Chapter 3

- **Agile Development**

Slide Set to accompany

Software Engineering: A Practitioner’s Approach, 7/e

by Roger S. Pressman

The 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.”

Kent Beck et al 2001

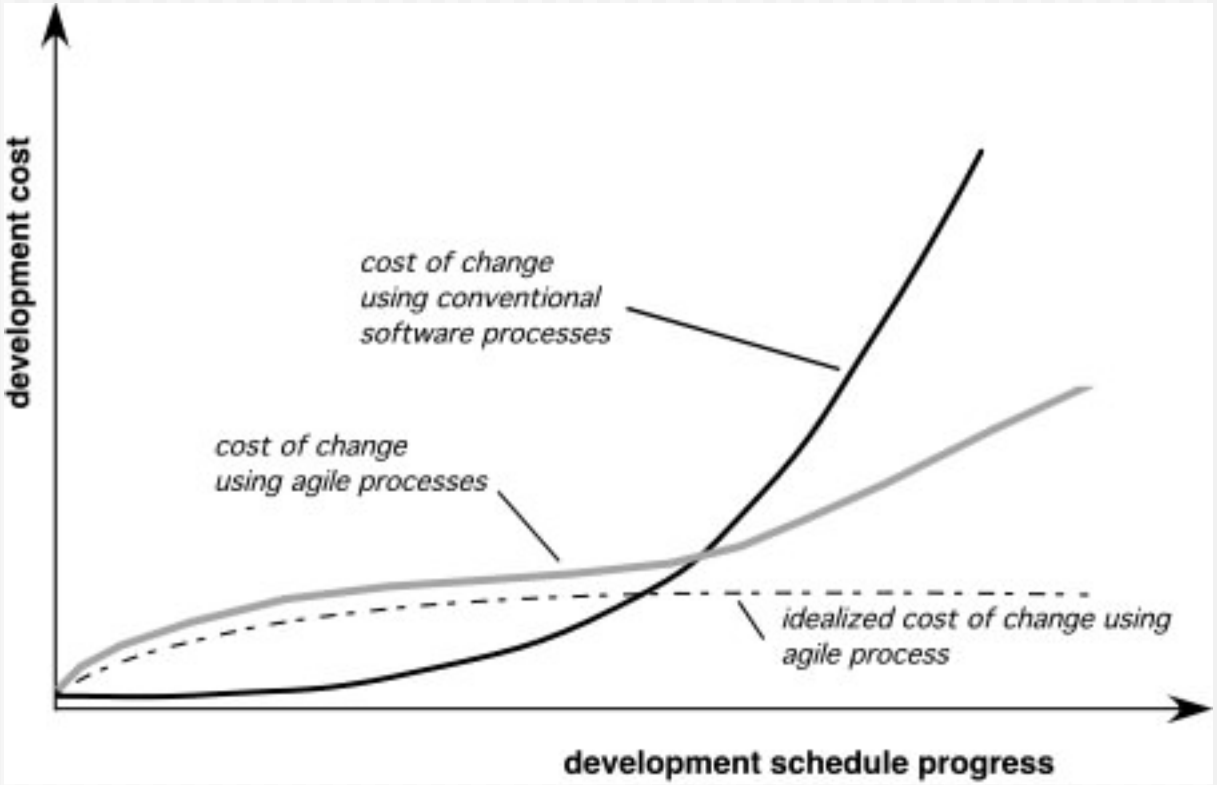
What is “Agility”?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding …

- Rapid, *incremental delivery* of software

Agility and the Cost of Change



These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/ e* (McGraw-Hill, 2009) Slides copyright 2009 by Roger Pressman.

An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adapts as changes occur

Agility Principles - I

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agility Principles - II

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face–to–face conversation.

Agility Principles - III

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility

Agility Principles - IV

10. Simplicity – the art of maximizing the amount of work not done – is essential.

11. The best architectures, requirements, and designs emerge from self–organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Human Factors

- *the process molds to the needs of the people and team*, not the other way around
- key traits must exist among the people on an agile team and the team itself:
    - **Competence** – encompasses innate talent, specific software-related skills, and overall knowledge of the process to be employed.
    - **Common focus** – deliver a working software increment to the customer within the time promised.
    - **Collaboration** – team members must collaborate with each other and all other stakeholders.

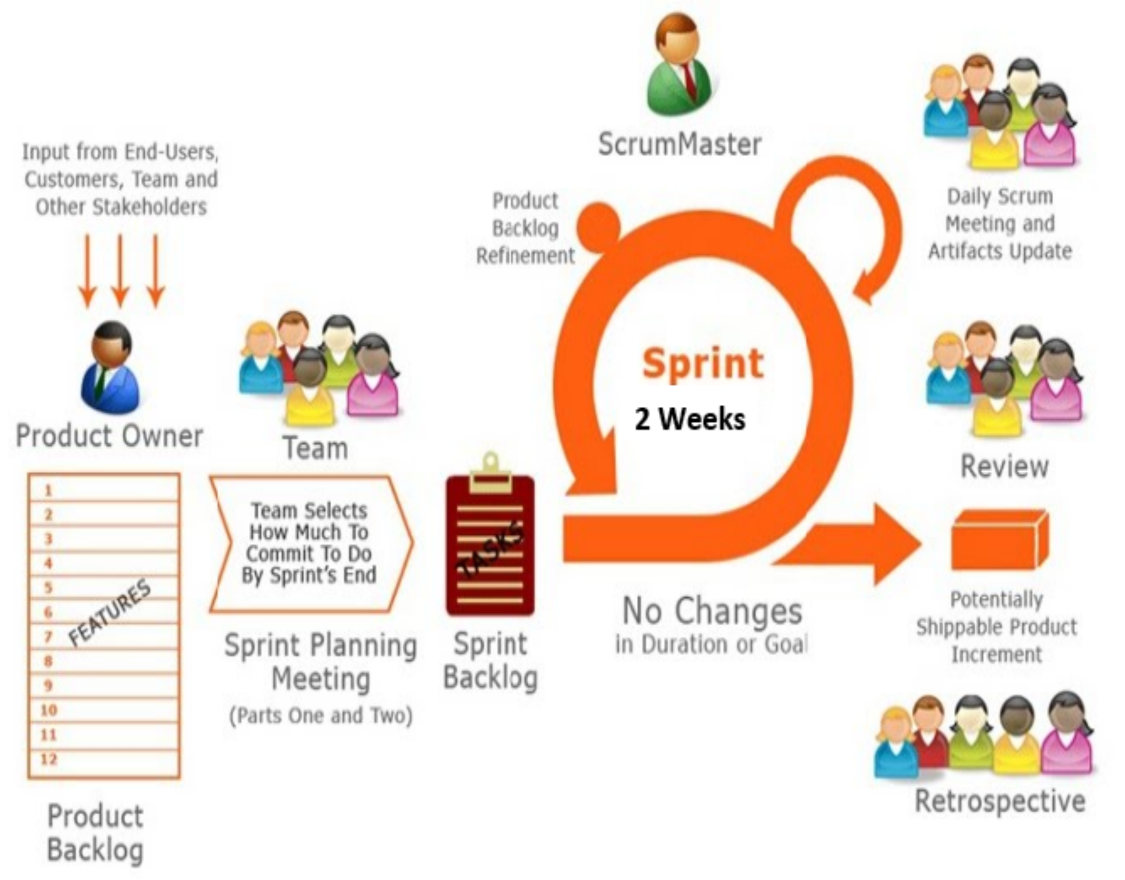
Human Factors

- **Decision-making ability** – team given autonomy (decision-making authority for both technical & non-technical issues).
- **Fuzzy problem-solving ability** – software managers must recognise that agile team will continually have to deal with *ambiguity*, and be buffeted by *change*. Lessons learned from addressing such issues may benefit team at later juncture.
- **Mutual trust and respect** – group becomes a *"jelled"* team.
- **Self-organization** – team *organises itself* for the work to be done; team *organises the process* to best accommodate local environment; team *organises work schedule* to best achieve increment-delivery.

Scrum

- Originally proposed by Schwaber and Beedle
- Scrum—distinguishing features
  - Development work is partitioned into "packets"
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in "sprints" and is derived from a "backlog" of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - "demos" are delivered to the customer with the time-box allocated

Scrum


Input from End-Users,
Customers, Team and
Other Stakeholders
Product Owner
Team
ScrumMaster
Product
Backlog
Refinement
Sprint
2 Weeks
Daily Scrum
Meeting and
Artifacts Update
Review
FEATURES
1
2
3
4
5
6
7
8
9
10
11
12
Team Selects
How Much To
Commit To Do
By Sprint's End
TASKS
No Changes
in Duration or Goal
Potentially
Shippable Product
Increment
Product
Backlog
Sprint Planning
Meeting
(Parts One and Two)
Sprint
Backlog
Retrospective