소프트웨어 공학 원리
(SEP521)

Agile Mehods

Jongmoon Baik

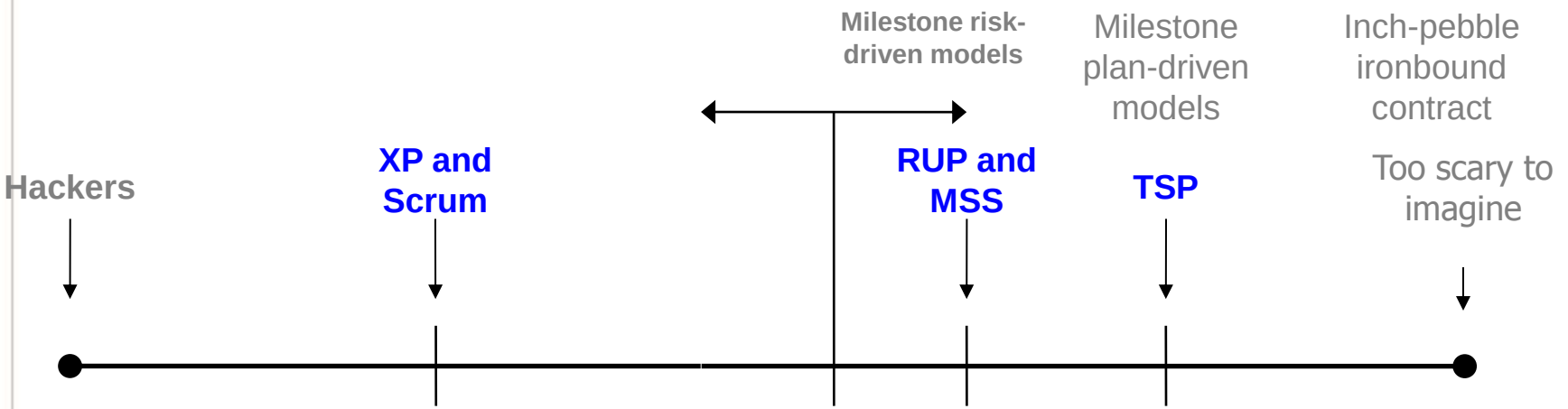
KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
KAIST
한국과학기술원
SINCE 1971

Problems in Current S/W Developments

- Systems are regularly delivered late or over budget if at all
- Customers do not trust nor work with developers because they have been burned so many times
- Software developers are getting burnt out because of long hours

Process Spectrum

Weight = amount of project overhead/code



"Light"
Artifact Challenged

"Heavy"
Artifact Heavy

Adapted from Justin Rockwood, "Choose your Weapon Wisely," 2003

What is Agile Methods?

- Webster Dictionary:
  - **"Marked by ready ability to move with quick easy grace"**
- As applied to Software Development:
  - **"Ability to change development in response to changing requirements"** **Cockburn**
- The Agile Manifesto [http://agilemanifesto.org/]
  - "A method of software development that aims for customer satisfaction through early and continuous delivery of useful software components"

Why Agile Processes ?

What agile proponents say:

- Flexibility
  - Market Changes
  - Technology Changes (Moore's Law)
  - Unclear Requirements
- More coding, less paper-work
- Higher quality, quicker

But, opponents say....

- No plan, no structure
  - Architecture?
  - Easily derailed
- Inefficient use of developers
  - pair programming
- No documentation
- Unrealistic customer involvement

Today Trend in Agile

" More than 2/3's of all corporate IT organizations will use some form of agile software development process in the next 18 months."

Giga Information Group Inc., 2002

- Cutter Report "Agile vs. Heavy"
- Use is increasing

Agile vs. Non-Agile Methods

AGILE

- Extreme Programming
- Crystal
- SCRUM
- Development Method (DSDM)
- Adaptive Software Development (ASD)
- Feature-Driven Development (FDD)
- Pragmatic Programming

NON-AGILE

- Waterfall
- Spiral
- RUP
- Cleanroom
- Prototyping
- CMM/CMMi
- PSP/TSP

Agile - History

- Started at a meeting on February 11-13 2001 at the Snowbird Ski resort in the Wasatch mountains of Utah, USA

- At the end of this meeting:
  - "Agile 'Software Development' Manifesto"
    - Kent Beck, Alistar Cockburn, etc

Principles of the Agile Manifesto - I

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
- Welcome changing requirements, even late in development. Agile processes harness change for customer's competitive advantage
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale
- Business people and developers must work together daily throughout the project
- Build projects around motivated individuals. Give them the environment and support that they need, and trust them to get the job done
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

Principles of the Agile Manifesto - II

- Working software is the primary measure of progress
- Agile processes promote sustainable development.
- The sponsors, developers, and users should be able to maintain a constant pace indefinitely
- Continuous attention to technical excellence and good design enhances agility
- Simplicity—the art of maximizing the amount of work not done—is essential
- The best architectures, requirements, and designs emerge from self-organizing teams
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts it's behavior accordingly

Agile Value Statements - I

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
  - Team dynamics
    - experience mix, team size
  - Physical workspace, communality, meetings
- Working software over comprehensive documentation
  - Code primary artifact
  - Iterative (subscription)
  - Value to the customer
  - QA inherent

Agile Value Statements - II

- Customer collaboration over contract negotiation
  - Customer Onsite (Involved/Knowledgeable)
  - Requirements Centric
  - Rapid Return of Perceived Value
  - Customer Expectation Management

- Responding to change over following a plan Developer / Customer Team
  - Emergent Requirements
  - Short Iterations
    - Smaller changes
  - Adaptation

Source: http://agilemanifesto.org/

Weaknesses of Agile Methods

- Communication is critical
- Projects with Non- Decomposability / Coupled Functionality
- Scalability
- Reliance on Corporate Knowledge
  - Document at End
- Maintenance
- Long Life Cycle
- Centralized management control
- “Big” Specifications
- Required Documentation
  - Safety Critical
- Non-flexible work environment
- Fixed Price and Scope

Agile Center Piece - Coding

- Emphasizing what we do best
  - What we prefer to do
- Lack of formal design, architecture
- Lack of documentation
  - But makes you think about what is important

Pros vs. Cons of Agile Methods

PROS

- Help organizations deal situations with uncertain requirements, resources, time, and risks
- Build working software quickly

CONS

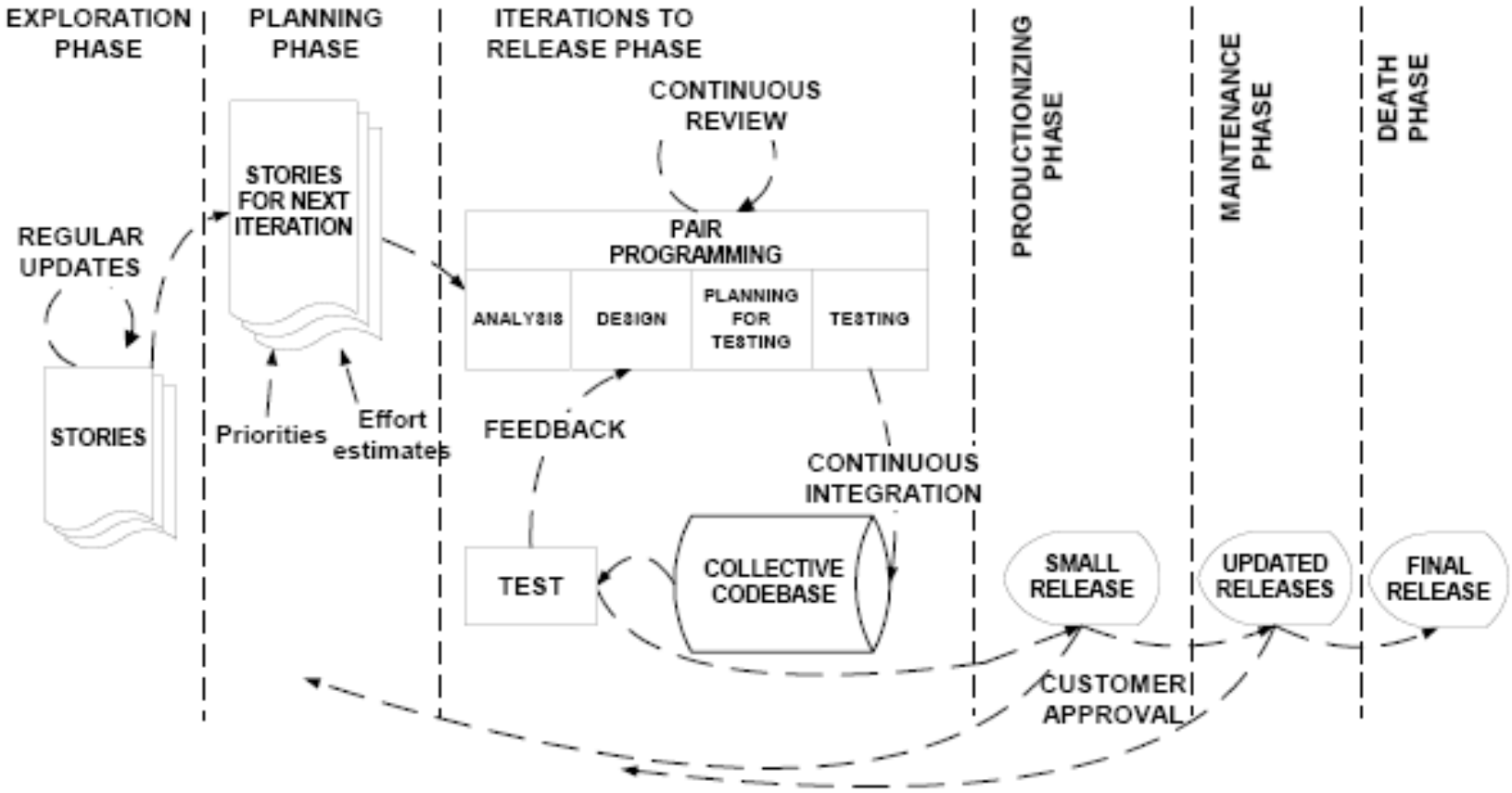
- Must consider the skills of the people involved
- Consider the process and how well they account for your unique needs
- Consider the management and reporting techniques used to control the project

Agile - Application Areas

- Applications that can be built quickly and don't require extensive quality assurance, analysis, design, code, and testing

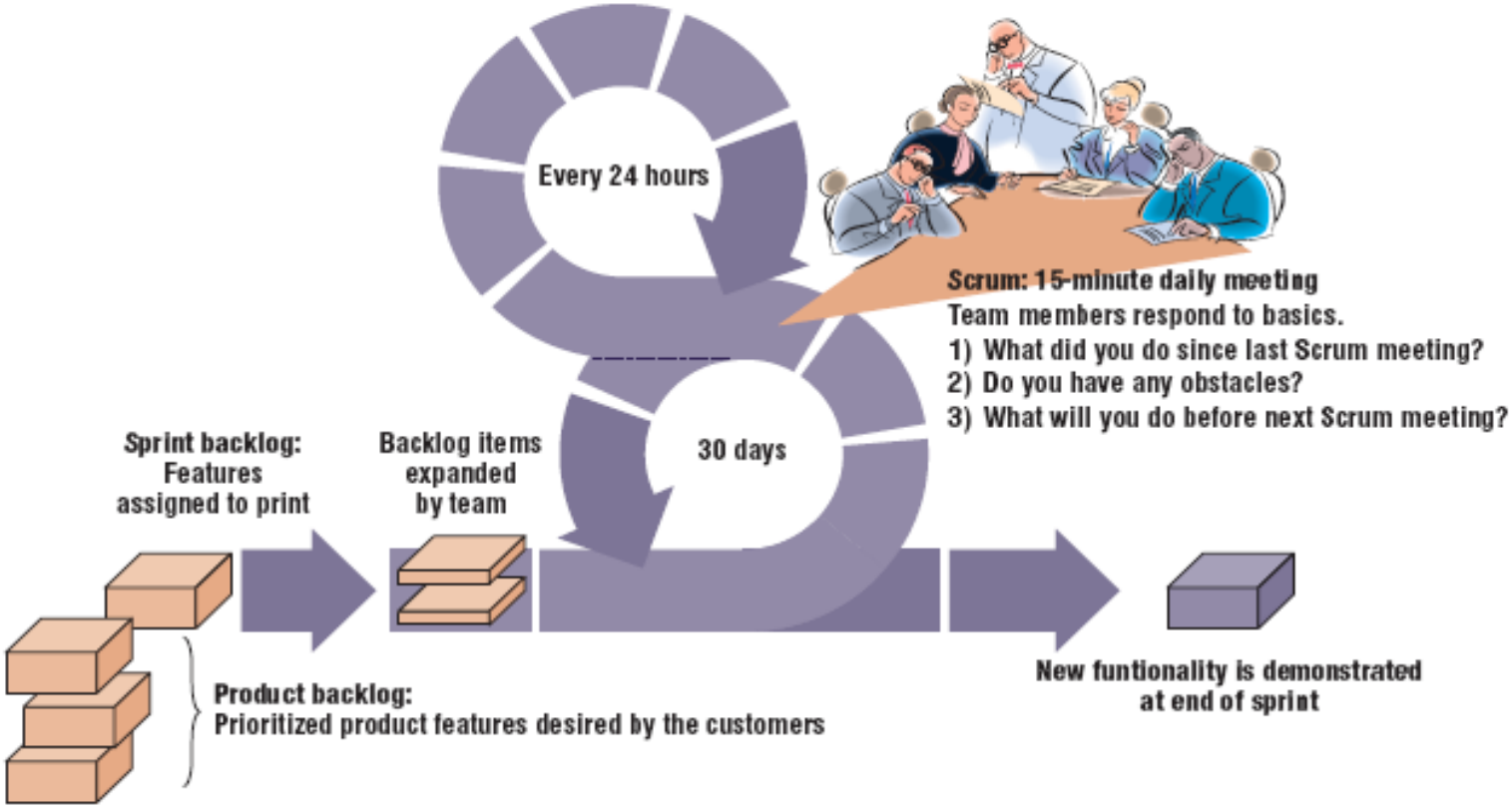
- Small in-house teams developing software for quick to market applications

eXtreme Programming



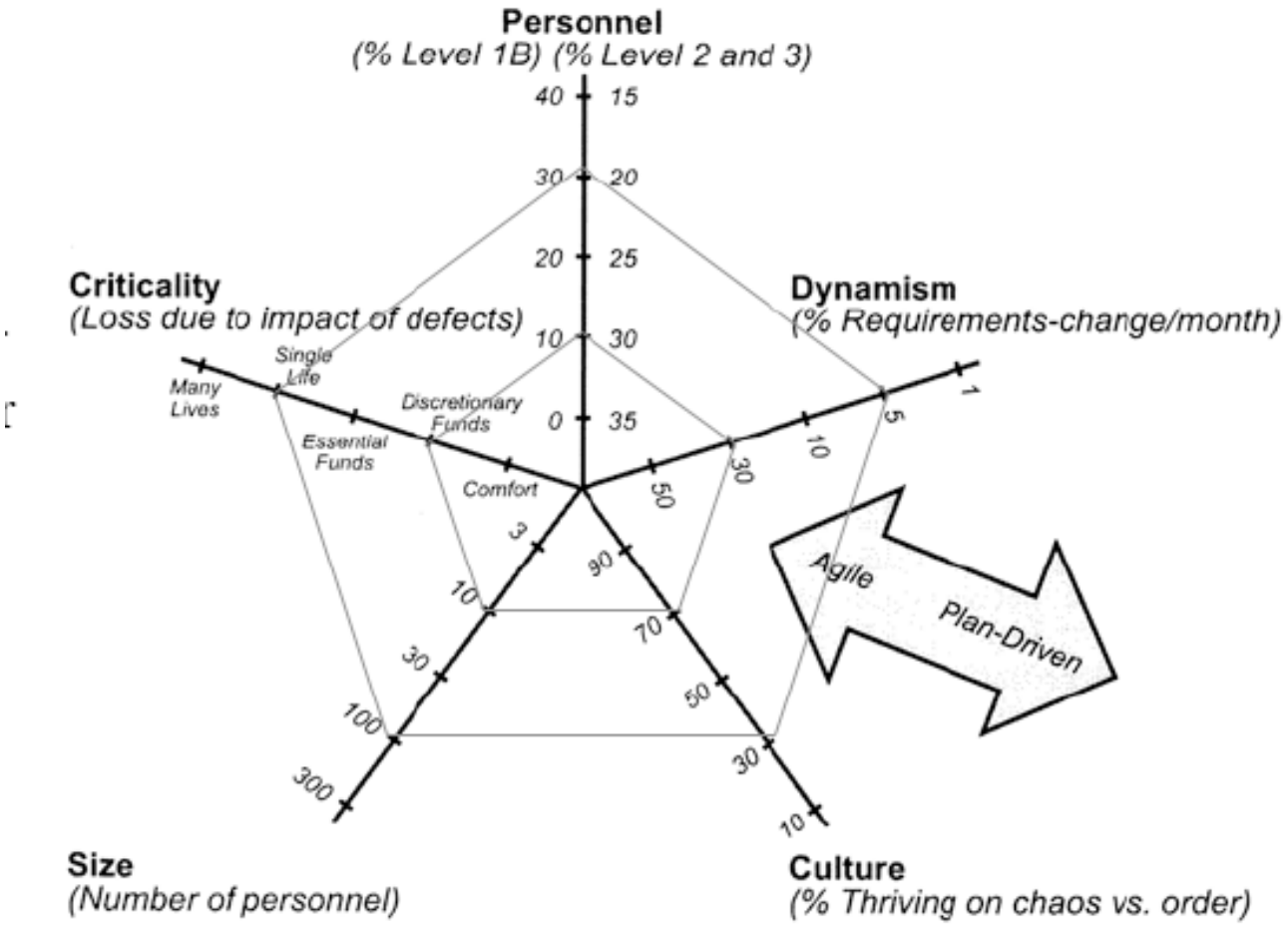
Source: Agile Software Development Methods: Review & Analysis

Scrum Process



Source: IEEE Software, Volume 22, Issue 5, Sept.-Oct. 2005

How to Balance???



Boehm and Turner: "Balancing Agility with Discipline," 2004

Q & A