Introduction of Software Engineering

Chapter 4:

Agile Development

VŨ THỊ TRÀ

CONTENTS

Agile Development

1. What is it?
   - ✓ Agile software engineering combines
     - A philosophy encourages
       - customer satisfaction
       - early incremental delivery of software;
       - small, highly motivated project teams;
       - informal methods;
       - minimal software engineering work products;
       - overall development simplicity.
     - A set of development guidelines stress
       - delivery over analysis and design
       - active and continuous communication between developers and customers

Agile Development

2. Who does it?
   - ✓ Software engineers
   - ✓ Project stakeholders (managers, customers, end users)

   … work together on *an agile team*.

A team that is sell-organizing and in control of its own destiny.

→ An agile team fosters communication and collaboration among all members

Agile Development

3. Why is it important?
   - ✓ The modern business environment that spawns computer-based systems and software products is last-paced and ever-changing.
   - ✓ Agile software engineering represents a reasonable alternative to conventional software engineering.
4. What are steps?
   - ✓ The five basic activities still remain, but they morph into a minimal task set that push the project team toward construction and delivery.

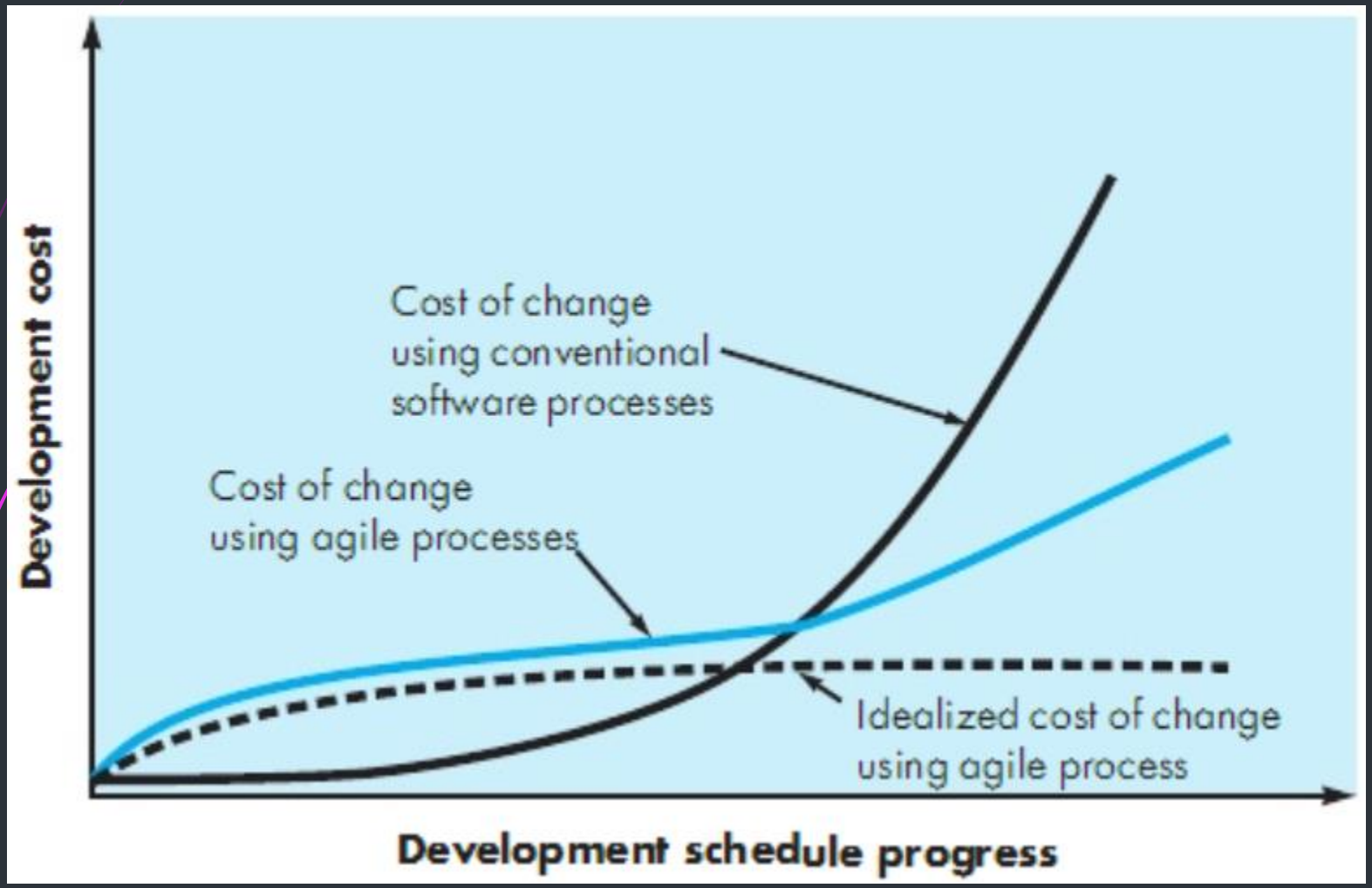
Agile Development

5. What are the work product?
   - ✓ the incremental packages of products.
6. How do I ensure that I've done it right?
   - ✓ If the agile team agrees that the process works, and the team produces deliverable software increments that satisfy the customer, you've done it right.

Agile Development

- *Don't make the mistake assuming that agile gived you license to seek out solutions. A process is required and discipline is essential.*
- Agility is dynamic, content specific, aggressively change embracing and growth oriented.
- An agile process reduces the cost of change because software is released in increments and change can be better controlled within an increment.

Change cost


Development cost
Cost of change
using conventional
software processes
Cost of change
using agile processes
Idealized cost of change
using agile process
Development schedule progress

Agile Princciples

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.

Agile Princciples

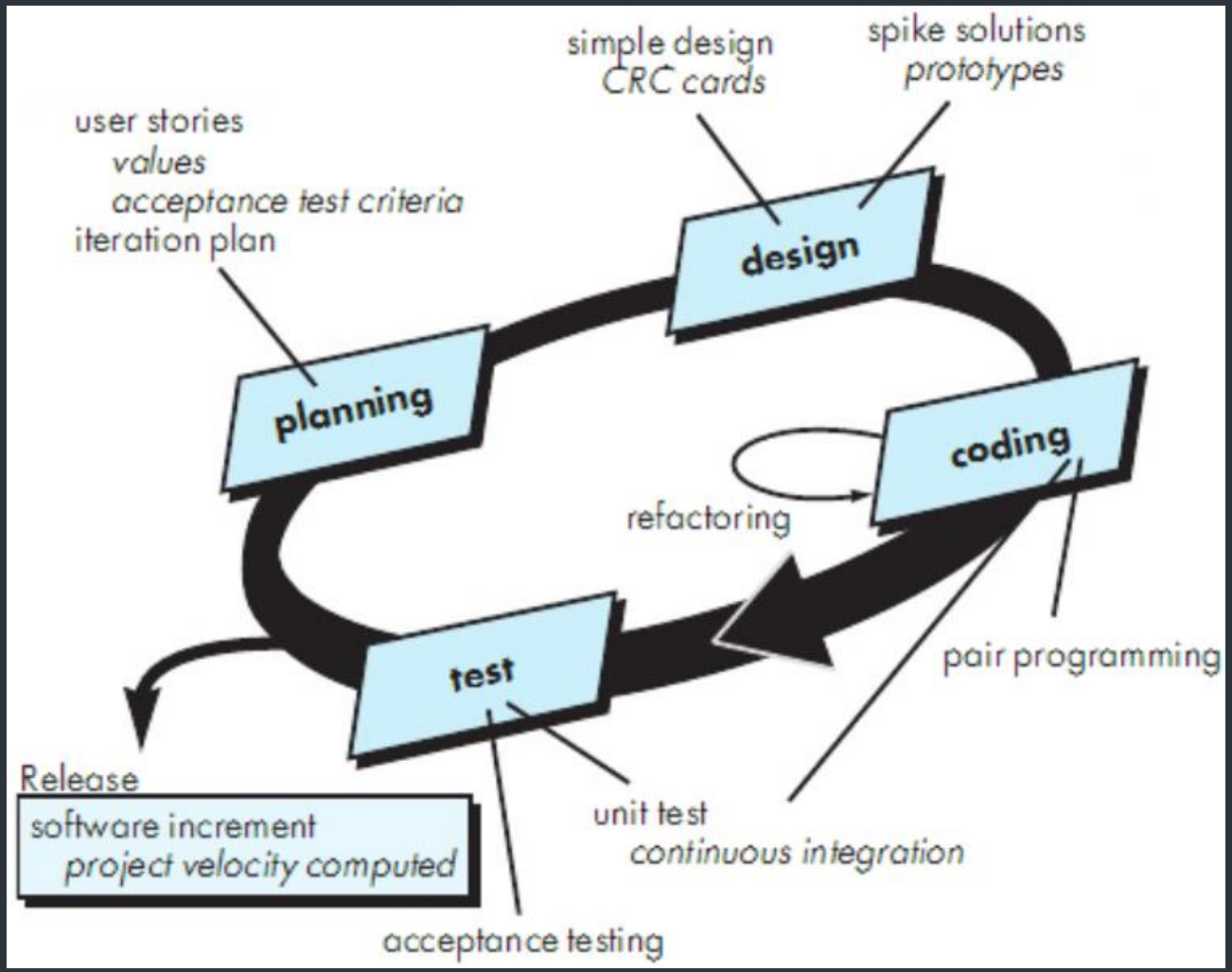
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Agile Princciples

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

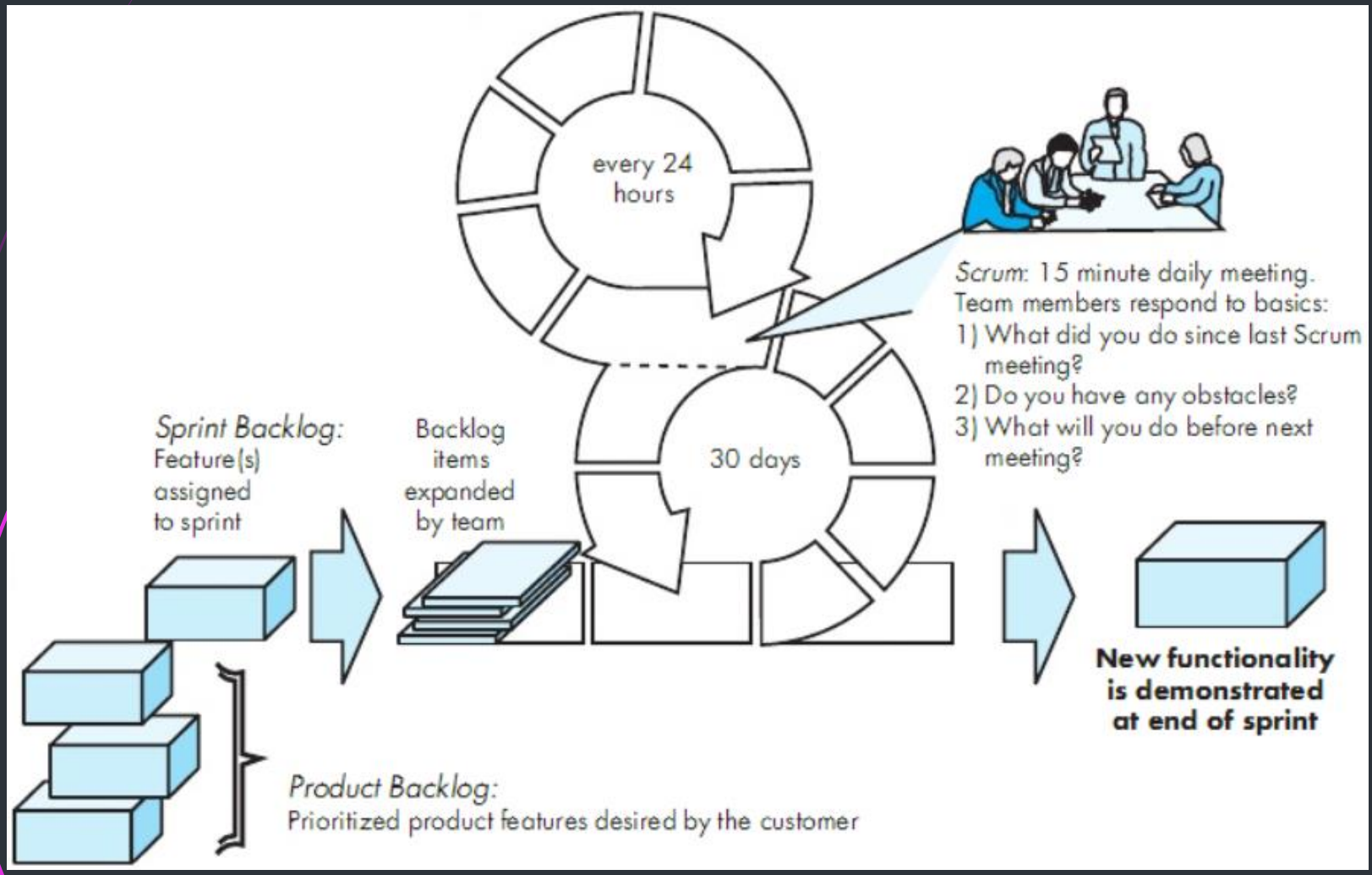
The Extreme Programming Process



CONTENTS

- Agile Development
- XP Process
- **Scrum Process**

Scrum Process Flow



Scrum Process

- Scrum principles are consistent with the agile manifesto and are used to guide development activities within a process that incorporates the following framework activities: requirements, analysis, design, evolution. and delivery.
- Within each framework activity, work tasks occur within a process pattern (discussed in the following paragraph) called *a sprint*.
- The work conducted within a sprint (*the number of sprints required for each framework activity will vary depending on product complexity and size*) is adapted to the problem at hand and is defined and often modified in real time by the Scrum team

In summary

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

<Roger S. Pressman & Bruce R. Maxim>