AGILE SOFTWARE DEVELOPMENT

- Agile methods
- Agile development techniques
- Agile project management
- Scaling agile methods

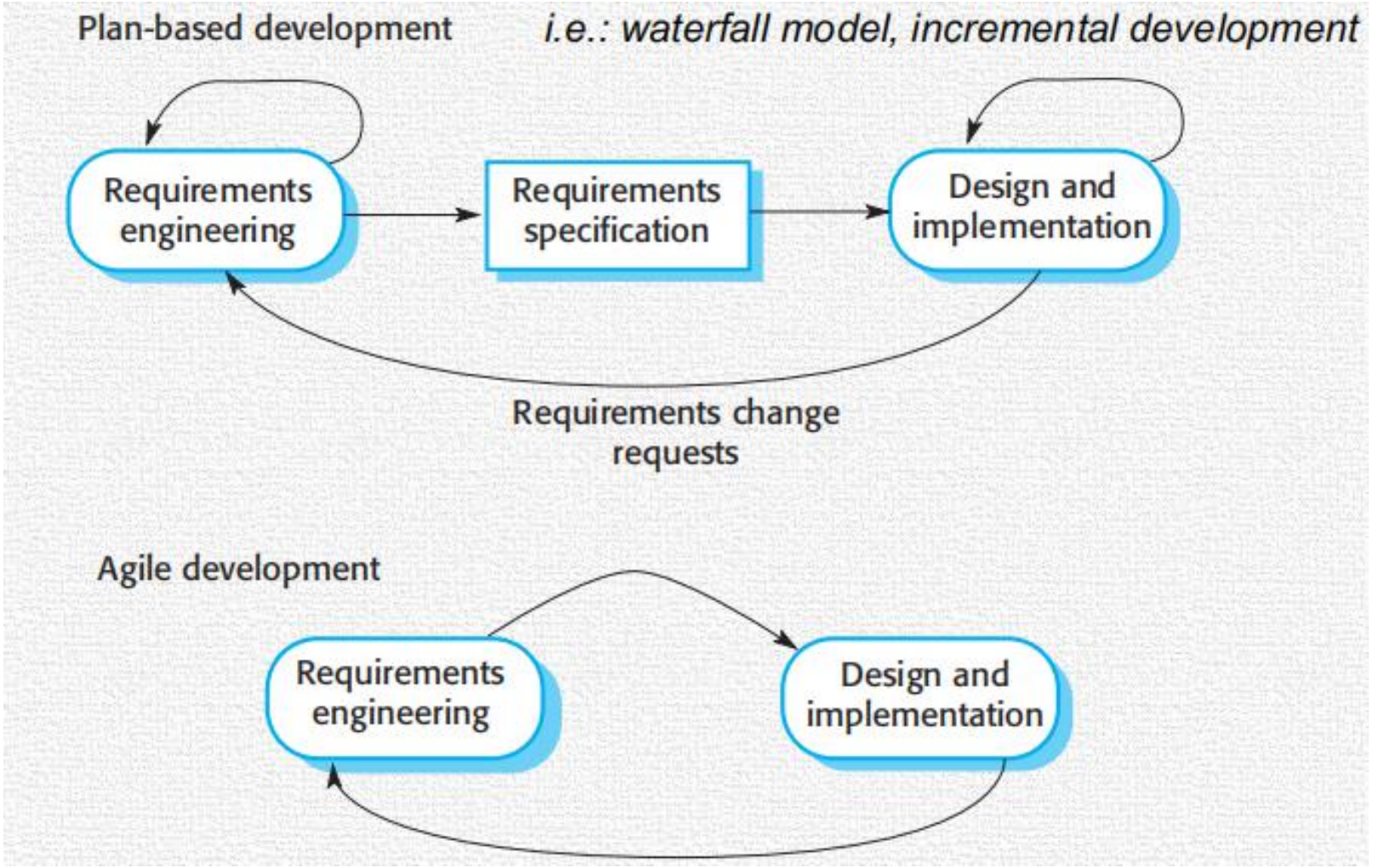
Rapid software development

- Rapid development and delivery is now often the most important requirement for software systems
- Businesses operate in a fast-changing requirement
- => practically impossible to have stable software requirements
- Software has to evolve quickly to reflect changing business needs.
- Plan-driven development (waterfall, incremental dev.) is essential for some types of system but does not meet these business needs.

AGILE DEVELOPMENT

- Late 1990s | Aim to radically reduce the delivery me for working software system
- Program specification, design and implementation are interleaved
  - The system = a series of versions/increments
  - Stakeholder involved in the version specification and evaluation
  - Frequent delivery of new versions for evaluation
- Minimal documentation - focus on working code
  - Extensive tool support (automated testing tools)

Plan-driven and agile development



AGILE METHOD

AGILE is a set of 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

Aim: to reduce overheads in the software process by limiting documentation; to respond quickly to changing requiements without excessive rework

AGILE METHOD

MANIFESTO for AGILE:

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.

The principles of agile methods

Principle	Description
Customer involvement	Customers should be closely involved throughout 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. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace **change**	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain **simplicity**	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Agile method applicability

For a small or medium-sized product for sale

• Virtually all software products and apps are now developed using an agile approach

Clear commitment from the customer to become involved

• in the development process

• and where there are few external rules and regulations that affect the software

AGILE DEVELOPMENT TECHNIQUES

EXTREME PROGRAMMING: An extreme approach to iterative development

+ New versions may be built several times per day
+ Increments are delivered to customers every 2 weeks
+ All tests must be run for every build and the build is only accepted if tests run successfully

XP practices: home read

EXTREME PROGRAMMING

KEY PRACTICES:

+ User stories for specifications: user is a part of XP team, who make decisions on requirement expressed as user stories or scenarios
+ Refactoring: constant code improvement (i.e: re-organize class hierarchy, tidy up code, rename, make them easy to understand, replace code with calls included in the library)
+ Test-first development
+ Pair programming: working in pair, developing code together as review process | pairs created dynamically so that all member work with each other during the development process.

XP and AGILE

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

AGILE PROJECT MANAGEMENT

Software project managers is to manage the project so that the software is delivered on time and within the planned budget for the project

Standard approach = Project plan
- what should be delivered?
- when it should be delivered?
- who will work on the development of the deliverables?

Agile project management?

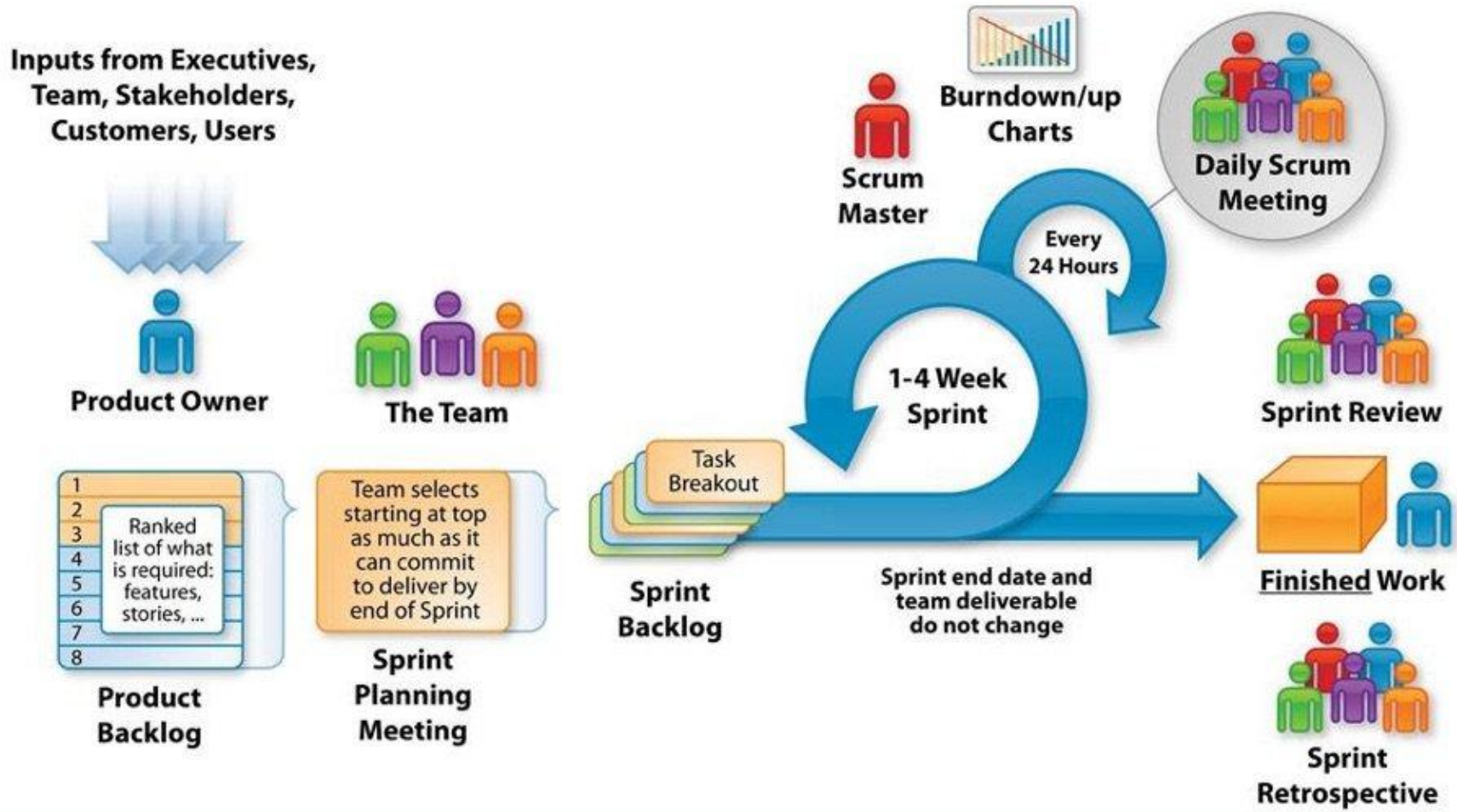
SCRUM

Scrum is an agile method that focuses on managing iterative development rather than specific agile practices.

There are three phases in Scrum.

- The initial phase is an outline planning phase where you establish the general objectives for the project and design the software architecture.
- This is followed by a series of **sprint cycles**, where each cycle develops an increment of the system.
- The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project

The Agile - Scrum Framework



SUMMARY

Agile methods are incremental development methods that focus on:

- rapid software development,
- frequent releases of the software,
- reducing process overheads by minimizing documentation
- and producing high-quality code.

Agile development practices include:

- User stories for system specification
- Frequent releases of the software,
- Continuous software improvement
- Test-first development
- Customer participation in the development team.

SUMMARY

Scrum is an agile method that provides a project management framework.

• It is centred round a set of sprints, which are fixed time periods when a system increment is developed.

Many practical development methods are a mixture of plan-based and agile development.

• Scaling agile methods for large systems is difficult.

Large systems need up-front design and some documentation and organizational practice may conflict with the informality of agile approaches