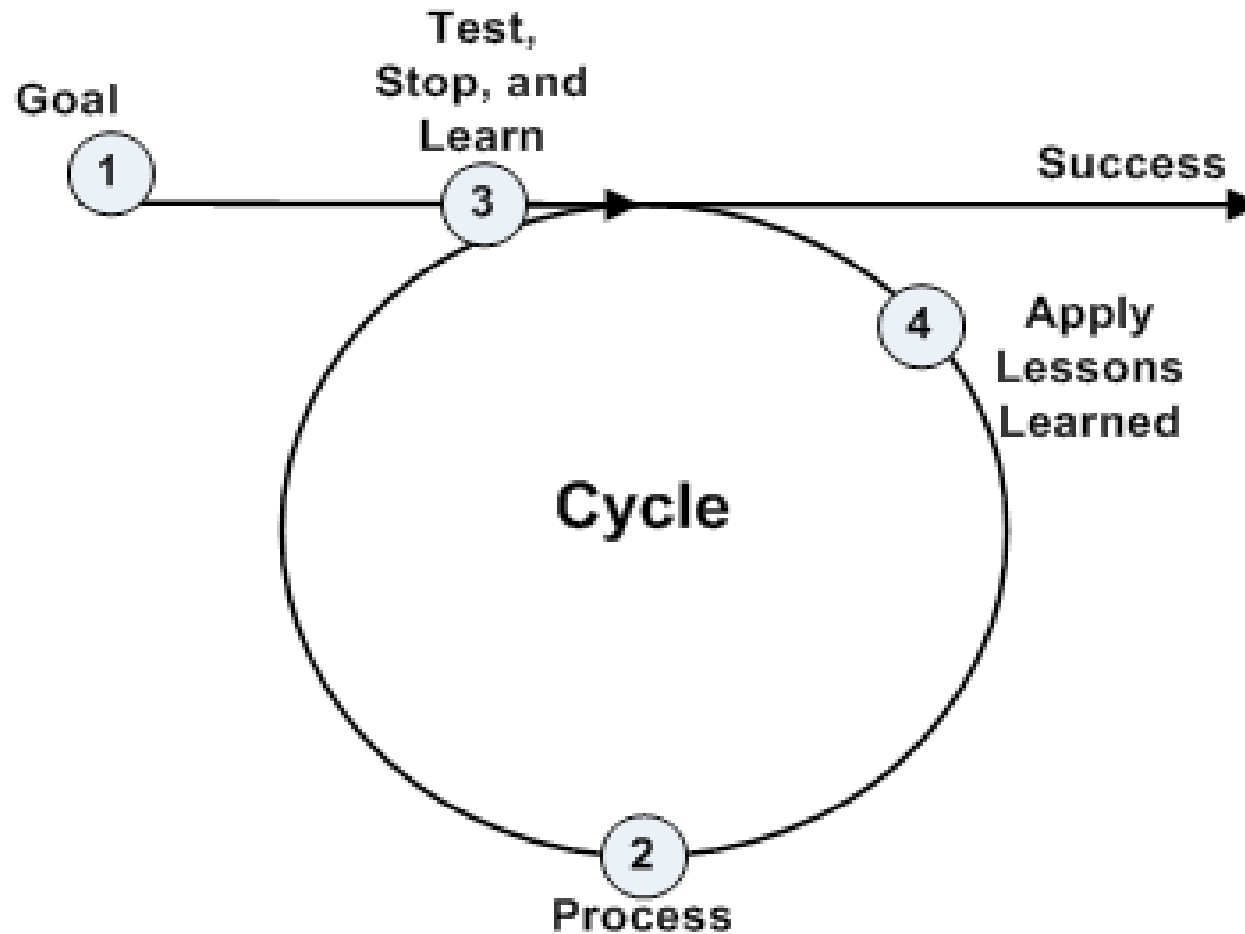
# Today's Goals

- Agree on the definition of Value.
- Understand the relationship between Value and Agile practices.
- Understand what Agile adoption patterns are, and how they can be leveraged.
- Examine as many practices as we can.

# What is Value?

- Is being Agile valuable?
- Is design valuable?
- Are requirements valuable?
- Is time to market valuable?
- Is user satisfaction valuable?
- Are meetings valuable?
- What is VALUE?

# A Learning Cycle



Gemb

a systems

# Agile Software Development

- Recognize and respond to change
- Feedback practices for the team
- Technical practices for developers



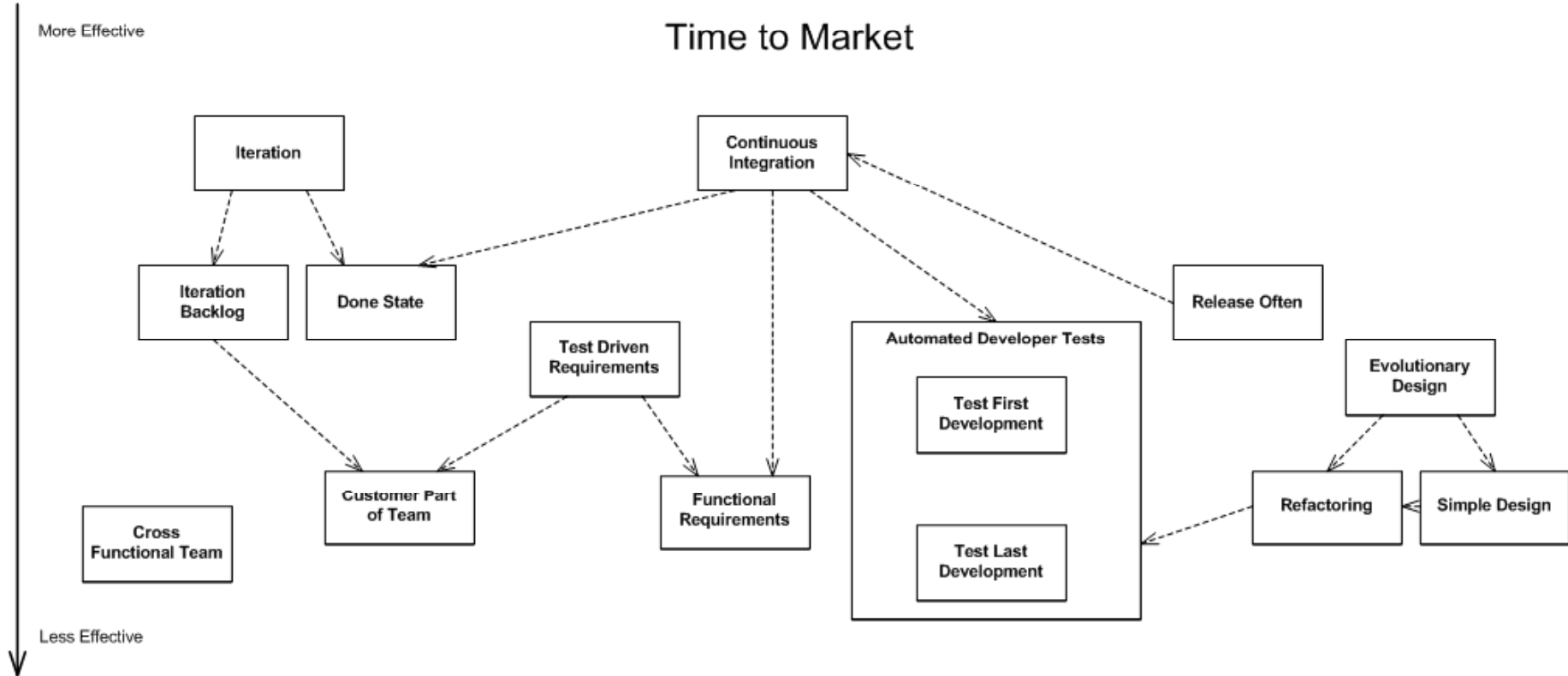
# Your Context: Business Value

- Reduce time to market
- Increase quality
- Increase flexibility
- Increase product utility
- Increase visibility
- Reduce cost
- Increase product lifetime

# Your Context: Smells

- Us vs. Them
- Customer asks for everything including the kitchen sink
- Direct input from customer is unrealistic
- Management is surprised
- Bottlenecked resources
- Churning projects
- Hundreds of bugs
- Hardening phase needed

# Decrease Time to Market



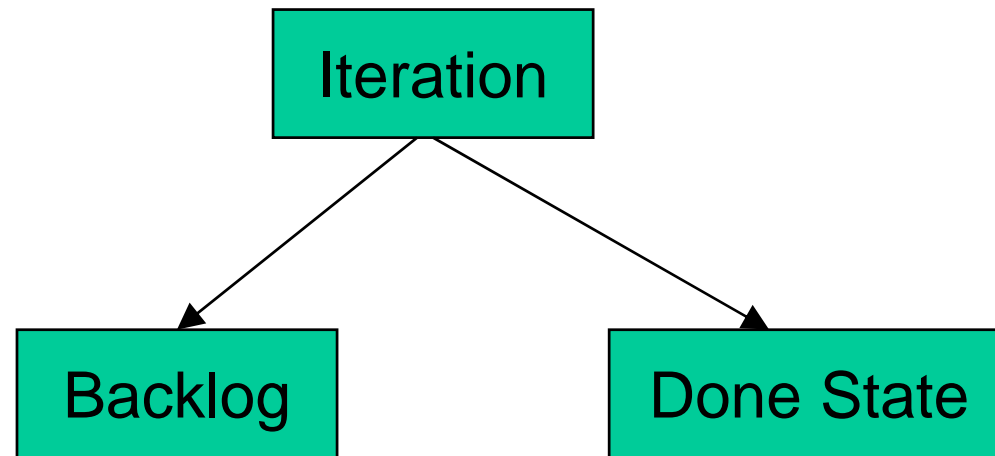
# Gemb

© 2008 Gemba Systems.

All rights reserved.

a systems

# Decrease Time to Market



# Backlog - Business Value

- Backlogs are a prioritized list of goals for iterations and releases.
- They improve product utility and increase visibility by enabling high quality feedback as the goals are met and reviewed regularly.
- In a sense, they also decrease time to market because the prioritized list enables negotiation of scope and early release with the most valuable functionality.

Gemb

# Backlogs - Context

- You are on a development team that has decided to adopt some agile development practices including Iteration.
- You have decided to go away from the old technique of up front requirements and a detailed specification document as a “hand off” from analysts to developers.
- You have the needed expertise on your team to incrementally expand and evolve requirements either via Customer Part of Team or any other practice.

Gemb

© 2008 Gemba Systems.

All rights reserved.

The logo for Gemba Systems, featuring a stylized lowercase 'a' in blue followed by the word 'systems' in a smaller, blue, sans-serif font.

# Backlogs - Adoption

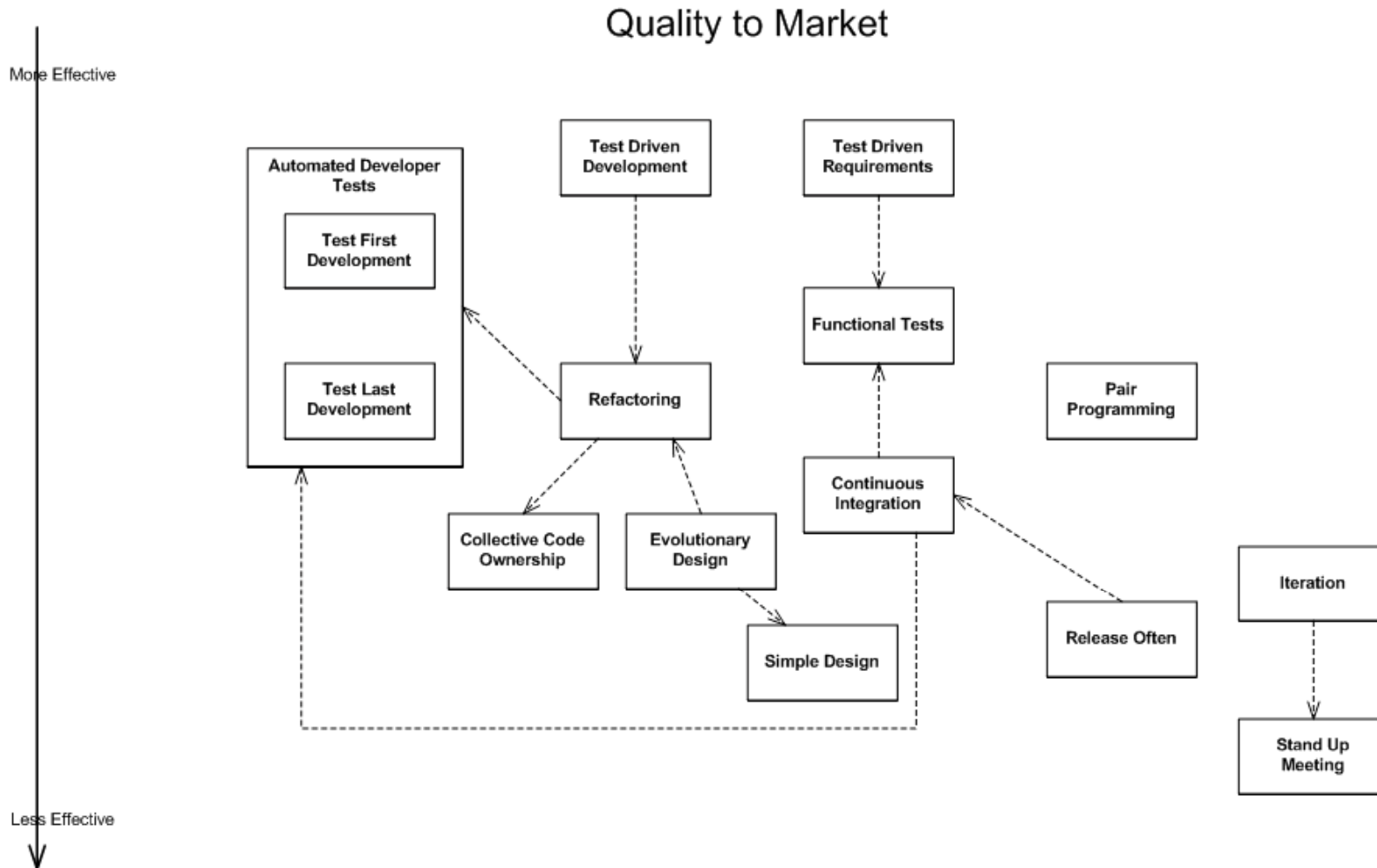
- The customer flushes out the coarse-grained requirements ahead of the iteration planning meeting. This frequently means that they are preparing for the next iteration half-way through the current iteration.
- At the beginning of each iteration, in the kickoff meeting, the team should understand the requirements to the level that they can:
  - roughly estimate (via planning poker, etc...)
  - to begin development
- The items chosen for development make up the iteration backlog.

# Backlogs - Smells

- Estimation paralysis.
- Techies take over.
- Multiple non-cross-functional teams have trouble working from one backlog.



# Increase Quality



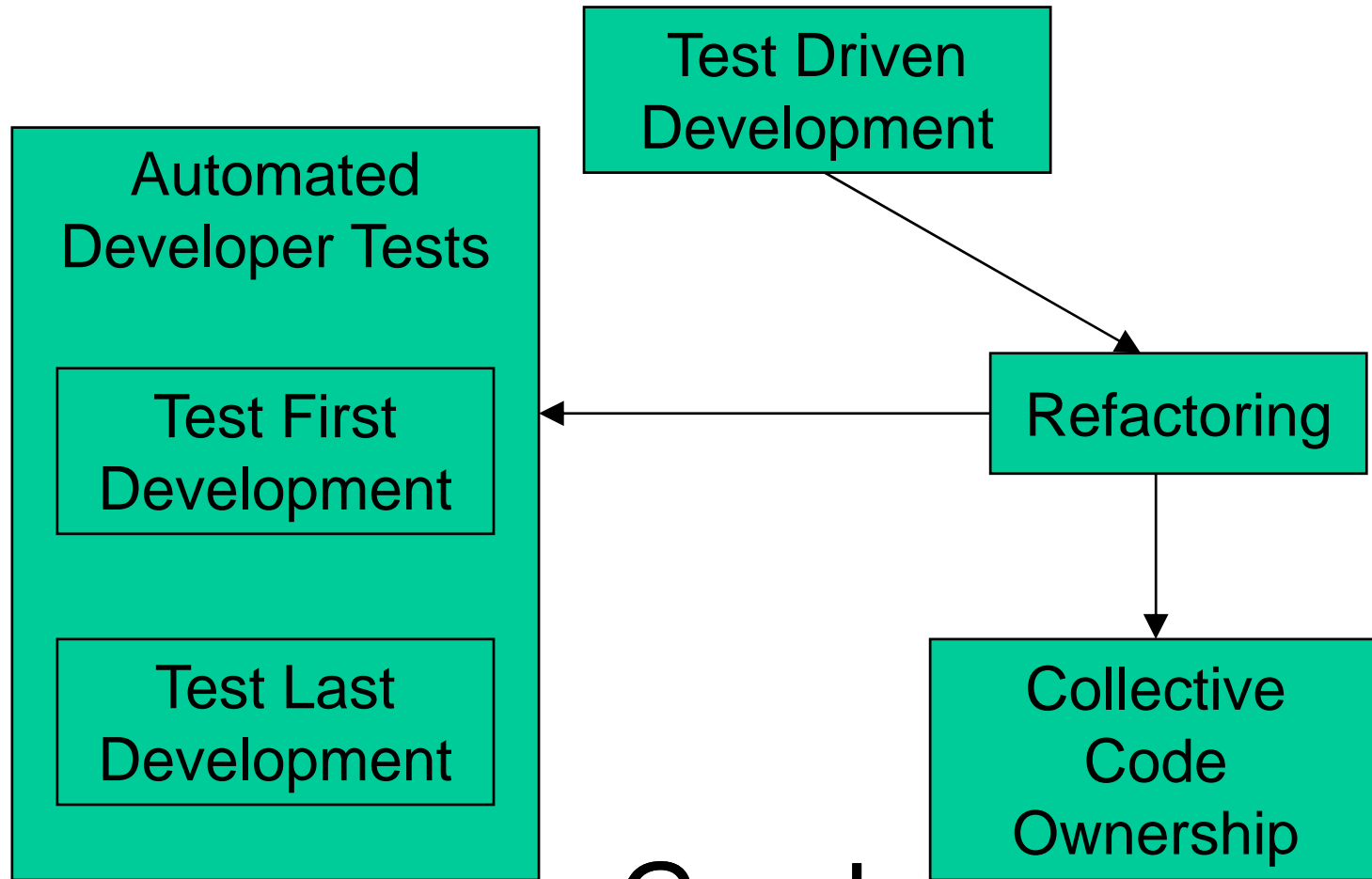
# Gemb

© 2008 Gemba Systems.

All rights reserved.

a systems

# Increase Quality



© 2008 Gemba Systems.

All rights reserved.

# Gemba

**a**systems

# Refactoring - Business Value

- Refactoring increases flexibility and the product lifetime by enabling and encouraging developers to change the design of the system as needed.
- Quality to market and costs are reduced because continuous refactoring keeps the design from degrading over time, ensuring that the code is easy to understand, maintain, and change.

# Refactoring - Context

- You are on a development team that is practicing automated developer tests.
- You are currently working on a requirement that is not well-supported by the current design.
- Or you may have just completed a task (with its tests of course) and want to change the design for a cleaner solution before checking in your code to the source repository.

# Refactoring - Adoption

- This is one of those "just do it!" patterns (well almost...).
- Keep in mind is that Refactoring is a practice and not a tool—although tool support helps.

# Refactoring - Adoption

1. Start automated developer tests until you are comfortable with the discipline of writing tests for all of your tasks.
2. In a team environment, adopt collective code ownership on your team—agree on how to handle broken tests from refactoring in a timely manner.
3. Pick up a copy of *Refactoring: Improving the Design of Existing Code* by Martin Fowler and a book about test-driven development that is exercise-driven.

## Gemb

# Refactoring - Adoption

4. Start. Perform the steps as described above. For every task inspect the design to see if it needs changing to accommodate the new work.
5. Run a bi-weekly study-group to share different refactorings that have been performed.
6. As you become comfortable with the canonical refactorings, be courageous and make significant changes. Work towards large design changes that you and your team have known were needed when appropriate

Gemb

# Refactoring - Smells

- Don't get carried away and refactor just for the sake of refactoring. Remember, refactoring delivers no direct business value.
- Many missed small refactorings build up over time causing the need for large refactorings.
- Beware of code-ownership and pride causing “commit wars” once something is refactored.
- In a team environment you will eventually refactor code that causes tests that you have not written to break. Some new to Agile practices may check in this code and rely on the developers who have written the tests to fix the broken test.

## Gemb



# More Information...

Amr Elssamadisy

[amr@gembasystems.com](mailto:amr@gembasystems.com)

<http://www.gembasystems.com>

Free book on InfoQ:

<http://www.infoq.com/minibooks/agile-patterns>

Business Value to Practice Mappings:

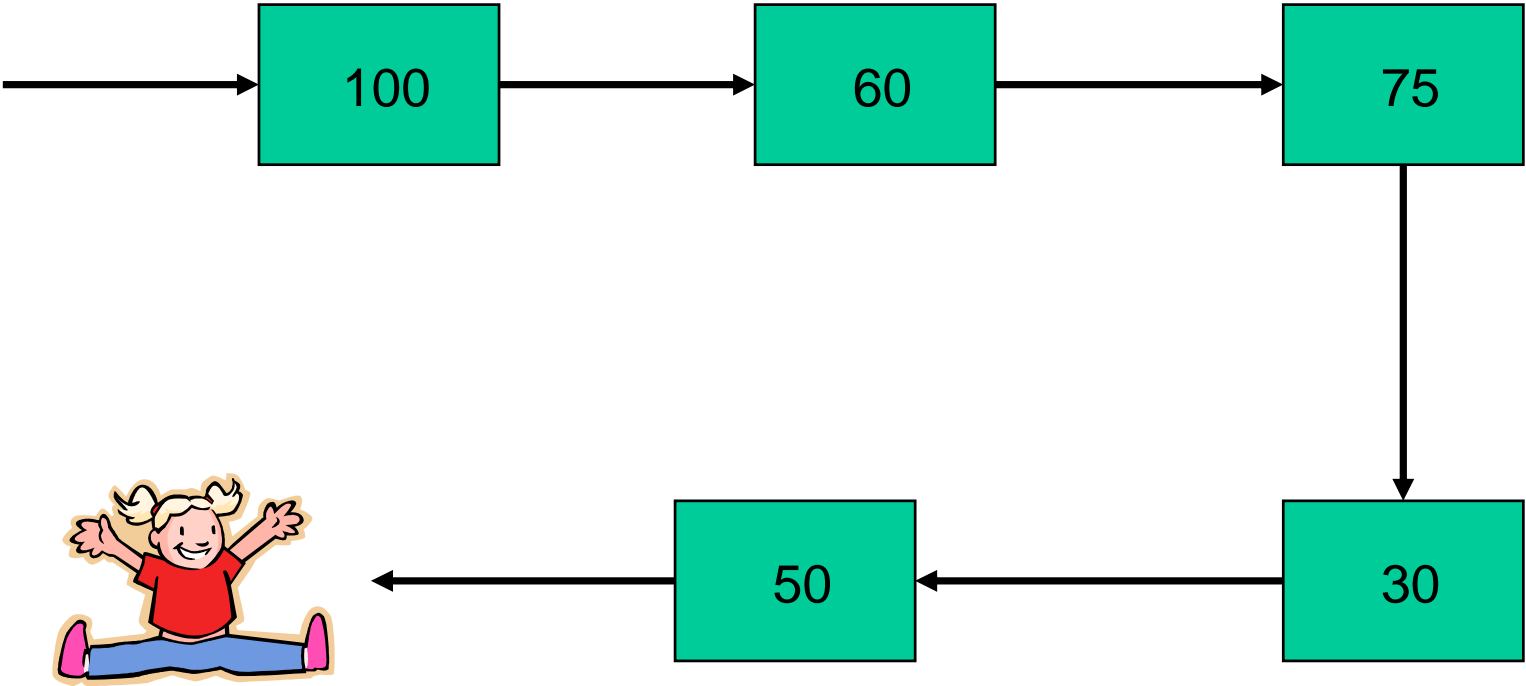
<http://www.informit.com/articles/article.aspx?p=1235050>

1.  Introduction: Agile is not the goal, but the means to achieving your goals.
2. Review different business goals and process smells
3. Tie goals and smells in (2) to Agile practices
4. Go through a group exercise led by the instructor to elaborate the goals of a fictional company and choose the practices to address those goals.
5. Run small group sessions in parallel, where the students work through (4) for their real-world situations
6. Review and present (5) and have a group critique.

# Lean Principles

1. Value is....
2. Value stream
3. Flow
4. Pull
5. Perfection

# Theory of Constraints



© 2008 Gemba Systems.  
All rights reserved.

# Gemba

 systems