


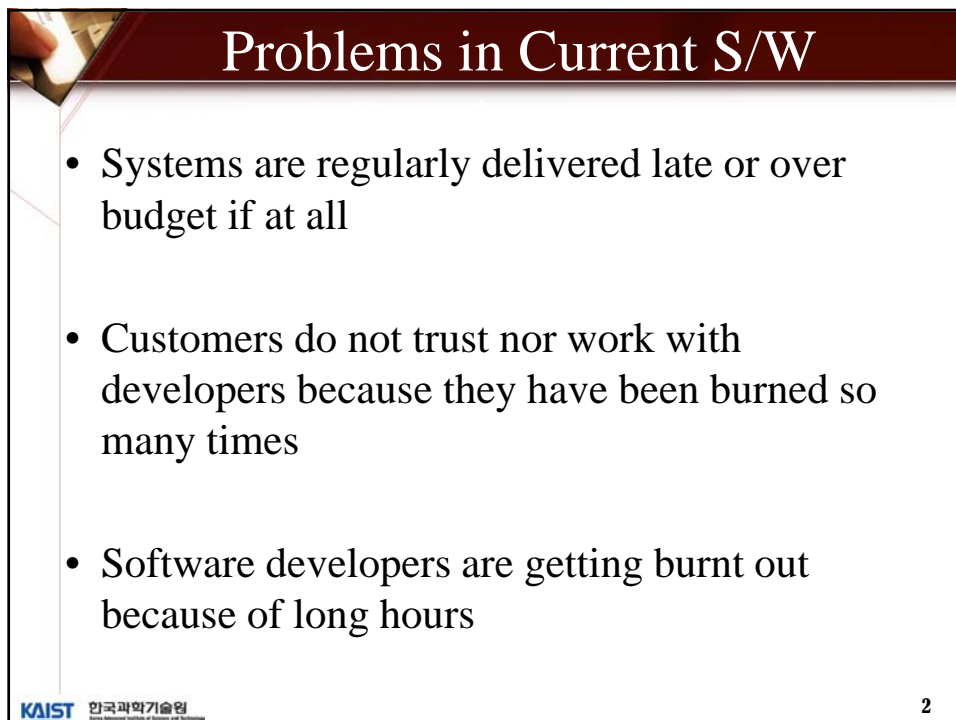
# Software Engineering Economics (CS656)

Agile Methods

Jongmoon Baik




KAIST  
한국과학기술원  
SINCE 1971



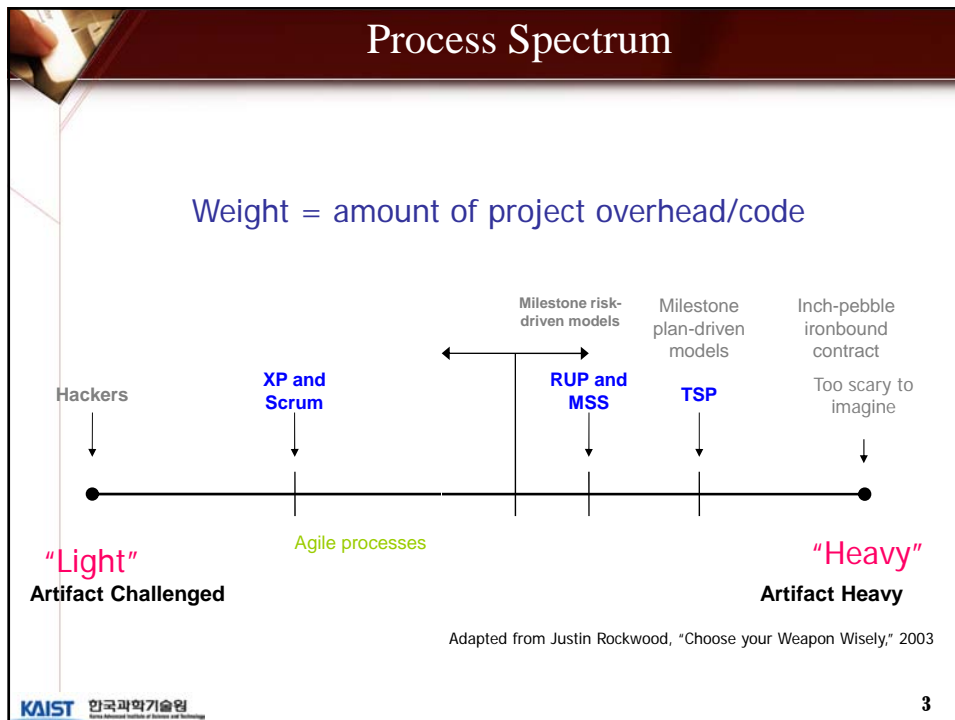
## Problems in Current S/W

- 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



KAIST 한국과학기술원  
SINCE 1971

2



## 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”

KAIST 한국과학기술원  
4

## 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 its behavior accordingly

KAIST 한국과학기술원  
11

## 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

KAIST 한국과학기술원  
12

## 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

KAIST 한국과학기술원 Source: <http://agilemanifesto.org/> 13

## 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

KAIST 한국과학기술원 14

## 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

KAIST 한국과학기술원  
15

## Pros vs. Cons of Agile Methods

PROS	CONS
<ul style="list-style-type: none"><li>• Help organizations deal situations with uncertain requirements, resources, time, and risks</li><li>• Build working software quickly</li></ul>	<ul style="list-style-type: none"><li>• Must consider the skills of the people involved</li><li>• Consider the process and how well they account for your unique needs</li><li>• Consider the management and reporting techniques used to control the project</li></ul>

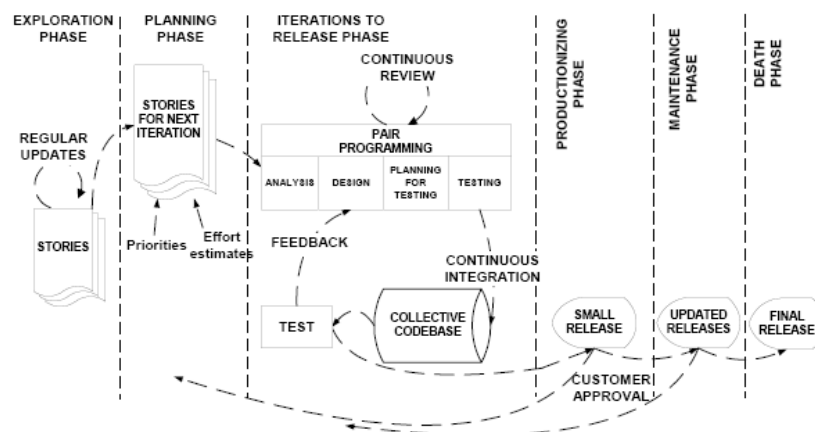
KAIST 한국과학기술원  
16



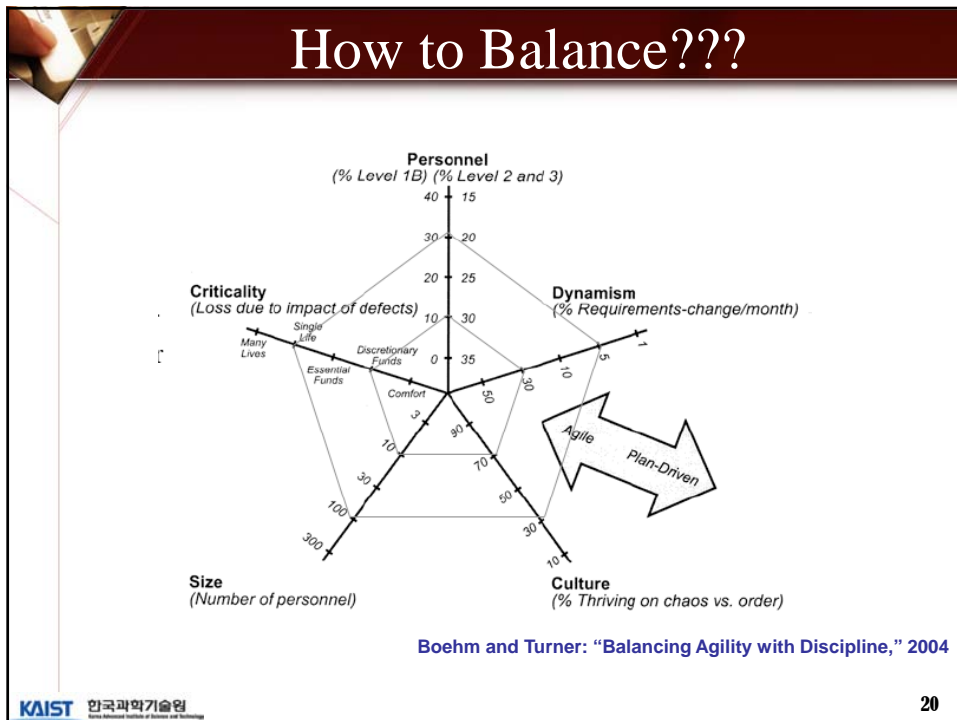
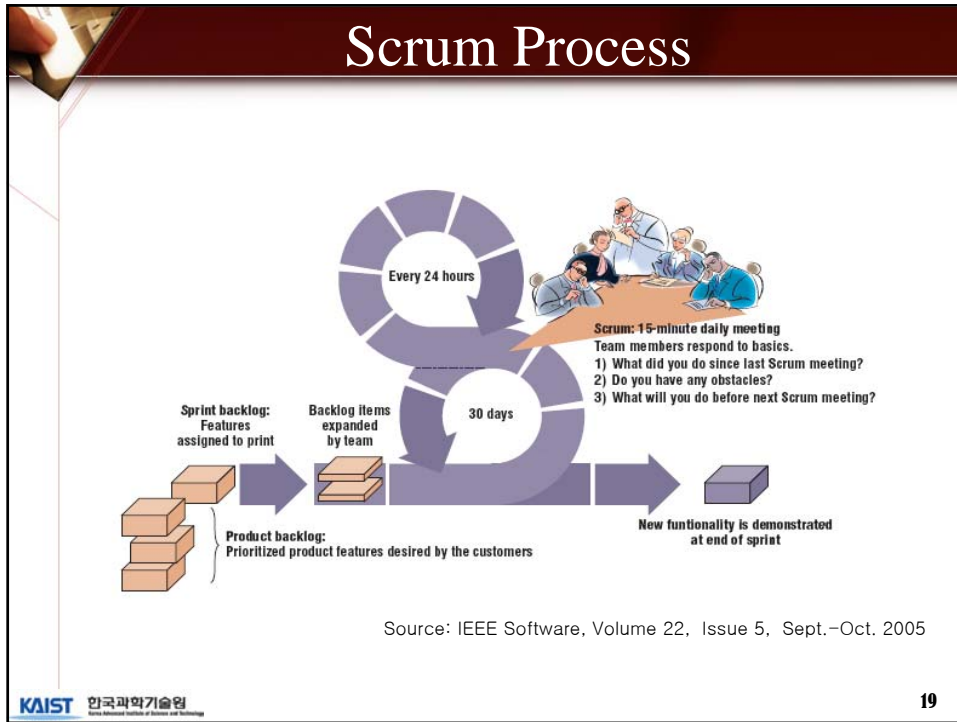
## Agile - Application Areas

- 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 (XP)



Source: Agile Software Development Methods: Review & Analysis



Q & A



KAIST 한국과학기술원  
21