



Introduction to Agile Software Development Process

Presenter: Soontarin W. (Senior Software Process Specialist)

Date: 24 November 2010

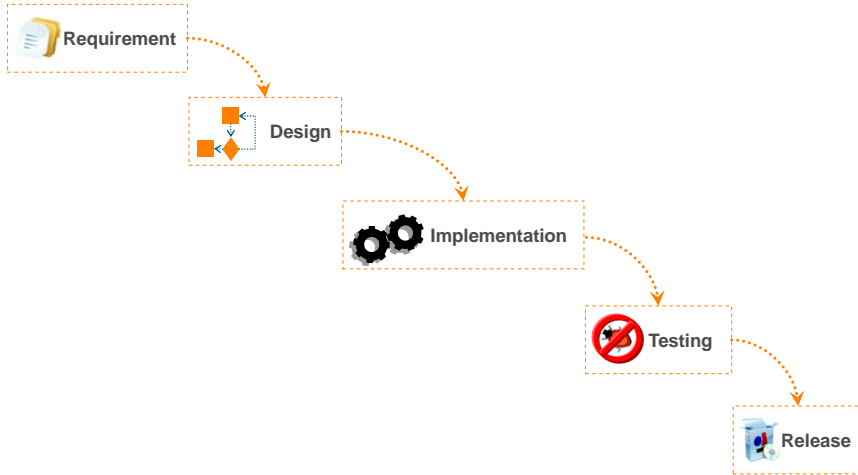


AGENDA

- Software Development Life Cycles
 - Waterfall Model
 - Iterative & Incremental Model
- Agile
 - What is Agile?
 - Agile Principles
 - Agile Practices
 - Retrospective result from Agile Team

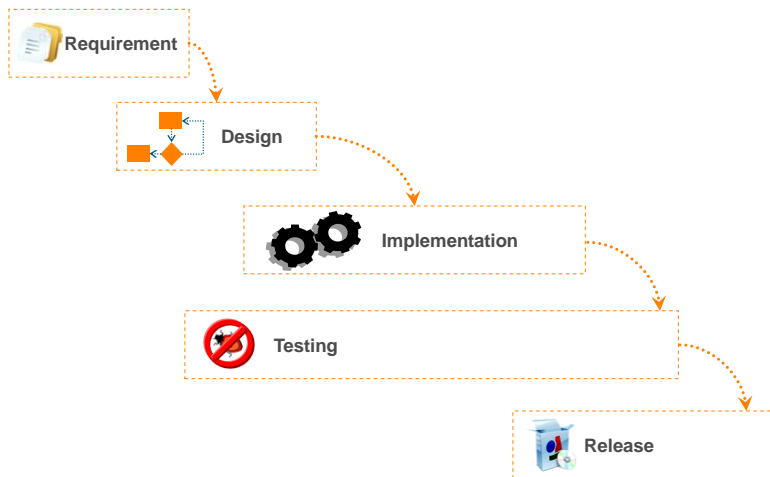
Waterfall Model

Traditional Model (since 1970)



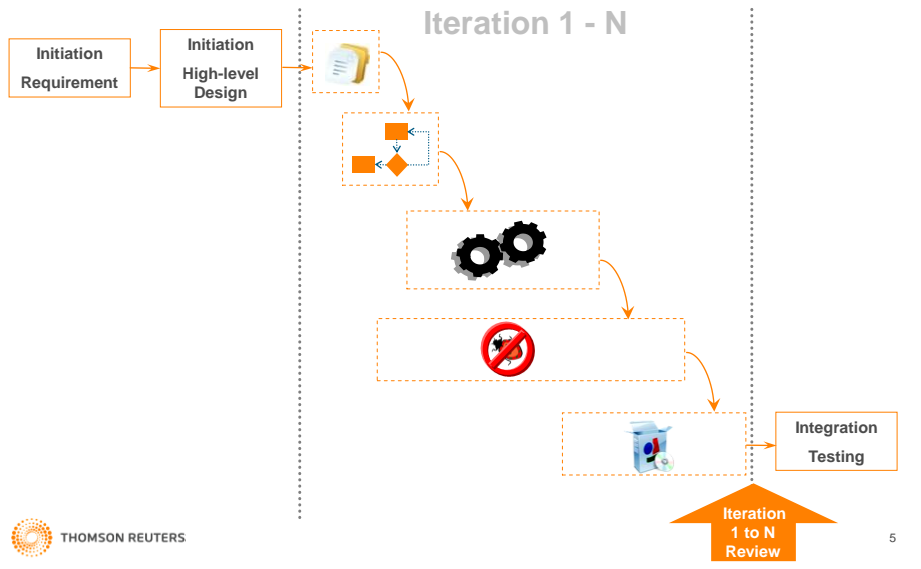
Waterfall Model

Customized in Real Life



Iterative & Incremental Model

Waterfall in Small Pieces



In early 2001, a group of industry experts in light-weight process like XP, Scrum, FDD, Crystal, DSDM, and etc, formed the Agile Alliance and created

The Manifesto of the Agile Alliance.

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.

agilemanifesto.org

Agile

Is It the New Thing????



What is Agile?

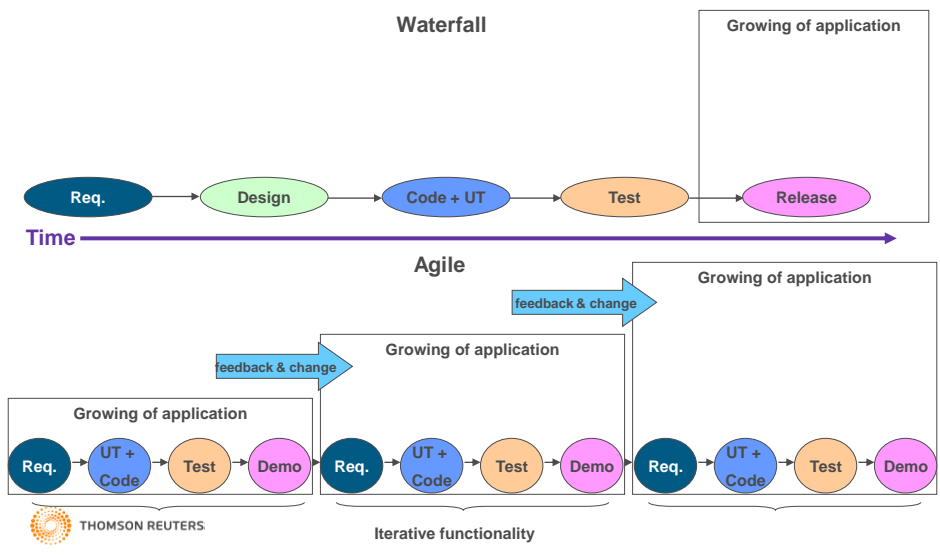
- **Group of software development methodologies** based on iterative development
- **Collection of values, principles, and practices** for modelling software that can be applied on a software development project in an effective and light-weight manner
- Examples of software development methodologies which are based on agile development are below:
 - Extreme Programming (XP)
 - Scrum
 - Feature-Driven Development (FDD)

Agile Principles

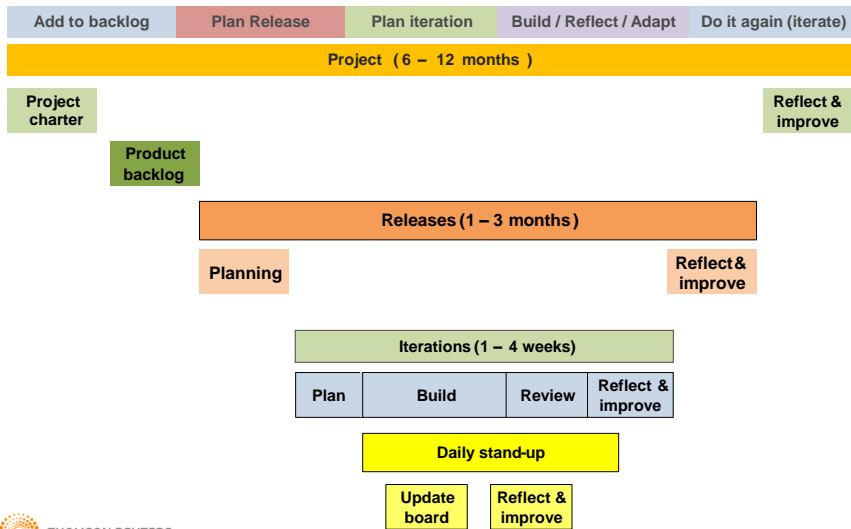
- **Customer satisfaction** by rapid, **continuous delivery** of useful software
- Working software is **delivered frequently**
- **Working software** is the principal **measure of progress**
- Even late **changes in requirements** are welcome
- Regular **adaptation to changing** circumstances
- **Close, daily, cooperation** between business **people** and developers
- **Face-to-face** conversation is the best form of communication
- Projects are built around **motivated individuals**, who should be **trusted**
- **Self-organizing teams**
- **Simplicity**
- Continuous attention to **technical excellence and good design**



Agile The Way Application is Made



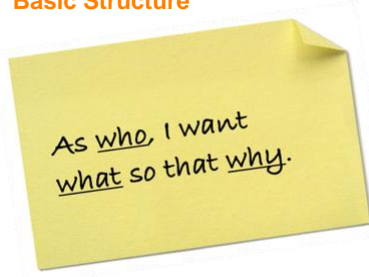
Agile Development Cycle



Agile Practices

User Story – Requirement is the story of user

Basic Structure

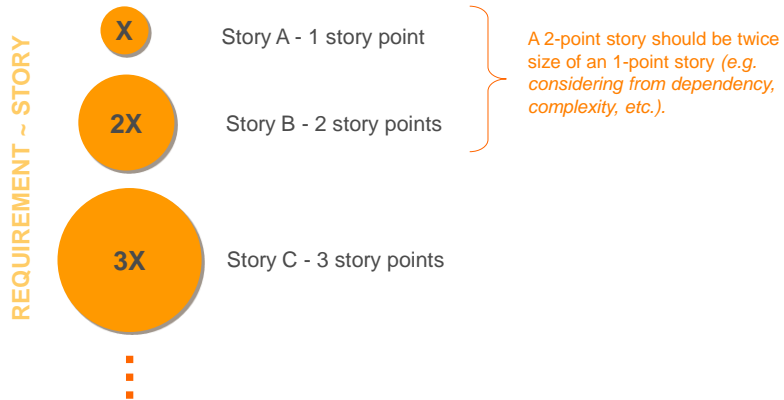


Example

- As a librarian, I want to be able to search for books by author name.
- As a system admin, I can configure permissions and user groups.
- As a buyer, I want to search a product online, so that I can see which one I like.

Agile Practices

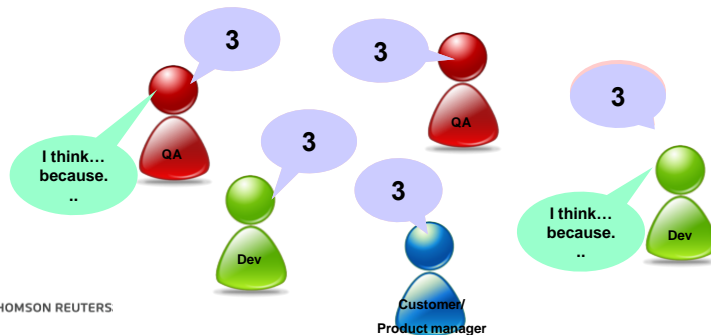
Story Point – Agile Size Estimation



Agile Practices

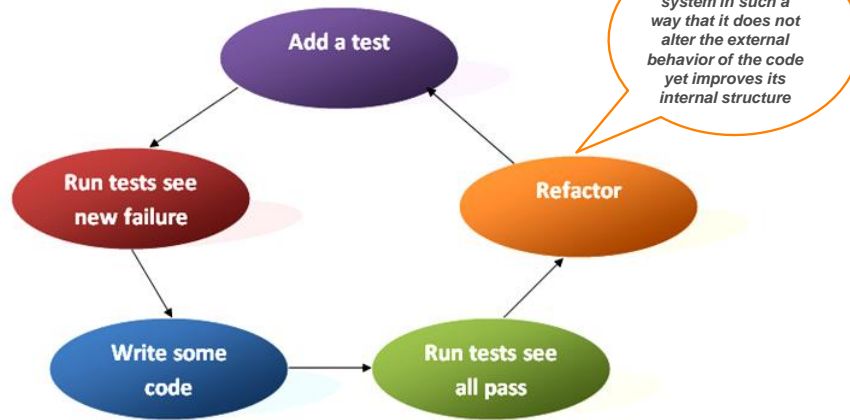
Planning Game

- User story:**
As a shopper, I want to pay by credit card.
- Let's define exit criteria of this user story.**
- How many story points you will give for this story?**
- Why 2 and why 5? Then re-estimate**



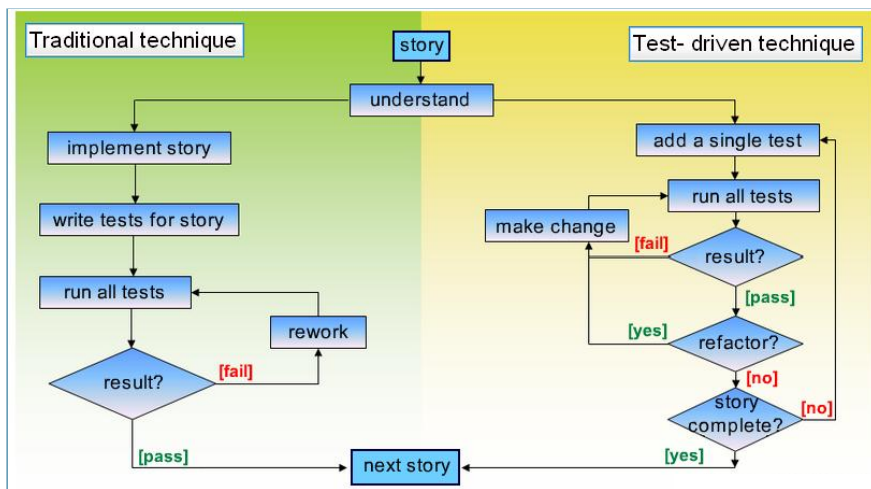
Agile Practices

TDD – Test Driven Development



Agile Practices

Traditional Technique vs. TDD Technique



Agile Practices

Pair Programming

Example Steps:

1. Well-defined before sit down
2. Agree on solution
3. Agree on tiny goal at a time
4. Rely on & support your partner
5. Talk a lot
6. Often sync. Up
7. Take a moment to celebrate when accomplishing a thing
8. Switch role frequently



Agile Practices

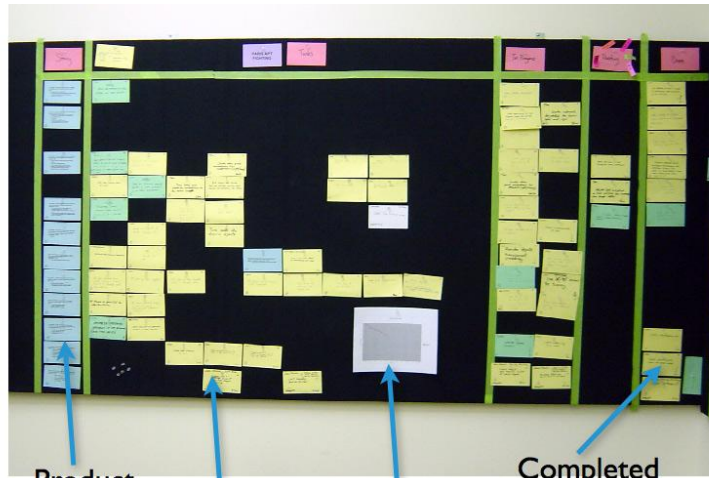
Scrum Meeting

- (Scrum) Daily Stand Up Meeting
- What will be discussed in daily stand up meeting?
 - What I did yesterday is....
 - I found X issues, they are....
 - Today I plan to do



Agile Practices

Story Board



Product backlog

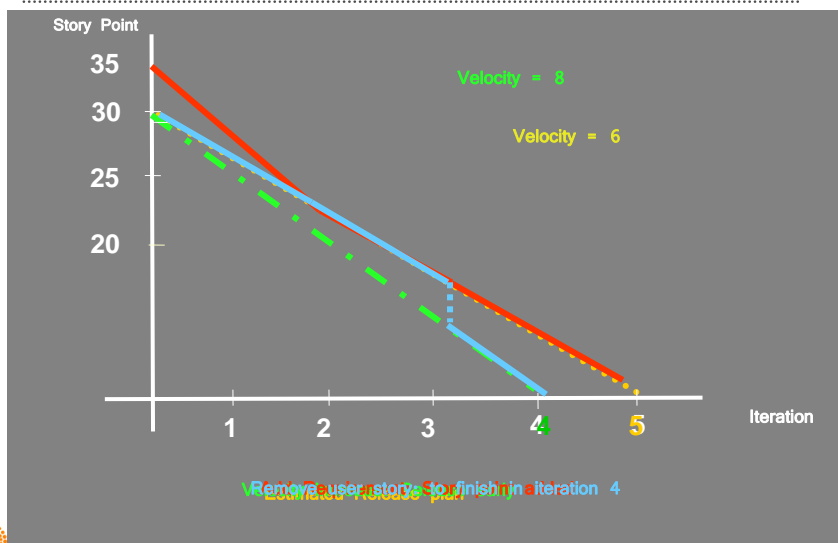
Tasks to do

Burndown chart

Completed tasks



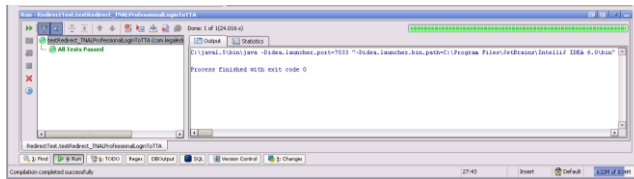
Burndown Chart



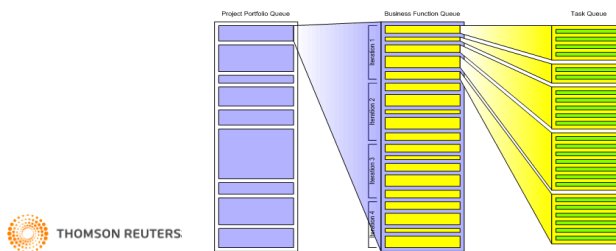
Agile Practices

Testing in Agile Project

- Automated Unit Tests



- Small, Testable Chunks of Work



Agile Practices

Continuous Integration

“a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day”

[Martin Fowler 2006]

Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible

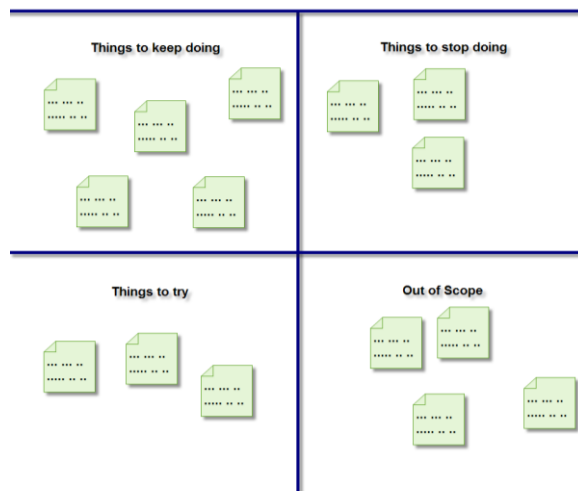
Agile Practices

Continuous Integration – Example Software

- **Cruise Control** – Java-based framework for a continuous build process
- **CruiseControl.NET** - .NET-based automated continuous integration server
- **Apache Continuum** – a continuous integration server supporting Apache Maven and Apache Ant
- **Hudson** – MIT licensed, written in Java, runs in servlet container, support CVS, Subversion, Ant, Maven, and shell scripts

Agile Practices

Retrospective



Retrospective result from Agile team

Positive Feedback

- 😊 Found and solved defects earlier
- 😊 Focus on each specific task
- 😊 Better understanding on requirement
- 😊 Able to release value software as chunk works
- 😊 Easy and fast track on stand up meeting
- 😊 Good collaboration between Dev & QA
- 😊 Get more attention and increase participation from team
- 😊 Sharing knowledge between team as a community



Retrospective result from Agile team

Challenging Feedback

- 😞 Difficult to apply user story and story point
- 😞 Difficult to provide all metrics
- 😞 Require time for changing team culture and learning technique
- 😞 Require coach to provide consult on techniques and process
- 😞 No product manager or customer involvement





*No Best Technique,
Only Suitable Technique
Team Find out*

THANK YOU