

Lecture02: Software Processes (I)

EGCI341: SOFTWARE ENGINEERING

Still remember?

- What are the differences between Functional and Non-functional properties?
- What are the attributes of good software?
 - Maintainability, Dependability, Efficiency, and Acceptability
- What are 4 activities of software process?
 - Specification, Development, Validation and Evolution
- What is CASE Tools?

Outline

- Process Framework
- Model Integration
- Process Patterns
- Process Assessment
- Personal and Team Models
- Process Models

Process Framework

A process framework establishes the foundation for a complete software process by

“Identifying a small number of framework activities that are applicable to all software projects”

Generic Process Framework

- Communication
- Planning
- Modeling
- Construction
- Deployment

Capability Maturity Model Integration (CMMI)

Software Engineering Institute (SEI)

- Developed a comprehensive process meta-model
- CMMI model is relied on a set of system and software engineering capabilities
- CMMI model presents the different levels of process capability and maturity

Capability Maturity Model Integration (CMMI) (cont.)

To achieve these capabilities,

- SEI demands that an organization should develop a process model
- This model conforms to (follows) The Capability Maturity Model Integration (CMMI) guideline [CMM02]

Capability Maturity Model Integration (cont.)

The CMMI represents a process meta-model in two different ways:

1. As a **continuous** model
2. As a **stage** model

For a continuous model:

Each process area is formally assessed against specific goals and practices and is rated according to the following **capability levels**:

- ▶ Level 0: Incomplete
- ▶ Level 1: Performed
- ▶ Level 2: Managed
- ▶ Level 3: Defined
- ▶ Level 4: Quantitatively Managed
- ▶ Level 5: Optimized

Capability Maturity Model Integration (Cont.)

CMMI defined a set of generic goals and related practices

Each of generic goals corresponds to one of the **five capability levels**

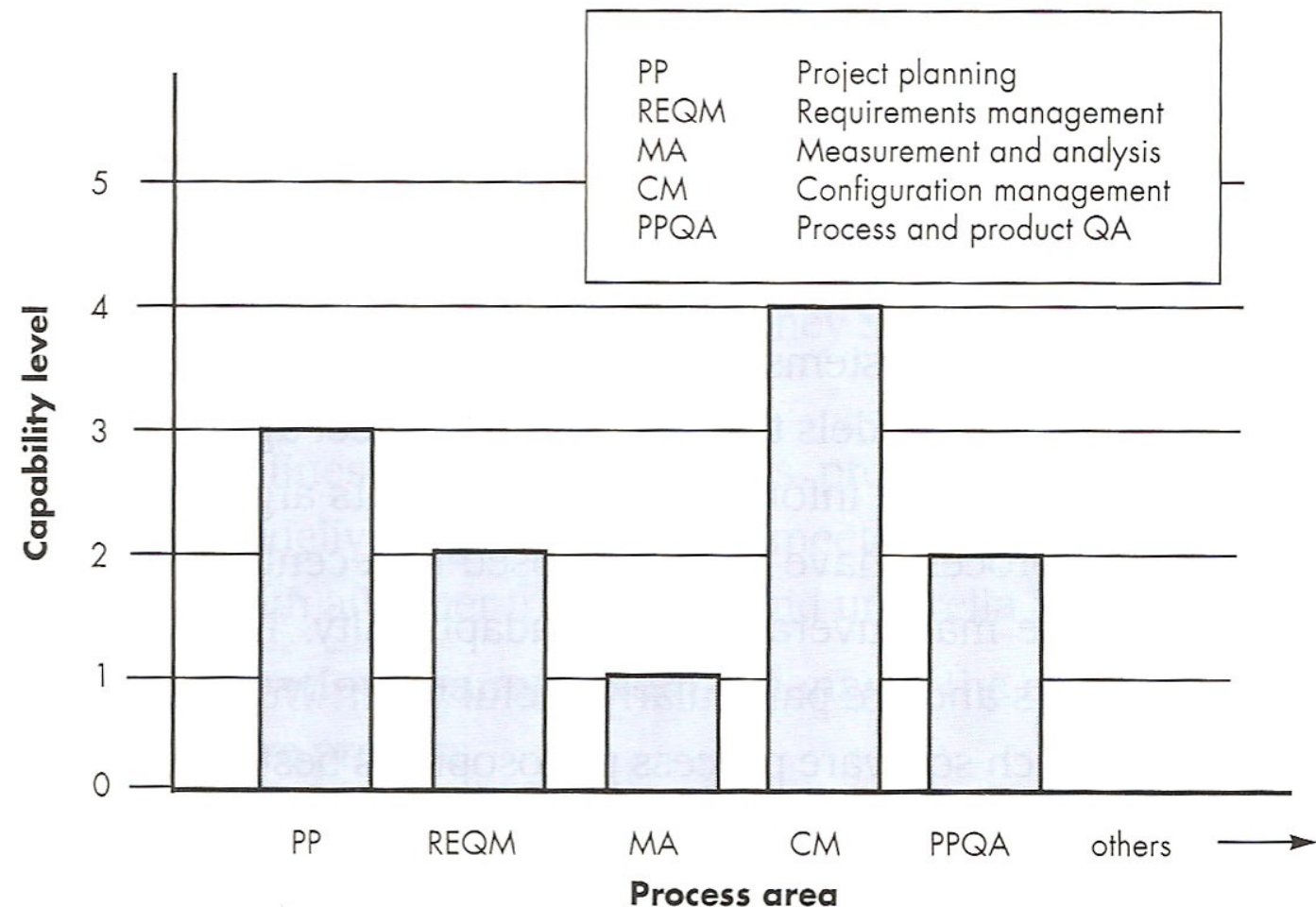
To achieve a particular capability level:

- The generic goal for that level and the generic practices that correspond to that goal must be achieved

Capability Maturity Model Integration (Cont.) [1]

The continuous CMMI meta-model describes a process in two dimensions as illustrated in this figure

0: Incomplete
1: Performed
2: Managed
3: Defined
4: Quantitatively
Managed
5: Optimized



Scenario

1. The development team starts the project right away by writing the codes. When they need more information, they contact the customer/client to gather/clarify some requirements.
2. The development team uses Git to help them back up and manage their project's codes.

Capability Maturity Model Integration(Cont.)

For a staged model:

The staged CMMI model defines:

- Same process areas, Goals, and Practices as the continuous model
- The primary difference is that the staged model defines:
 - ▶ Five maturity levels, rather than five capability levels

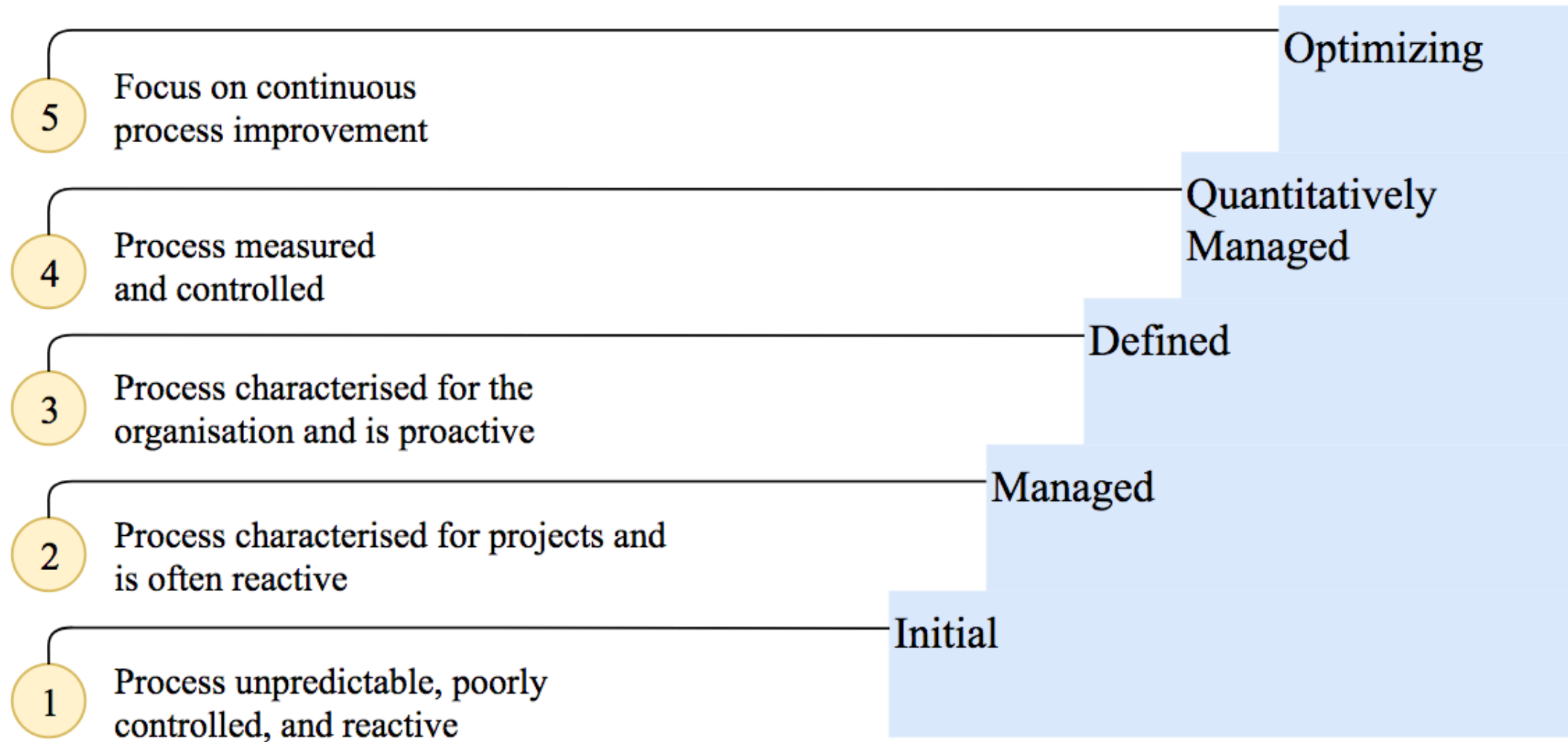
To achieve a maturity level

- The specific goals and practices associated with a set of process areas must be achieved

Maturity Levels [1]

Level	Focus	Process Areas
Optimizing	<i>Continuous process improvement</i>	Organizational Innovation and Deployment Causal Analysis and Resolution
Quantitatively managed	<i>Quantitative management</i>	Organizational Process Performance Quantitative Project Management
Defined	<i>Process standardization</i>	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Integrated Supplier Management Risk Management Decision Analysis and Resolution Organizational Environment for Integration Integrated Teaming
Managed	<i>Basic project management</i>	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
Performed		

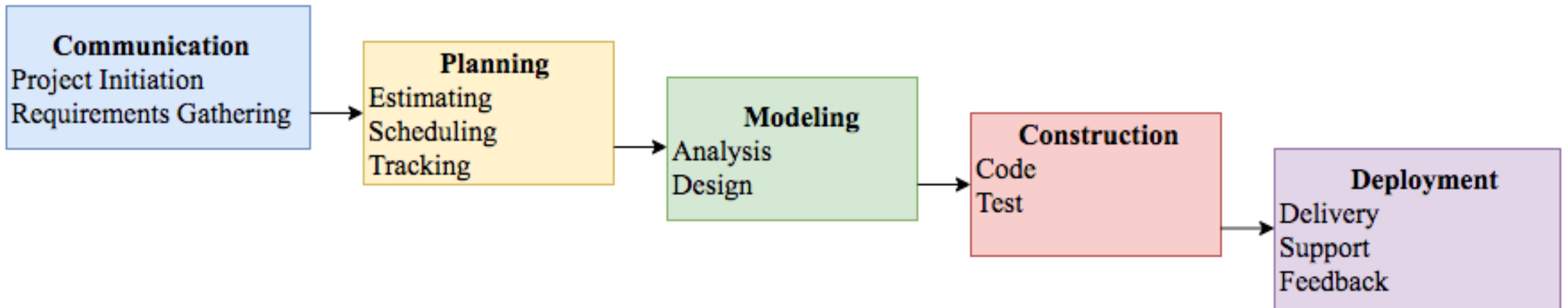
Maturity Level



Process Models

- A structured set of activities required to develop a software system
- A software process model is an abstract representation of a process
 - It presents a description of a process from some particular perspective

Waterfall Model (Cont.)



What is Agility?

Agile software engineering combines:

- A philosophy and a set of development guidelines

The philosophy encourages customer satisfaction and early incremental delivery of

- Software
- Small, highly motivated project team
- Minimal software engineering work products
- Overall development simplicity

The development guidelines stress delivery over

- Analysis and design
- Active and continuous [communication](#) between developers and customers

What is Agile Process?

3 key assumptions about the majority of software projects:

1. 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 a project proceeds
1. For many types of software, design and construction are interleaved
 - That is both activities should be performed in sequence 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.
2. Analysis, design, construction, and testing are not as predictable as we might like

Agile Method (AM)

- Agile Method (AM) is a practice-based methodology for effective modeling and documentation of software-based systems.
- This software that can be applied on a software development project in an effective and light-weight manner.
- Agile models are more effective than traditional models because they are just barely good, they do not have to be perfect.

Component-based software engineering (CBSE)

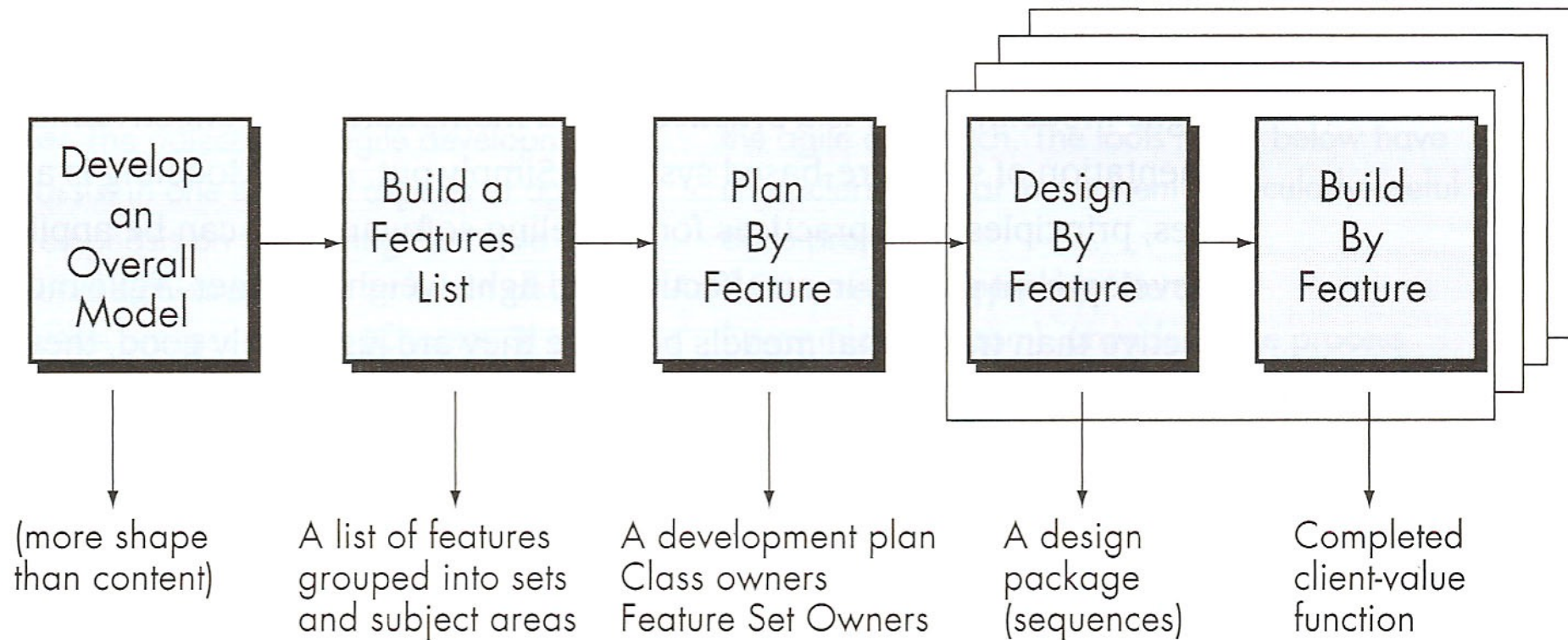
Process stages

- Component analysis
- Requirements modification
- System design with reuse
- Development and integration

This approach is becoming increasingly used as component standards have emerged

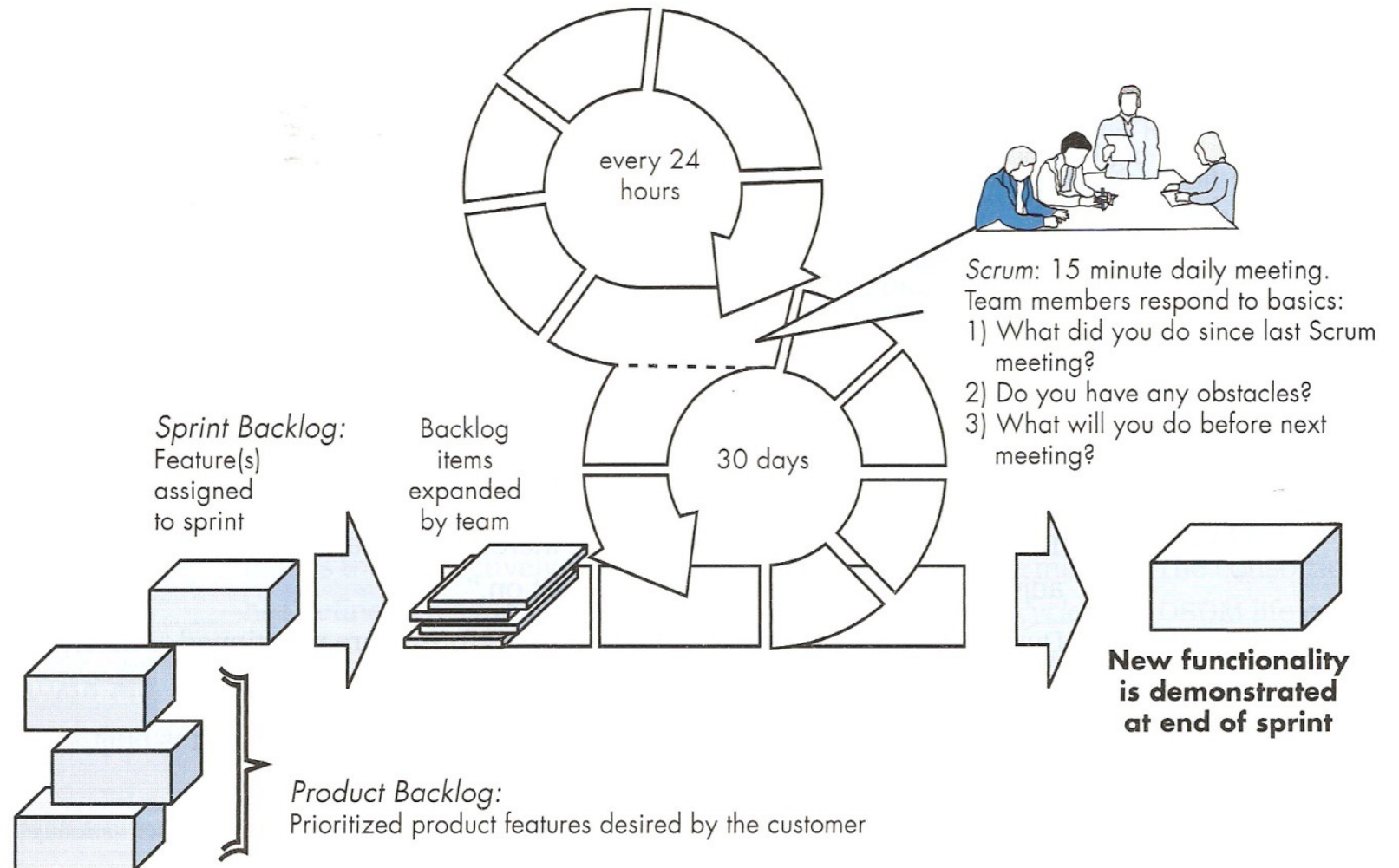
Feature Driven Development (FDD)

Originally conceived by Peter Coad and his colleagues as a practical process model for object-oriented software engineering.



Scrum

Scrum is an agile process model that was developed by Jeff Sutherland and his team in the early 1990s.



References

1. Ian Sommerville, Software Engineering 10th Edition, Pearson, April 2015

Wait! It's not done yet...

Any Questions?

:O)

Thank you

Review

- What is the purpose of CMMI?
- What is the difference between Capability Level and Maturity Level

Group Discussion

7 Groups

- 1) Scrum
- 2) Kanban
- 3) Feature Driven Development (FDD)
- 4) Component-based Development (CBD)
- 5) Lean Software Development
- 6) Unified Process (UP/RUP/AUP)
- 7) Dynamic Systems Development Method (DSDM)

Please explain the list below:

- 1) Is it Agile?
- 2) Explain about the software process life-cycle and How it works?
- 3) An example of the Software or the System that use this kind of software process to develop a software