ICT 2402 Software Engineering

Advanced process models

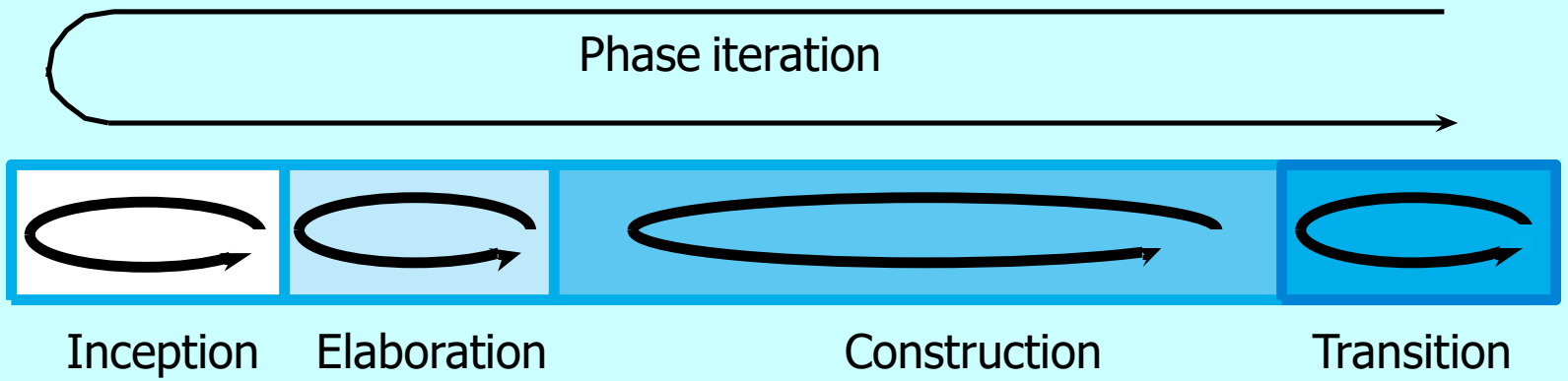
Topics covered

- Rational Unified Process
- Agile processes
  - Extreme Programming
  - Scrum development
- Process improvement

The Rational Unified Process

- A modern process model derived from the Work on the UML and associated process.
- Normally described from 3 perspectives
  - A dynamic perspective that shows phases over time;
  - A static perspective that shows process activities;
  - A practice perspective that suggests good practice.

RUP phase model (dynamic perspective)



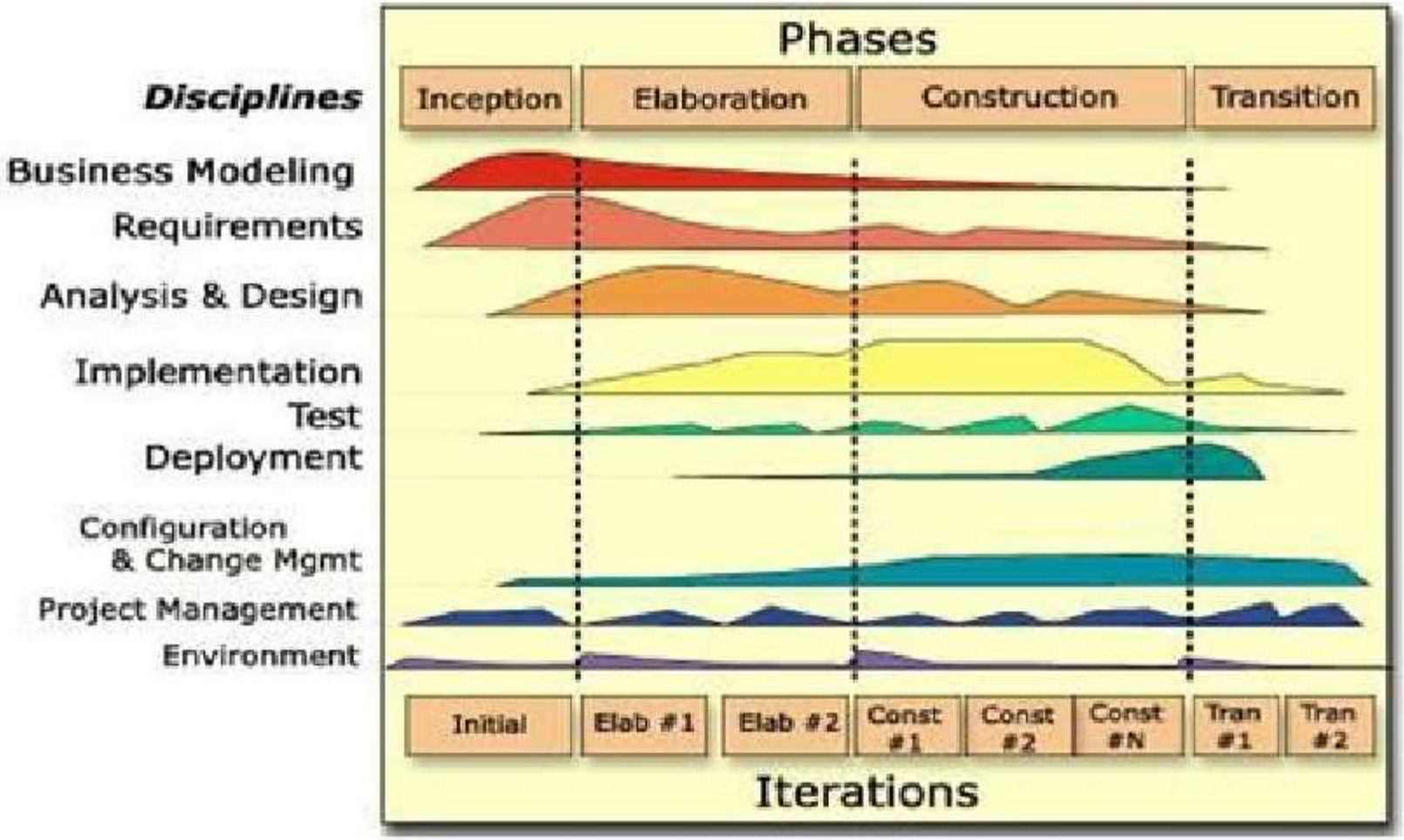
RUP phases

- Inception
  - Establish the business case for the system.
- Elaboration
  - Develop an understanding of the problem domain and The system architecture.
- Construction
  - System design, programming and testing.
- Transition
  - Deploy the system in its operating environment.

Work flows (static perspective)

Workflow	Description
Business modeling	The business processes are modeled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to Model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into Implementation sub-systems. Automatic code generation from design models Helps accelerate this process.
Test	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and Change management	This supporting work flow managed changes to the system.
Project management	This supporting work flow manages the system development.
Environment	This work flow is concerned with making appropriate software tools available to The software development team.

RUP dynamic and static perspectives



RUP good practice

- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Visually model software
- Verify software quality
- Control changes to software

Agile Methods

- Dissatisfaction with the overheads involved in design methods led to the creation of agile methods.These methods:
  - Focus on the code rather than the design
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements
- Agile methods are probably best suited to small/medium-sized business systems or PC products

Principles of Agile Methods

Principle	Description
Customer Involvement	The customer should be closely involved through out the Development process. Their role is provide and prioritize new System requirements and to evaluate the iterations of the system
Incremental Delivery	The software is developed in increments with the customer Specifying the requirements to be included in each increment
People, Not Process	The skills of the development team should be recognized and exploited. The team should be left to develop their own ways of Working without prescriptive processes
Embrace Change	Expect the system requirements to change and design the system So that it can accommodate these changes
Maintain Simplicity	Focus on simplicity in both the software being developed and in the Development process used. Wherever possible, actively work to Eliminate complexity from the system

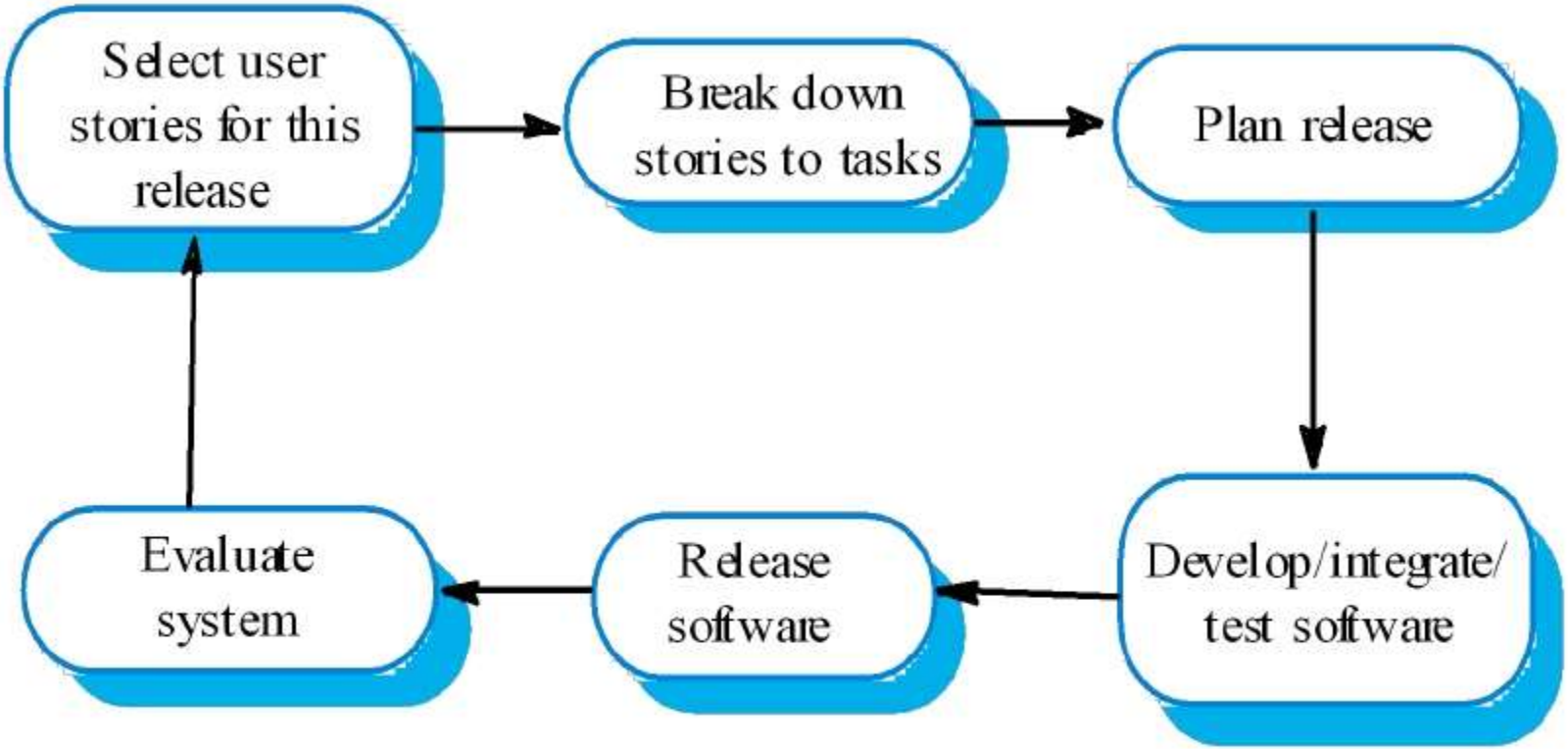
Problems with Agile Methods

- It can be difficult to keep the interest of customers who are Involved in the process
- Team members may be not be able to cope with the Intensity of team involvement needed in agile methods
- Prioritizing changes can be difficult where there are multiple stakeholders
- Maintaining simplicity requires extra work
- Contracts may be a problem as with other approaches to Iterative development

Extreme Programming

- Perhapsthe best-known and most widely used agile method
- Extreme Programming (XP) takes an 'extreme' approach To certain practices of Agile development
  - New versions maybe built several times per day
  - Increments are delivered to customers every 2 weeks
  - Fulltime engagement of the customer

The XP Release Cycle



Extreme Programming Principles 1

Incremental Planning	Requirements are recorded on Story Cards and the Stories to be Included in a release are determined by the time available and their relative priority. The developers break these Stories in to development‘Tasks’
Small Releases	The minimal use full set of functionality that provides business value Is developed first. Releases of the system are frequent and Incrementally add functionality to the first release
Simple Design	Enough design is carried out to meet the current requirements and no more
Test-First Development	An automated unit test framework is used to write tests for a new piece of functionality before the functionality is implemented
Refactoring	All developers are expected to refactor the code continuously as so on as possible code improvements are found. This keeps the code simple and maintainable

Extreme Programming Principles 2

Pair Programming	Developers work in pairs, checking each other's work and providing The support to always do a good job
Collective Ownership	The pairs of developers work on all areas of the system, so that no Islands of expertise develop and all the developers own all the code. Anyone can change anything
Continuous Integration	As soon as work on a task is complete it is integrated into the whole system. After any such integration, all the unit tests in the system must pass
Sustainable Pace	Large amounts of over-time are not considered acceptable as the Net effect is often to reduce code quality and medium term Productivity
On-Site Customer	A representative of the end-user of the system(the Customer) Should be available full time for the use of the XP team. In an Extreme programming process, the customer is a member of the Development team and is responsible for bringing system Requirements to the team for implementation

XP and Agile Principles

- Incremental development is supported through small, frequent system releases
- Customer involvement means full-time customer engagement with the team
- Emphasis on people, not process through pair programming, collective ownership and a process that avoids long working hours
- Change supported through regular system releases
- maintaining simplicity through constant refactoring of code

Requirements Scenarios

- In XP, user requirements are expressed as scenarios or user stories
- These are written on cards and the Development team break them down in to Implementation tasks. These tasks are the Basis of schedule and cost estimates
- The customer chooses the stories for Inclusion in the next release based on their Priorities and the schedule estimates

Story card for Document Downloading

Downloading and printing an article

First, you select the article that you want from a displayed list. You then have to tell the system how you will pay for it - this can either be through a subscription, through a company account or by credit card.

After this, you get a copyright form from the system to fill in and, when you have submitted this, the article you want is downloaded onto your computer.

You then choose a printer and a copy ofthe article is printed. You tell the system if printing has been successful.

If the article is a print-only article, you can keep the PDF version so it is automatically deleted from your computer.

Task Cards for Document Downloading

Task1: Implement principal workflow

Task2: Implement articlecatalog and selection

Task3: Implement payment collection

Payment may be made in 3 different ways. The user selects which way they wish to pay If the user has a library subscription, then they can input the subscriber key which should be checked by the system.Alternatively, they can input an organisational account number. Ifthis is valid, a debit of the cost ofthe article is posted to this account. Finally they may input a 16 digit credit card number and expiry date.This should be checked for validity and, if valid a debit is posted to that credit card account.

XP and Change

- Conventional wisdom in software engineering is to design for change. It is worth spending time and effort Anticipating changes as this reduces costs later in the life cycle.
- XP, however, maintains that this is not worth while as changes cannot be reliably anticipated.
- Rather, it proposes constant code improvement (refactoring) to make changes easier when they have to be implemented

Testing in XP

- Test-first development
- Incremental test development from scenarios
- User involvement in test development and validation
- Automated test harnesses reused to run all Component tests each time that a new Release is built

Test Case Description

Test 4:Test credit card validity

Input:

A string representing the credit card number and two integers representing the month and year when the card expires

Tests:

Check that all bytes in the string are digits

Check that the month lies between 1 and 12 and the year is greater than or equal to the current year.

Using the first 4 digits of the credit card number, check that the card issuer is valid by looking up the card issuer table. Check credit card validity by submitting the card number and expiry date information to the card issuer

Output:

OK or error message indicating that the card is invalid

Test-First Development

- Writing tests before code clarifies the requirements to be implemented
- Tests are written as programs rather than data so that they can be executed automatically.The test includes a check that it has executed correctly
- All previous and new tests are automatically run when new functionality is added, which checks that the new functionality has not introduced errors

Pair Programming

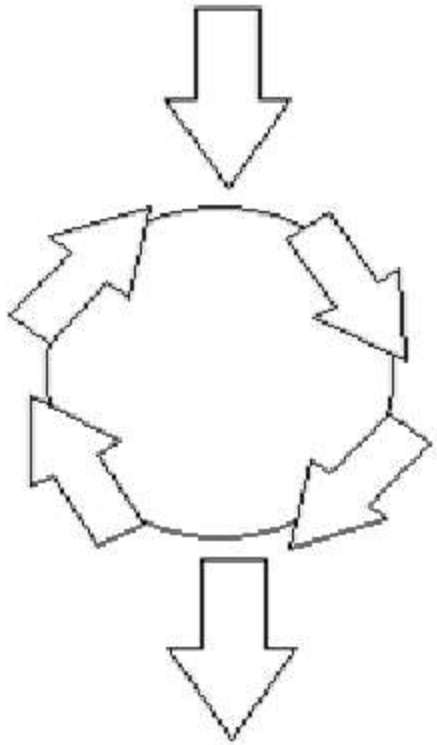
- In XP, programmers work in pairs, sitting together to develop code
- This helps develop common ownershipof code and spreads knowledge across the team
- It serves as an informal review process as each line of code Is looked at by more than 1 person
- It encourages refactoring as the whole team can benefit from this
- Measurements suggest that development productivity with pair programming is similar to that of two people working independently

Scrum development

- Scrum is named after the game of Rugby in which a group is responsible for picking up the ball and moving It forward.
- It is an iterative, incremental process for developing Any product or managing any work.
- Scrum focuses on the entire organization for its implementation to be a success.

Scrum Methodology

- **Pregame**
  - Planning
  - System Architecture/High Level Design
- **Game**
  - Sprints (Concurrent Engineering)
  - Develop (Analysis,Design,Develop)
  - Wrap
  - Review
  - Adjust
- **Postgame**
  - Closure



Scrum processes

Pregame

- PlanningandArchitecture:
  - Identifyproject
  - Prioritizing functional requirements
  - Identifyresources available
  - Establishing the target environment

Scrum processes (cont.)

Game

- Sprints:-lasts for 30 days

  -Analysis, Design,Develop

  -Testing (thishappensthroughoutsprint)

  - Review

  -Adjust

Postgame

- Closure (thisincludes deliveringa functioning deliverable, sign-off, start nextsprint.

Scrum roles

- ScrumMaster
- Developer
- QA
- Documentationmember

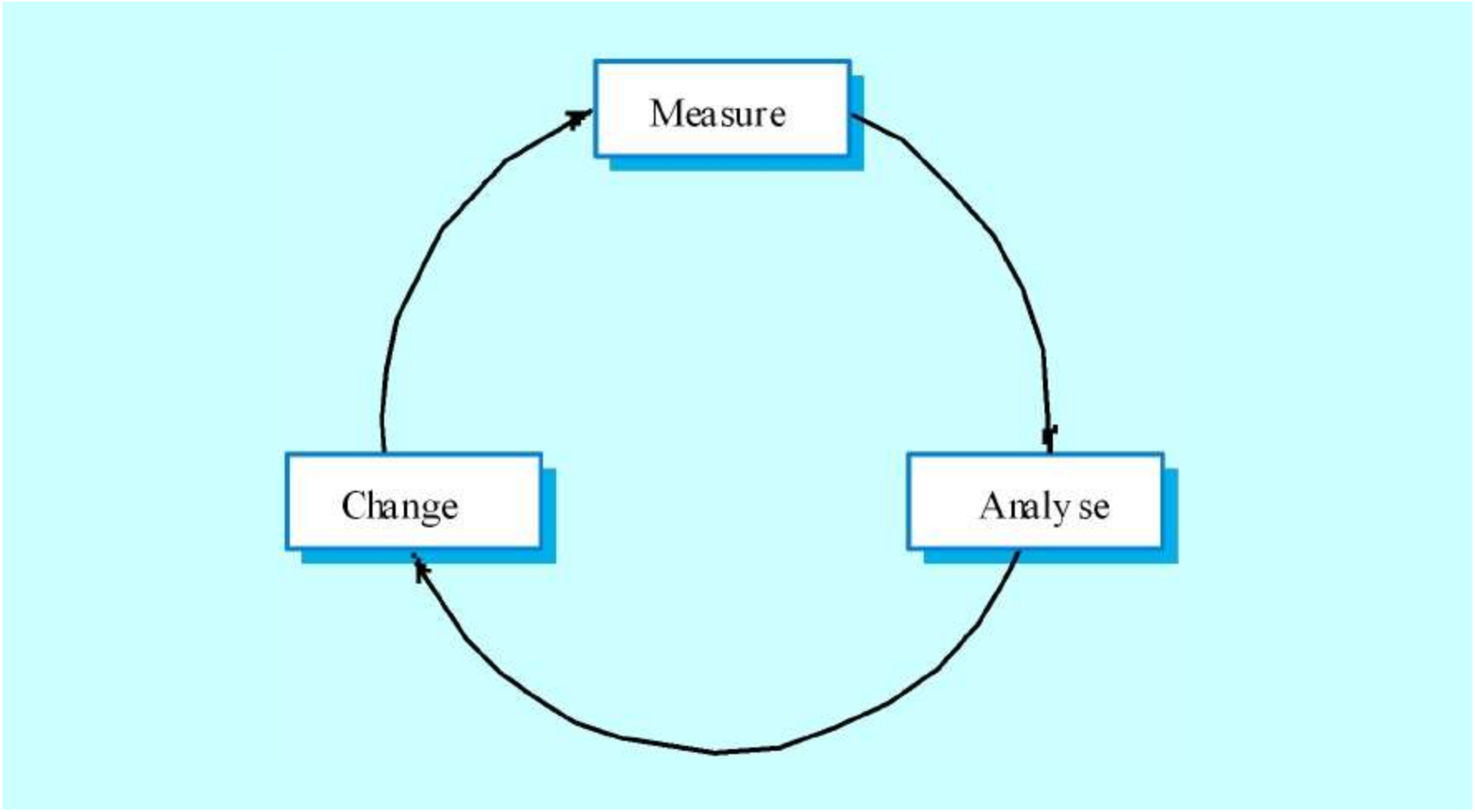
Scrum values

- Flexible deliverable
- Flexible schedule
- Small teams
- Frequentreviews
- Collaboration
- TeamEmpowerment
- Adaptability

Process improvement

- Understanding existing processes and Introducing process changes to improve product quality, reduce costs or accelerate schedules.
- Most process improvement work so far has focused on defect reduction. This reflects the increasing attention paid by industry to quality.
- However, other process attributes can also be the focus of improvement

The process improvement cycle



Process improvement stages

- Processmeasurement
  - Attributes of the current process are measured. These are a baseline for assessing improvements.
- Processanalysis
  - The current process isassessed and bottlenecks and weaknesses are identified.
- Processchange
  - Changes to the process that havebeen identified During the analysis are introduced.

The CMMI framework

- TheCMMI framework is the current stage of work on process assessment and improvement that started at the Software Engineering Institutein the 1980s.
- The SEI's mission is to promote software technology transfer particularly to US defence contractors.
- It has had a profound influence on process improvement
  - CapabilityMaturity Model introducedin the early 1990s.
  - Revisedmaturity framework (CMMI)introducedin 2001.

CMMI model components

- Process areas
  - 24 process areas that are relevant to process capability and Improvement are identified.These are organized in to4 groups.
- Goals
  - Goals are descriptions of desirable organisational states.Each process area has associated goals.
- Practices
  - Practices are ways of achieving a goal -however, they are advisory and other approaches to achieve the goal may be used.

CMMI process areas 1

Process management	Organisational process definition
	Organisational process focus
	Organisational training
	Organisational process performance
	Organisational innovation and deployment
Project management	Project planning
	Project monitoring and control
	Supplier agreement management
	Integrated project management
	Risk management
	Integrated teaming
	Quantitative project management

CMMI process areas 2

Engineering	Requirements management
	Requirements development
	Technical solution
	Product integration
	Verification
	Validation
Support	Configuration management
	Process and product quality management
	Measurement and analysis
	Decision analysis and resolution
	Organisational environment for integration
	Causal analysis and resolution

CMMI goals

Goal	Processarea
Corrective actions are managed to closure when the project's performance or results deviate significantly from The plan.	Specific goal in Project Monitoring And Control
Actual performance and progress of the project is monitored Against the project plan.	Specific goal in project monitoring And control
The requirements are analyzed and validated and a Definition of the required functionality is developed.	Specific goal in requirements development.
Root causes of defects and other problems are Systematically determined.	Specific goal in causal analysis and resolution.
The process is institutionalized as a defined process.	Genericgoal

CMMI practices

Practice	Associatedgoal
Analyse derived requirements to ensure that they are Necessary and sufficient	The requirements are analysed and Validated and a definition of the required Functionality is developed.
Validaterequirementstoensurethattheresultingproduct willperformasintendedintheuser'senvironmentusing multipletechniquesasappropriate.	
Select the defects and other problems for analysis.	Root causes of defects and other Problems are systematically determined.
Perform causal analysis of selected defects and other Problems and propose actions to address them.	
Establish and maintain an organizational policy for Planning and performing the requirements development process.	The process is institutionalized as a Defined process.
Assign responsibility and authority for performing the process, developing the work products and providing the Services of the requirements development process.	

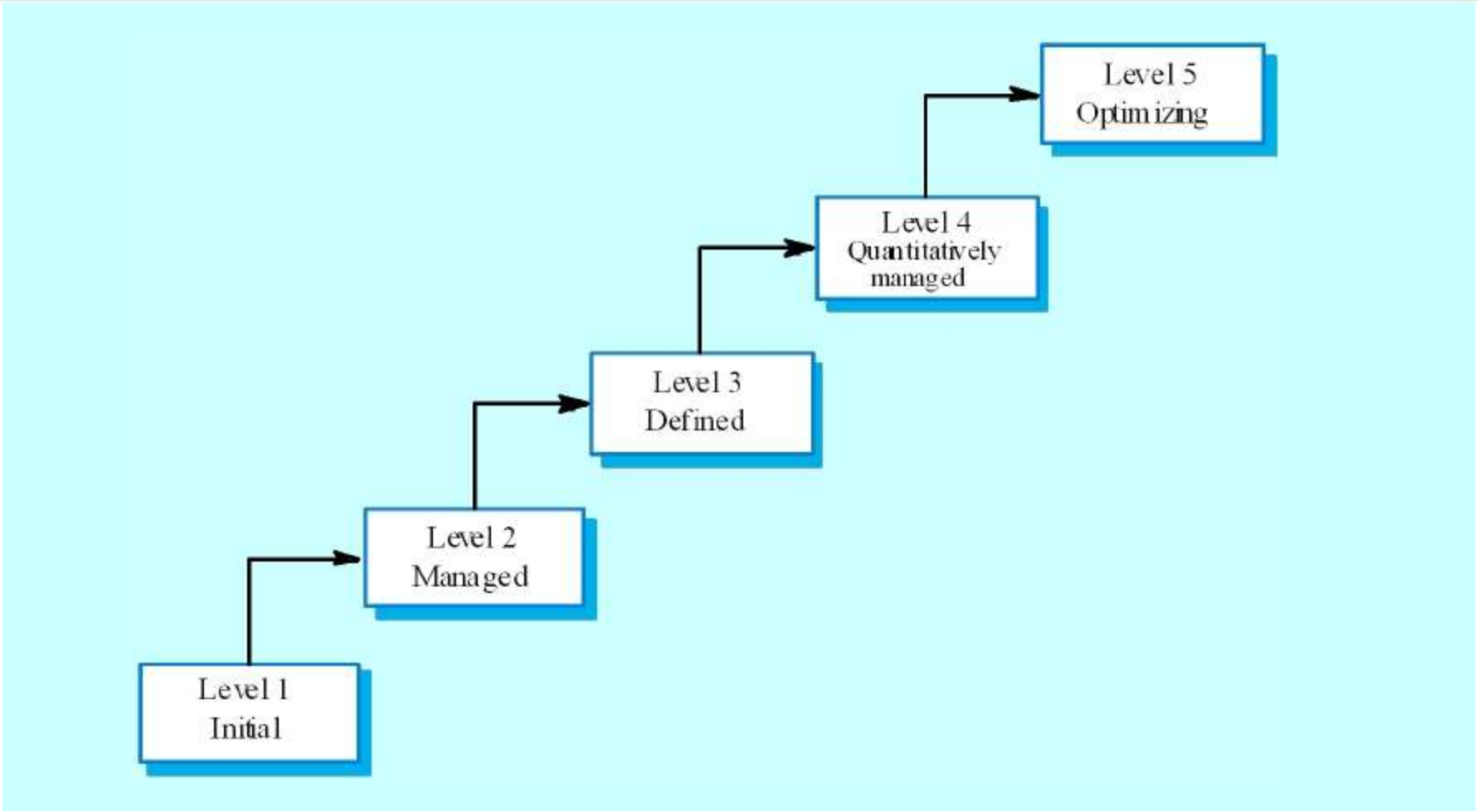
CMMI assessment

- Examines the processes used in an organisation and assesses their Maturity in each process area.
- Based on a6-point scale:
  - Not performed;
  - Performed;
  - Managed;
  - Defined;
  - Quantitatively managed;
  - Optimizing.

The staged CMMI model

- Each maturity level has process areas and goals. For example, the process area associated with the managed level include:
  - Requirements management;
  - Project planning;
  - Project monitoring and control;
  - Supplier agreement management;
  - Measurement and analysis;
  - Process and product quality assurance.

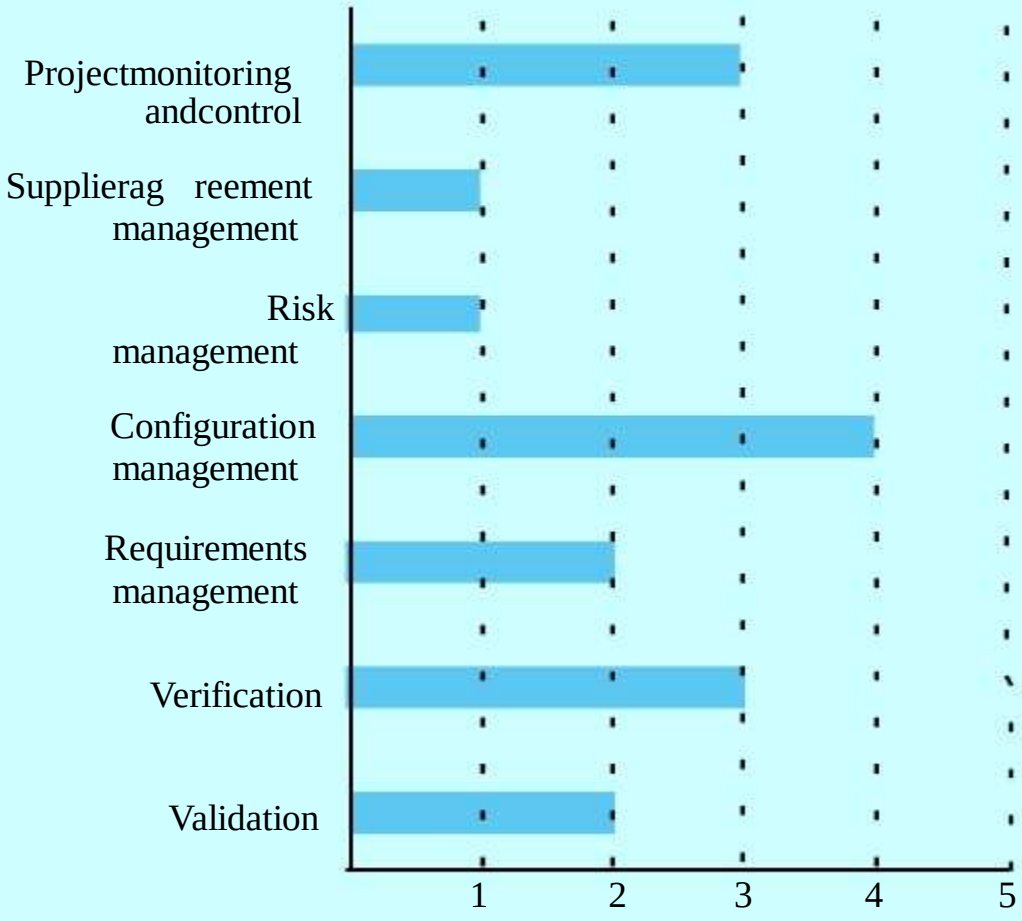
The staged CMMI model



The continuous CMMI model

- This is a finer-grain model that considers individual or groups of practices and assesses their use.
- The maturity assessment is not a single value but is a set of values showing the organizations maturity in each area.
- The CMMI rateseach process area from levels 1 to5.
- The advantage of a continuous approach is that organizations can pick and choose process areas to improve according to their local needs.

A process capability profile



References

- SoftwareEngineering –9thEdition
  IanSommerville

Questions?