Agile

Values, Principles, & Practices for Software Dev't

Agile Manifesto

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**.

Values & Mindset

The most important part of Agile is the *mindset*.

The *mindset* is expressed (or informed) by the *values*.

Different people express the values differently:

- Communication

- Customer Collaboration

- Incremental Development & Minimalism

- Sustainability

Practices Based on Values

The Agile Practices come from the values.

Adopting practices without values often fails.

Example: Daily Meetings

A project manager wants to improve his team's ability to delivery good quality software on-time.

He/she reads about Agile and decides that the **daily standup meeting** sounds like a great place to start!
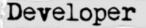
(Many organizations do this.)

So he tells his team...

Ref: example from Chapter 1 of *Learning Agile* by Stellman and Green. This is the best book I know about Agile.

Let's have a Daily Standup

What Went Wrong?

The development team <u>resents</u> the idea.

They see the standup meeting as:

a) interrupting their work,

b) lack of trust that they are effectively using their time,

c) nagging

Lack of Mutual Understanding

What do they want to get from the meeting?

Manager:

- wants to know if progress is deviating from plan

- wants to know what developers are doing

Developer:

- wants the meeting to end as soon as possible

Change in Mindset

Manager needs to change his mindset and communicate the purpose of the meeting...

- Not about deviating from <u>his</u> plan
- *Meeting is <u>not</u> a status report from developers*

Instead:
- *<u>everyone on the team</u> works together to create a plan*
- *involve whole team in discovering ways to work better*
- *help developers obtain the time & resources they need*
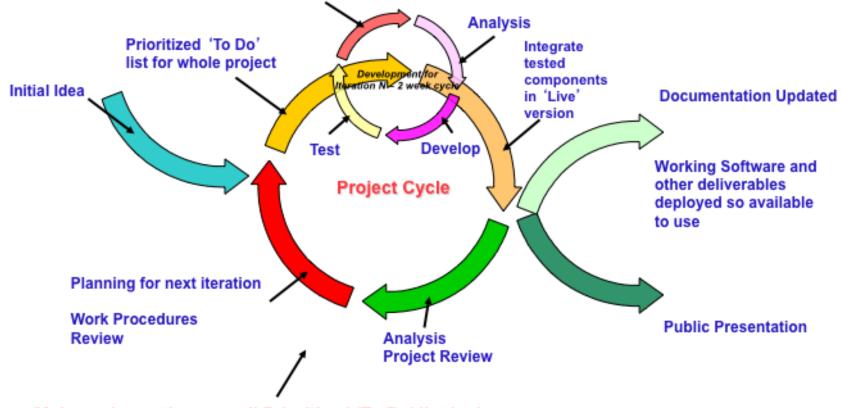
Change in Mindset of Developers

When developers feel that they "own" the project plan and the meeting is about their input...

Agile Development Lifecycel

Iteration diagram adapted for Agile methodology

Produce / Revise 'To Do' list just for this iteration of project – development cycles until list is complete for this iteration

Analysis

Prioritized 'To Do' list for whole project

Integrate tested components in 'Live' version

Initial Idea

Development for Iteration N – 2 week cycle

Documentation Updated

Test

Develop

Project Cycle

Working Software and other deliverables deployed so available to use

Planning for next iteration

Work Procedures Review

Analysis Project Review

Public Presentation

Main cycle continues until Prioritized 'To Do' list is done

Agile Process Characteristics

□ create customer "value" at each iteration

□ welcome evolving requirements

□ working software as primary measure of progress

□ lack of up-front architecture design (YAGNI)

□ simple design (XP: "*do the simplest thing that ...*")

□ small, self-organizing teams at one site

□ frequent customer feedback

□ shared understanding instead of *comprehensive* documents … but *do* write documentation ...

Some Agile Processes

- eXtreme Programming

  - Kent Beck: Chrysler

- Scrum - called "more a management technique"

  - iterative development in "Sprints"

  - daily stand-up meeting

  - no advise about requirements

- Crystal

  - a family of methods to address different types of projects

- Synchronize and Stabilize (Microsoft process)