COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

CHAPTER 3

AGILE SOFTWARE DEVELOPMENT

PROF. DR. KAMRUDDIN NUR
PROFESSOR, CS, AIUB
https://cs.aiub.edu/profile/kamruddin

AGILE DEVELOPMENT

"Plan-driven methods work best when developers can determine the requirements in advance . . . and when the requirements remain relatively stable, with change rates on the order of one percent per month."~ *Barry Boehm*

- Agility is the ability to create and respond to change in order to profit in a turbulent business environment
- Companies need to
  - innovate better and faster operations
  - respond quickly to
    - competitive initiatives
    - new technology
    - customer's requirements

AGILE MODEL

- ❑ Subset of iterative and evolutionary methods

Iterative Products

- Each iteration is a self-contained, mini-project with activities that span requirements analysis, design, implementation, and test
- Leads to an iteration release (which may be only an internal release) that integrates all software across the team and is a growing and evolving subset of the final system
- The purpose of having short iterations is so that feedback from iterations N and earlier, and any other new information, can lead to refinement and requirements adaptation for iteration N + 1

AGILE METHODS VS. PAST ITERATIVE METHODS

- A key difference between agile methods and past iterative methods is the length of each iteration

- In the past, iterations might have been three or six months long
- In agile methods, iteration lengths vary between one to four weeks, and intentionally do not exceed 30 days
- Research has shown that shorter iterations have lower complexity and risk, better feedback, and higher productivity and success rates

TIMEBOX & SCOPE

- ❑ The pre-determined iteration length serves as a timebox for the team.
- ❑ Scope (set of tasks) is chosen for each iteration to fill the iteration length.
- ❑ Rather than increase the iteration length to fit the chosen scope,
  the scope is reduced to fit the iteration length.

AGILE PROPONENTS BELIEVE

- Current software development processes are too heavyweight or cumbersome
  - Too many things are done that are not directly related to software product being produced
- Current software development is too rigid
  - Difficulty with incomplete or changing requirements
- Short development cycle is needed
- More active customer involvement needed

WHAT IS AN AGILE METHOD?

- Agile methods are considered
  - Lightweight (do not concentrate on the whole s/w development at once)
  - People-based rather than Plan-based
- Several agile methods
  - No single agile method
  - Different agile methods can be combined in s/w development (Hybrid)
- No single definition. Agile Manifesto closest to a definition
  - Set of principles
  - Developed by Agile Alliance

AGILE VALUE STATEMENT

- In 2001, Kent Beck and 16 other noted software developers, writers, and consultants (known as Agile Alliance) signed a manifesto as following:

"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

AGILE VS. PLAN DRIVEN PROCESS

Agile Process	Plan Driven Process
Small products and teams; scalability limited	Large products and teams; hard to scale down
Inappropriate for safety-critical products because of frequent changes	Handles highly critical products
Good for dynamic, but expensive for stable environments.	Good for stable, but expensive for dynamic environments
Require experienced Agile personnel throughout	Require experienced personnel only at start if stable environment
Personnel succeed on freedom and chaos	Personnel succeed on structure and order

AGILE ASSUMPTION

- It is difficult to predict in advance which software
  - requirements will persist and which will change
  - It is equally difficult to predict how customer priorities will change as the project proceeds

- Design and construction are interleaved in many types of software. That is, both activities should be performed tightly so that design models are proven as they are created. It is difficult to predict how much design is necessary before construction is used to prove the design.
- Analysis, design, construction, and testing are not as predictable (from a planning point of view) as we might like.

AGILE MANIFESTO (POLICY)

1. Our highest priority is to satisfy the costumer through early and continuous delivery of valuable software
2. Welcome changing requirements, even late in development. Agile process harness (control) change for the customer´s competitive advantage
3. Deliver working software frequently with a preference to the shorter timescale
4. Business people and developers must work together daily throughout the project
5. Build projects around motivated individuals. Give them the environment and support their 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

AGILE MANIFESTO (POLICY)

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 persistently
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity – use simple approaches to make changes easier
11. The best architectures, requirements, and designs emerge from self-organizing teams (iterative development rather defined plans)
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

HUMAN FACTORS IN AGILE DEVELOPMENT

- ❑ Competence/skill/capability
- ❑ Common focus
- ❑ Collaboration
- ❑ Decision-making ability
- ❑ Fuzzy (vague) problem-solving ability
- ❑ Mutual trust and respect
- ❑ Self-organization

AGILE METHODS

- Extreme Programming (XP)
- Scrum
- Dynamic Systems Development Method (DSDM)
- Feature-Driven Development (FDD)
- Crystal Methods
- Lean Development (LD)
- Adaptive Software Development (ASD)

REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.