投票 最多可选1项

设置

Are you ready?

A Yes

B No

提交

Software Engineering

Part 1
Software Process

Chapter 5
Agile Development

Reproduced by Ning Li , 2022

Contents

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

Four values

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

The **Agile Manifesto** was written in (?Year) by seventeen independent-minded software practitioners.

A 1999

B 2001

C 2009

提交

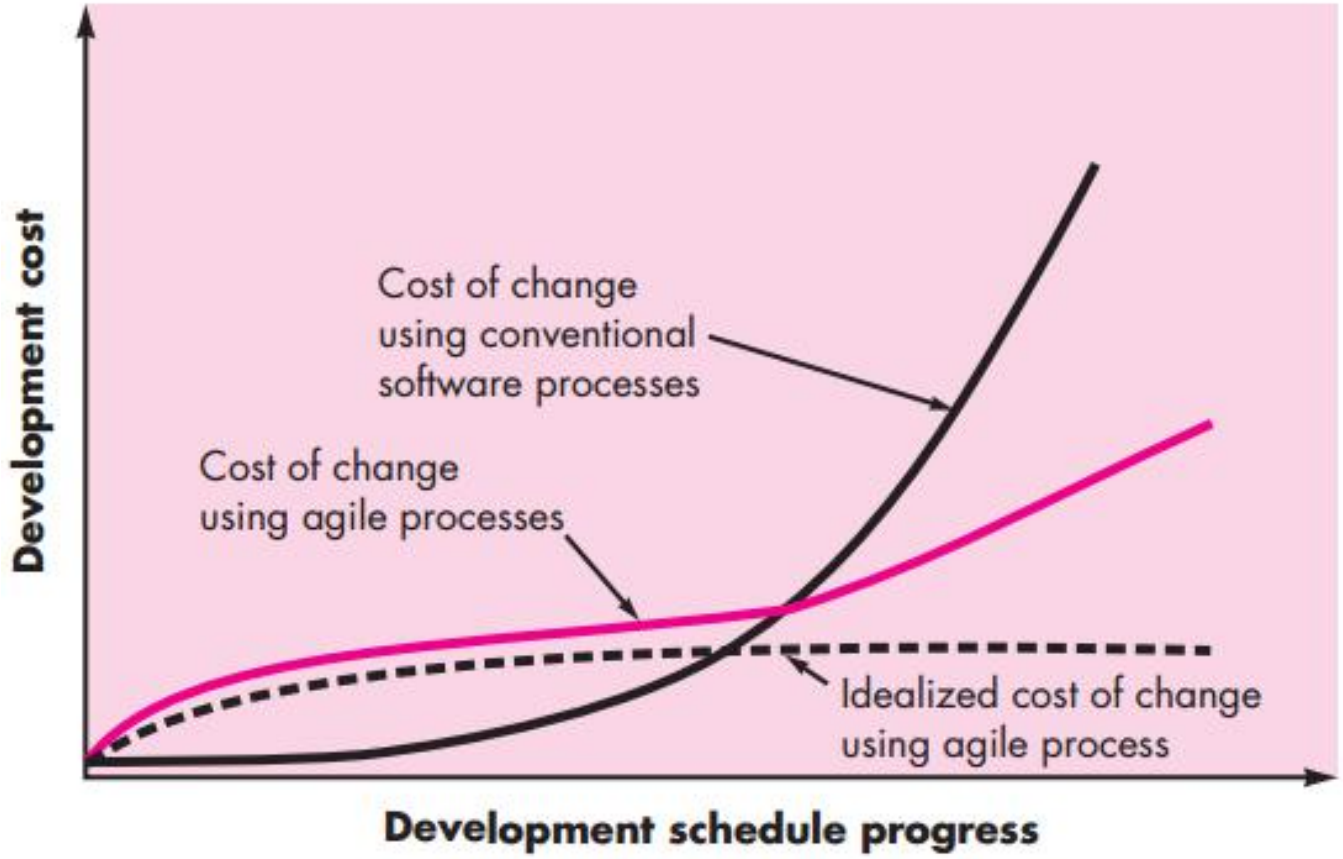
5.1 What is “Agility”?

- 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

5.2 Agility and the Cost of Change



5.1 What is “Agility”?

https://www.bilibili.com/video/av79167229?from=search&seid=3957501565857713048

5.3 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

5.3.1 Agility Principles(12) - 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.
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.

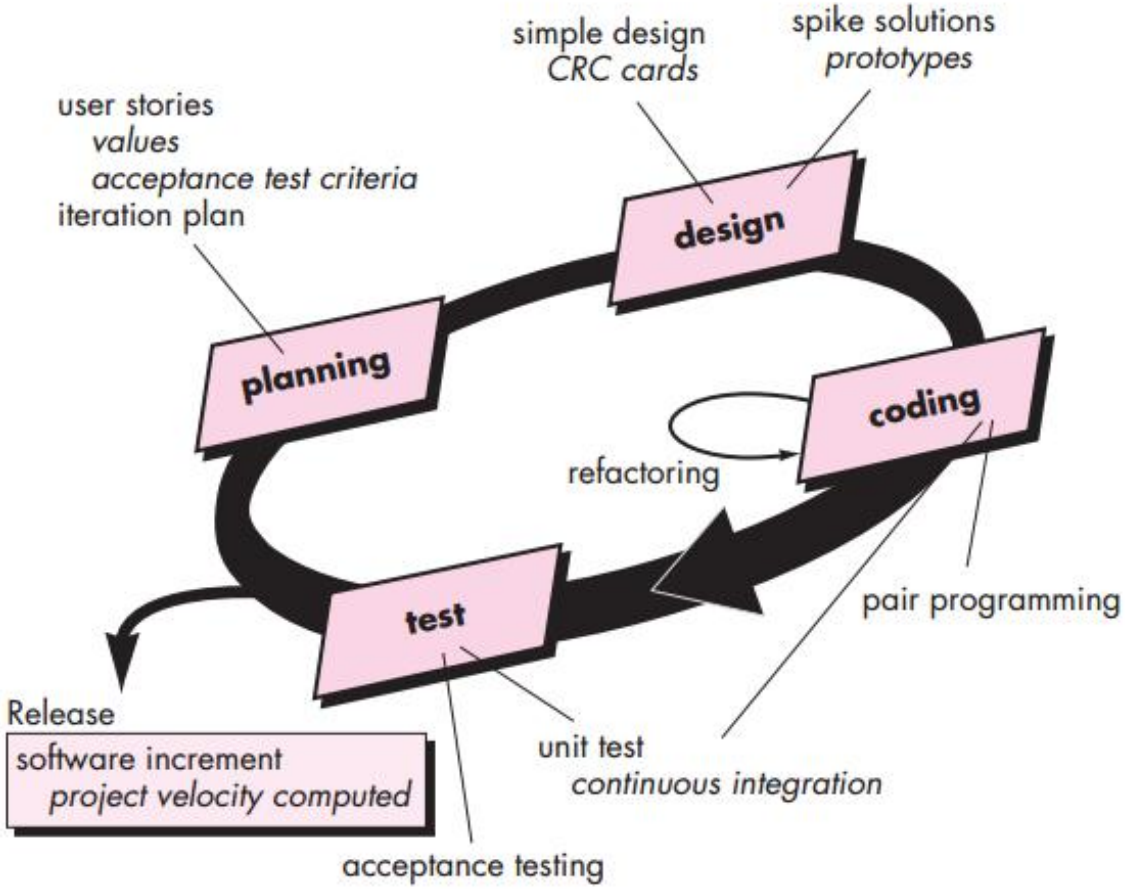
5.3.1 Agility Principles(12) - II

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.
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.

Contents

5.4 Extreme Programming (XP)

The most widely used agile process, originally proposed by Kent Beck



5.4.1 Extreme Programming (XP)

- XP Planning
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment is made on delivery date
  - After the first increment "project velocity" is used to help define subsequent delivery dates for other increments

5.4.1 Extreme Programm P

User stories (example):

As a
registered user

I want to
change my password

So I can
keep my account secure

As an
admin user
I want to
disable a user
So I can
prevent unauthorized logins by past employees

As a
developer
I want to
database with all tables to model the data
So I can
Store information the application needs

5.4.1 Extreme Programming (XP)

- XP Design
  - Follows the KIS principle
  - Encourage the use of CRC(class responsibility collaborator) cards
  - For difficult design problems, suggests the creation of "spike solutions"—a design prototype
  - Encourages "refactoring"—an iterative refinement

- XP Coding
  - Recommends the construction of a unit test for a store *before* coding commences
  - Encourages "pair programming" (Video)

- XP Testing
  - All unit tests are executed daily
  - "Acceptance tests" are defined by the customer and excuted to assess customer visible functionality
  - continuous integration

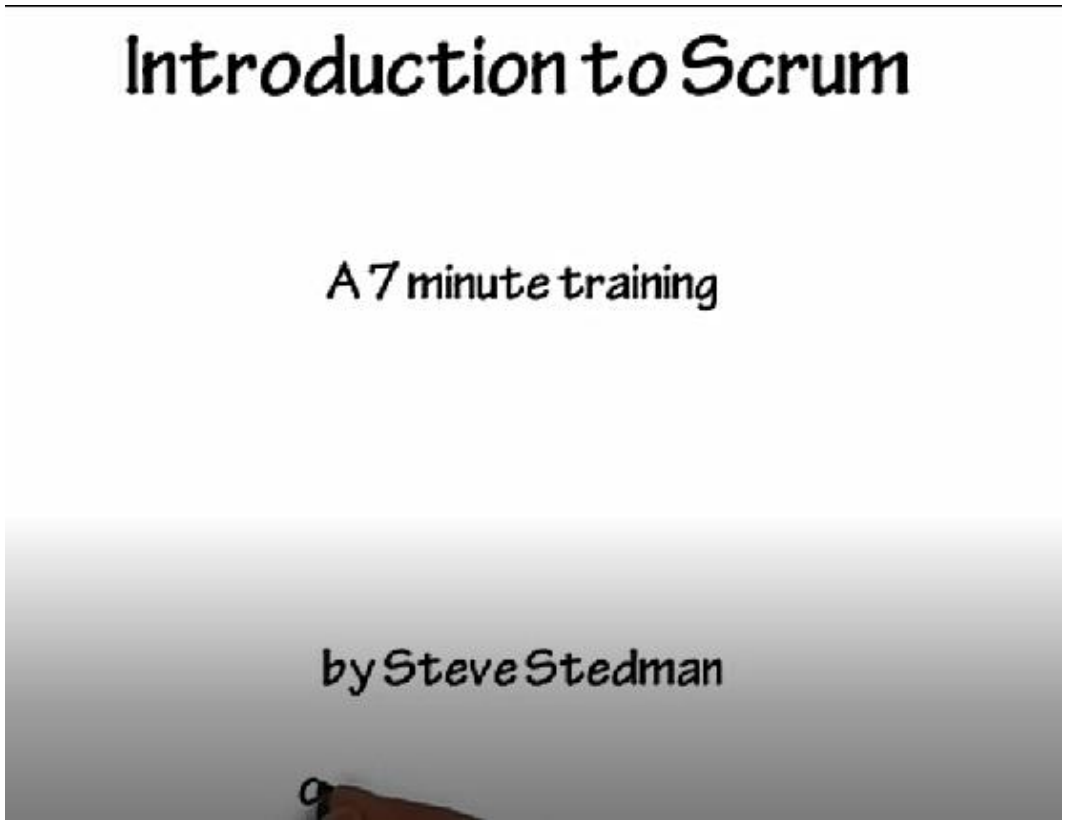
5.4.2 Industrial XP (IXP)

- IXP has greater inclusion of management, expanded customer roles, and upgraded technical practices.

- IXP incorporates six new practices:
  - Readiness assessment (all members on board)
  - Project community
  - Project chartering
  - Test driven management
  - Retrospectives (review)
  - Continuous learning

5.5.1 Scrum

Let's watch a video



https://www.bilibili.com/video/av46426581?from=search&seid=15001626826728777887

5.5.1 Scrum

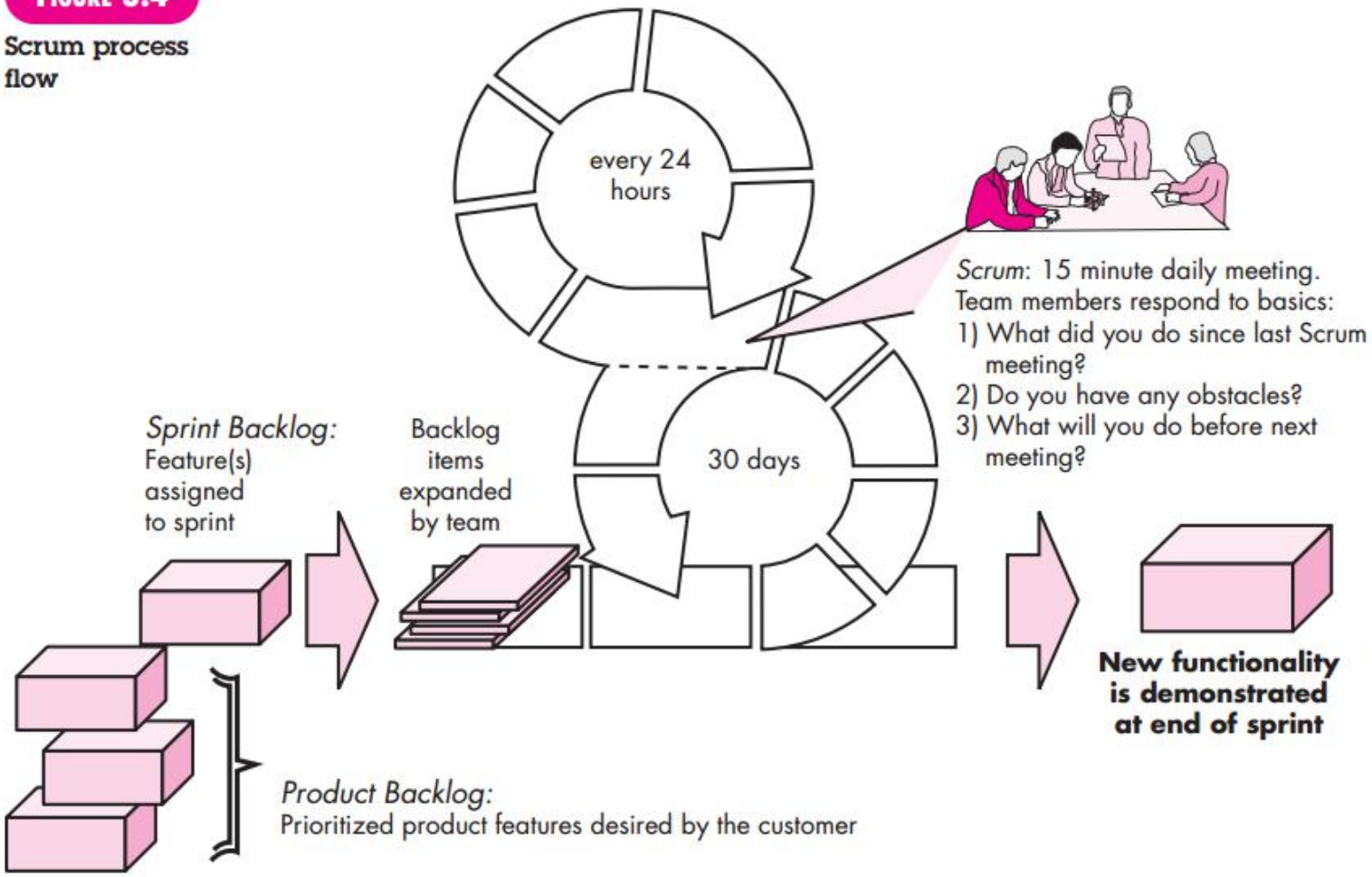
- Originally proposed by Schwaber and Beedle

- 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" (prioritized list of features) of existing requirements
  - Meetings are very short (15 min) and sometimes conducted without chairs
  - "demos" are delivered to the customer with the time-box allocated

5.5.1 Scrum

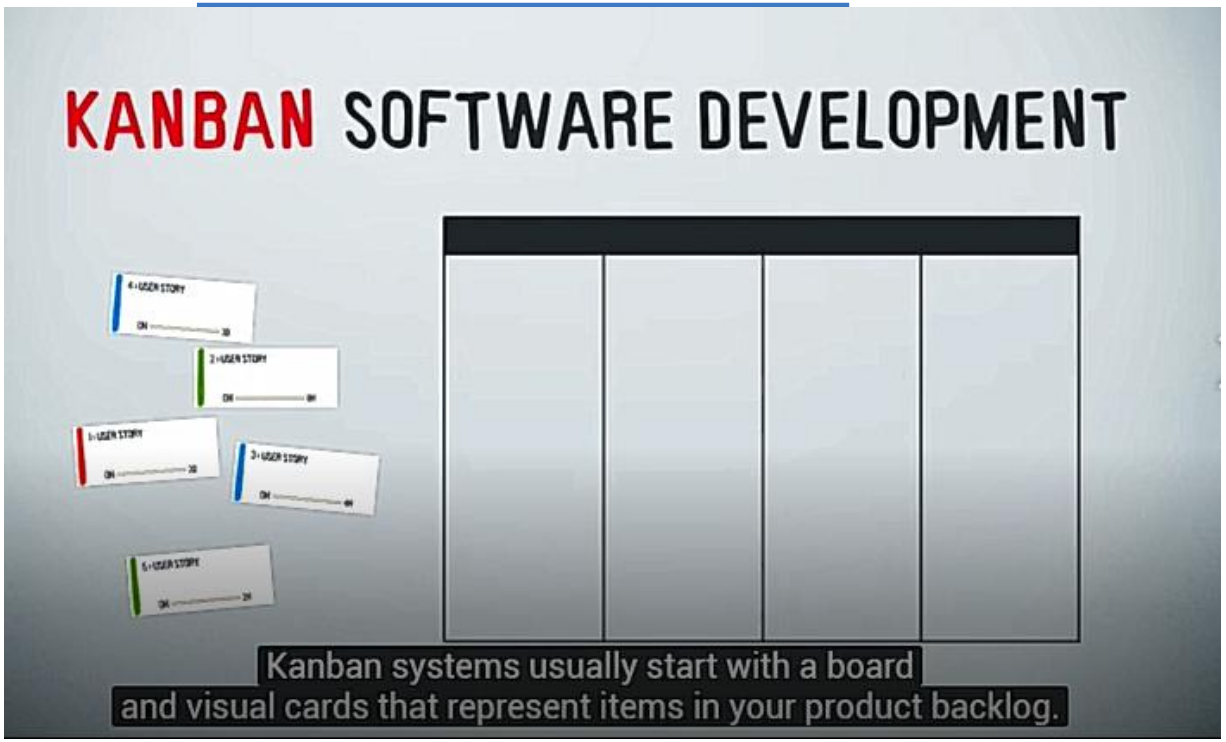
FIGURE 3.4

Scrum process flow



5.5.1 Kanban

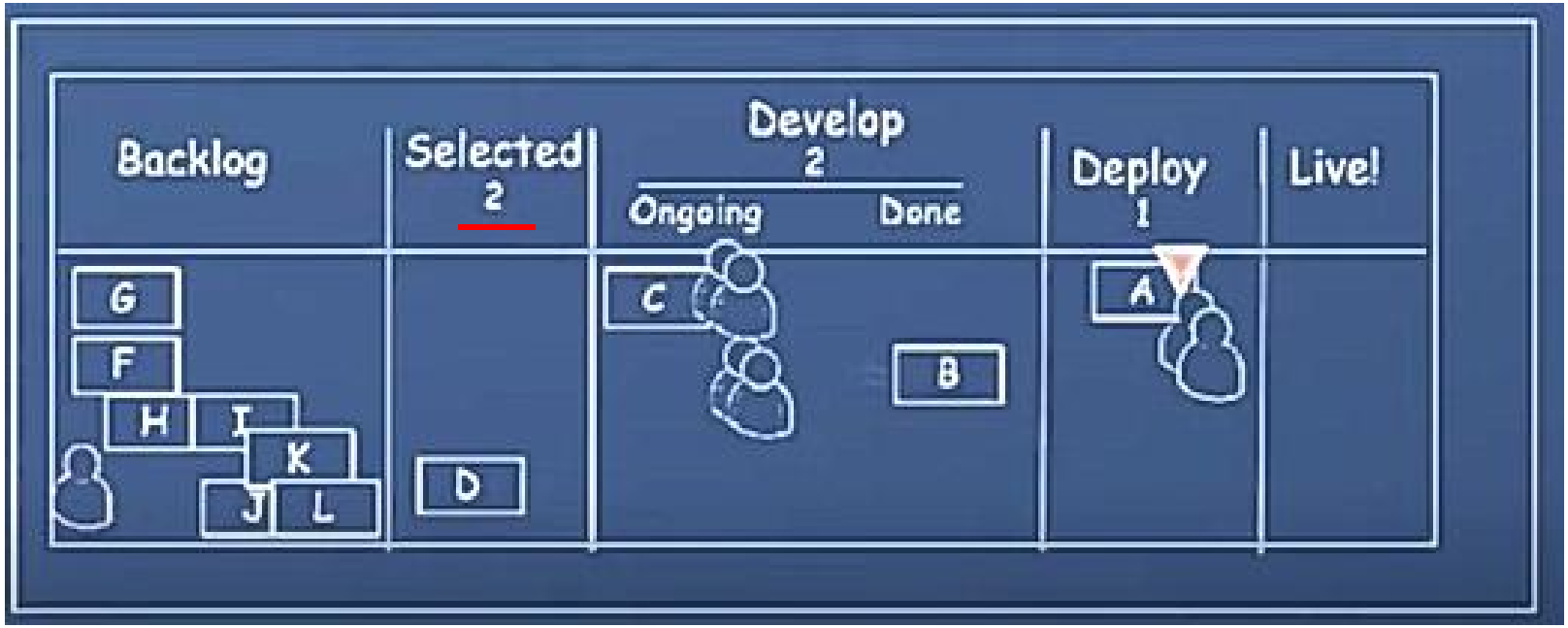
Let's watch a video



https://www.youtube.com/results?search_query=kanban

5.5.1 Kanban

WIP Limit



Deploy: 1 (Max work number in process)

Cited from: DevOps (Nanjing University)
https://www.icourse163.org/learn/NJU-1003664002?tid=1003903001#/learn/content?type=detail&id=1007584279

主观题 10分

设置

What is the difference between scrum and Kanban?

正常使用主观题需2.0以上版本雨课堂

作答

5.5.3 Agile Modeling

- Originally proposed by Scott Ambler

- Suggests a set of agile modeling principles
  - Model with a purpose
  - Use multiple models
  - Travel light
  - Content is more important than representation
  - Know the models and the tools you use to create them
  - Adapt locally

5.5.4 Agile Unified Process

- Each AUP iteration addresses these activities:
  - Modeling
  - Implementation
  - Testing
  - Deployment
  - Configuration and project management
  - Environment management

5.6 A Tool Set for the Agile Process

SOFTWARE TOOLS

Agile Development

Objective: The objective of agile development tools is to assist in one or more aspects of agile development with an emphasis on facilitating the rapid generation of operational software. These tools can also be used when prescriptive process models (Chapter 2) are applied.

Mechanics: Tool mechanics vary. In general, agile tool sets encompass automated support for project planning, use case development and requirements gathering, rapid design, code generation, and testing.

Representative Tools:[18]

Note: Because agile development is a hot topic, most software tools vendors purport to sell tools that support the agile approach. The tools noted here have characteristics that make them particularly useful for agile projects.

OnTime, developed by Axosoft (**www.axosoft.com**), provides agile process management support for various technical activities within the process.

Ideogramic UML, developed by Ideogramic (**www.ideogramic.com**) is a UML tool set specifically developed for use within an agile process.

Together Tool Set, distributed by Borland (**www.borland.com**), provides a tools suite that supports many technical activities within XP and other agile processes.

Practice for you

Find some agile tool and try to use.

Share your tool with us

THE END