NANYANG TECHNOLOGICAL UNIVERSITY

CZ3002 - Advanced Software Engineering

Software Project Management - Introduction to Agile Software Development

Faculty : Dr Shen Zhiqi
School : School of Computer Science and Engineering
Email : zqshen@ntu.edu.sg
Office : N4-02B-43

Lesson Objectives

At the end of the lesson, you should be able to:

- Outline the principles of agile methods
- Compare between plan-driven and agile development approaches
- Discuss the technical, human and organisational issues in agile processes

What is Agile?

- An iterative and incremental (evolutionary) approach performed in a highly collaborative form to produce high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.
- The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

Why Agile Development?

- Dissatisfaction with the overheads involved in conventional software design methods:
  - It is impossible to gather all the requirements at the beginning of a project
  - Whatever requirements you do gather are guaranteed to change
  - There will always be more to do than time and money will allow

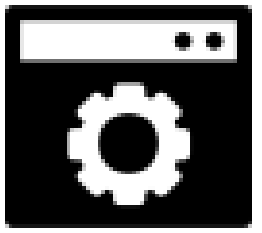
- These methods:
  - Focus on the code rather than the design
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements

The Manifesto of Agile Methods

Agile Methods

- Individuals and interaction over processes and tools

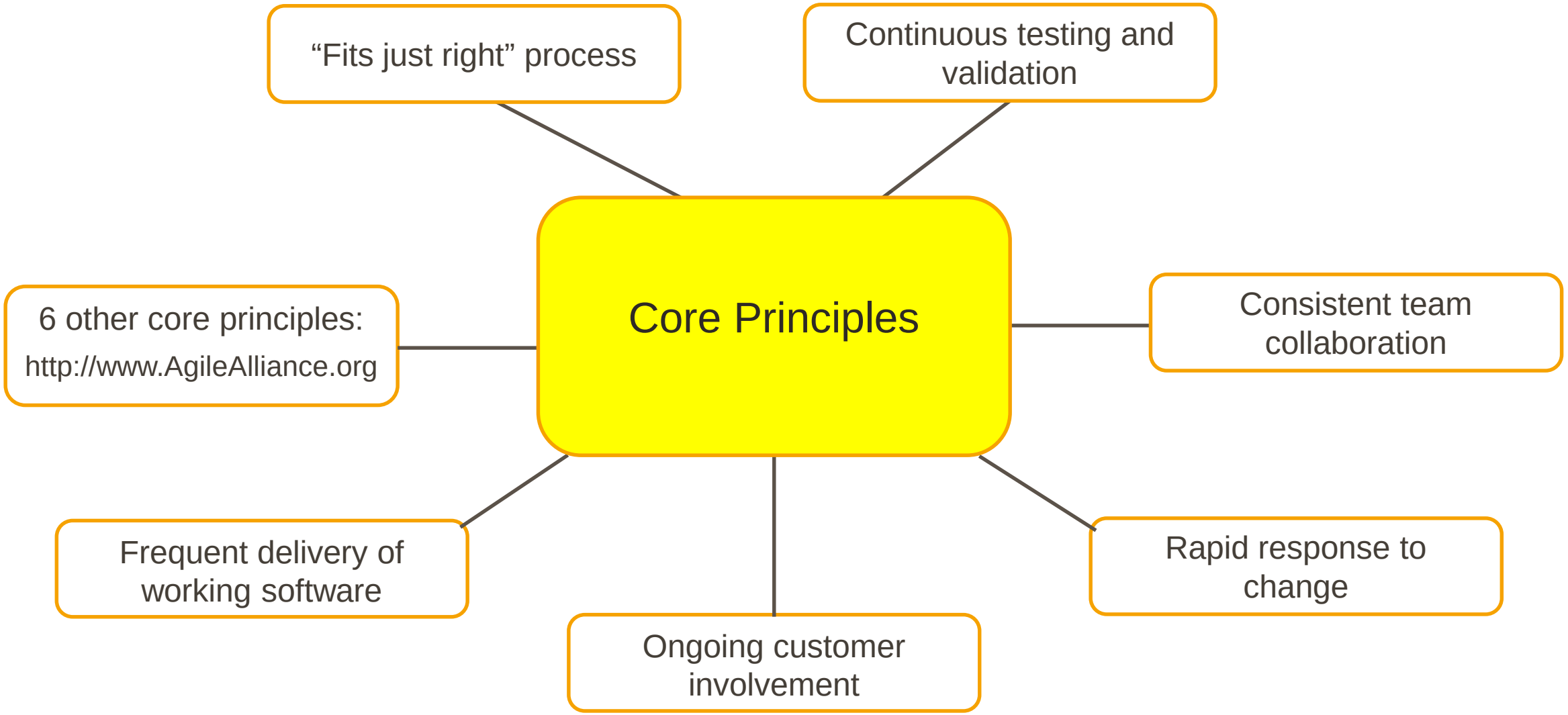
- Working software over comprehensive documentation



- Customer collaboration over contract negotiation

- Responding to change over following a plan

Some of the Core Principles



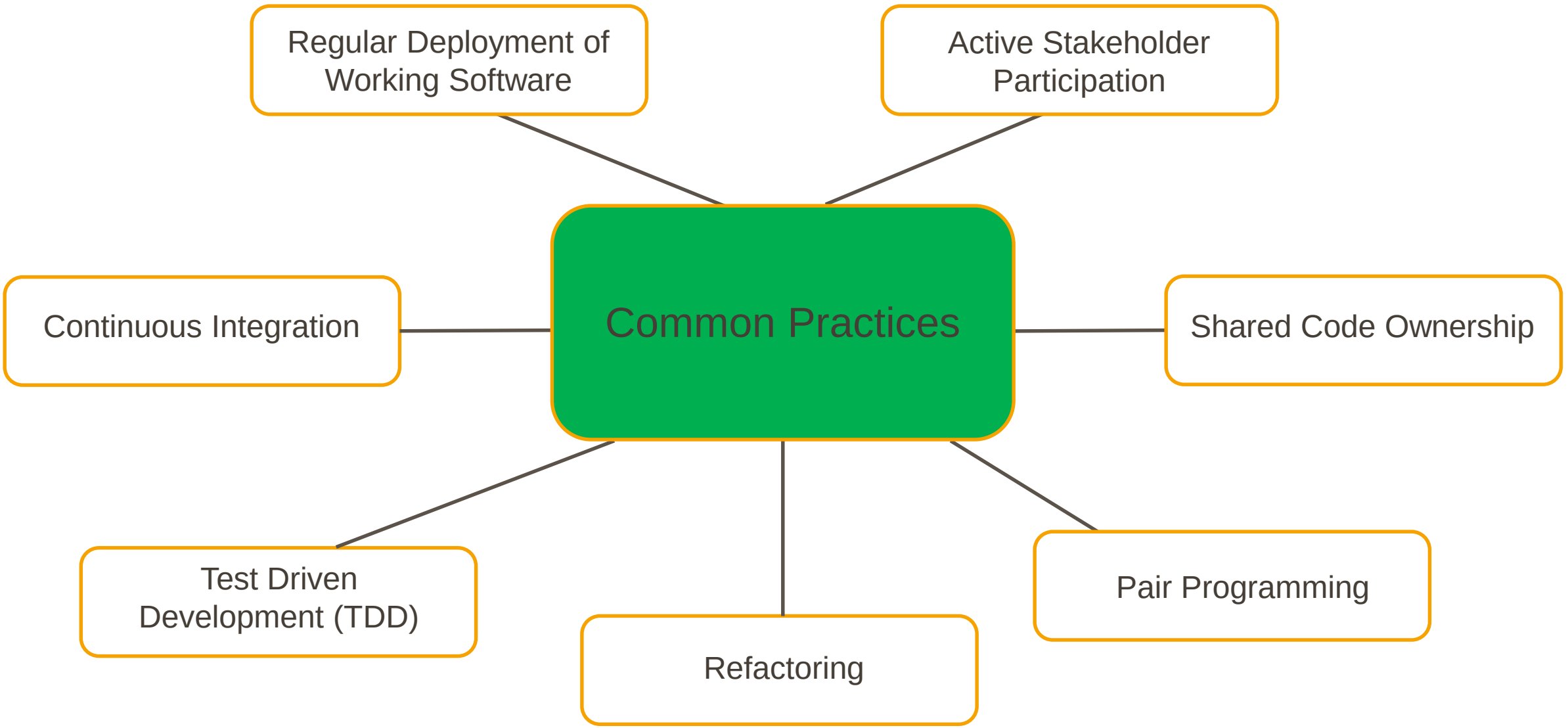
The Principles of Agile Methods

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is to provide and prioritise 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 recognised 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.

Characteristics of Agile Software Development

- Lightweight methodology
- Small to medium sized teams
- Vague and/ or changing requirements
- Vague and/ or changing techniques
- Simple design
- Minimal system into production

Some Common Practices



Agile Method Applicability

- Product development where a software company is developing a small or medium-sized product for sale
- Custom system development within an organisation, where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations that affect the software
- Because of their focus on small, tightly-integrated teams, there are problems in scaling agile methods to large systems

Agile Methods

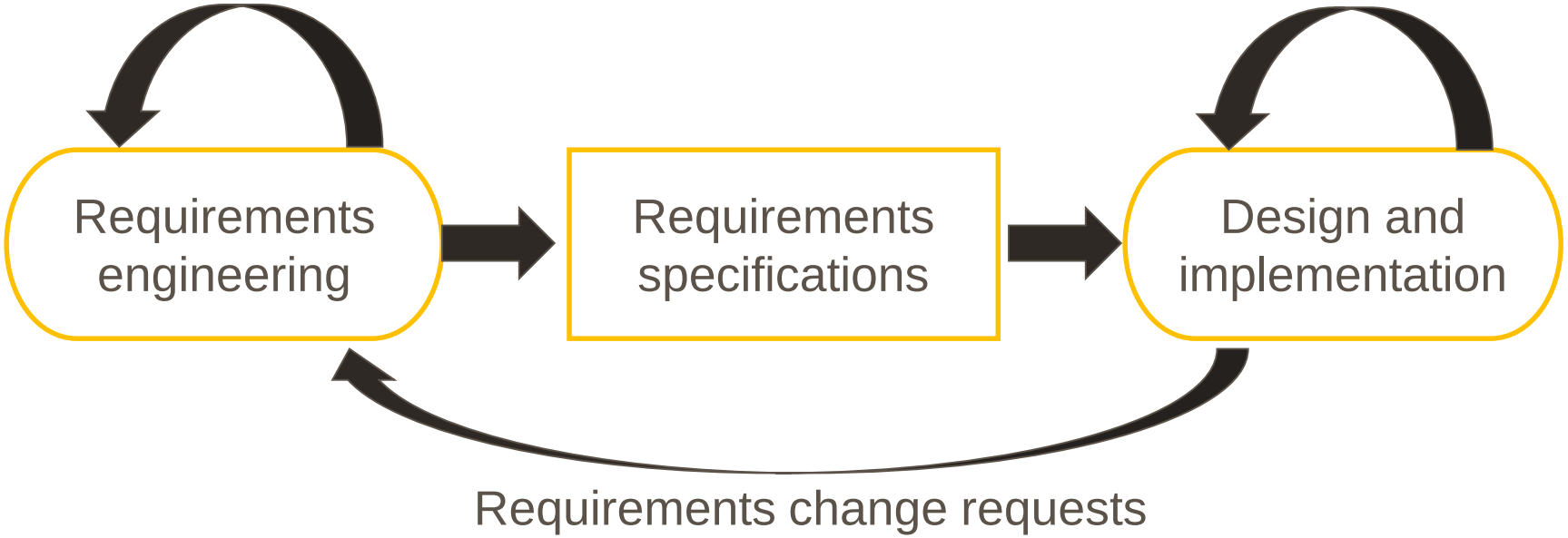
- Adaptive Software Development
- Agile Unified Process
- Crystal
- Dynamic Systems Development Method
- Extreme Programming
- Feature-Driven Development
- Lean Software Development
- Rational Unified Process
- Scrum
- …

Plan-Driven and Agile Development

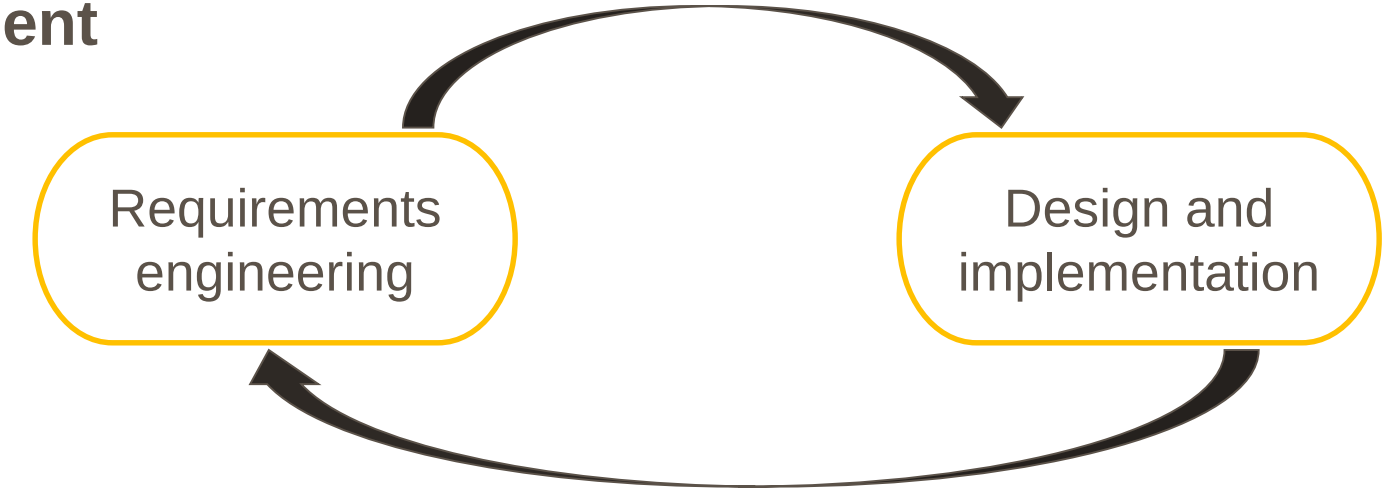
Plan-driven development	Agile development
A plan-driven approach to software engineering is based around separate development stages with the outputs to be produced at each of these stages planned in advance	Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process
Not necessarily waterfall model – plan-driven, incremental development is possible	
Iteration occurs within activities	

Plan-Driven and Agile Specification

Plan-based development



Agile development



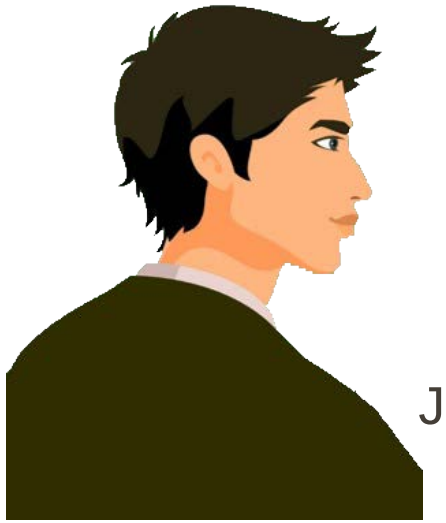
Technical, Human, Organisational Issues

- Most projects include elements of plan-driven and agile processes. Deciding on the balance depends on:
  - Is it important to have a very detailed specification and design before moving to implementation? If so, you probably need to use a plan-driven approach.
  - Is an incremental delivery strategy, where you deliver the software to customers and get rapid feedback from them, realistic? If so, consider using agile methods.
  - How large is the system that is being developed? Agile methods are most effective when the system can be developed with a small co-located team who can communicate informally. This may not be possible for large systems that require larger development teams so a plan-driven approach may have to be used.

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

What type of system is being developed?



James

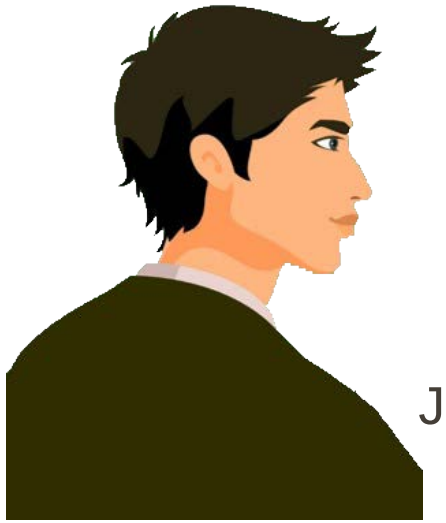
Plan-driven approaches may be required for systems that require a lot of analysis before implementation (e.g. real-time system with complex timing requirements).

Adam

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

What is the expected system lifetime?



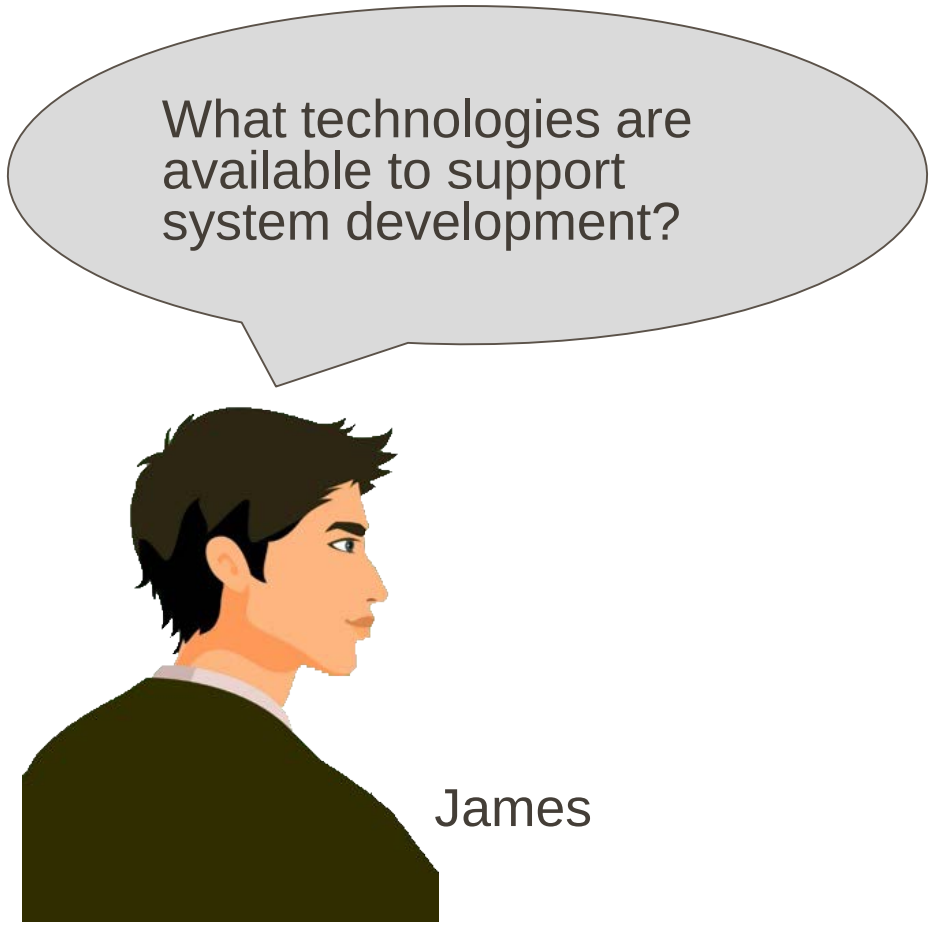
James

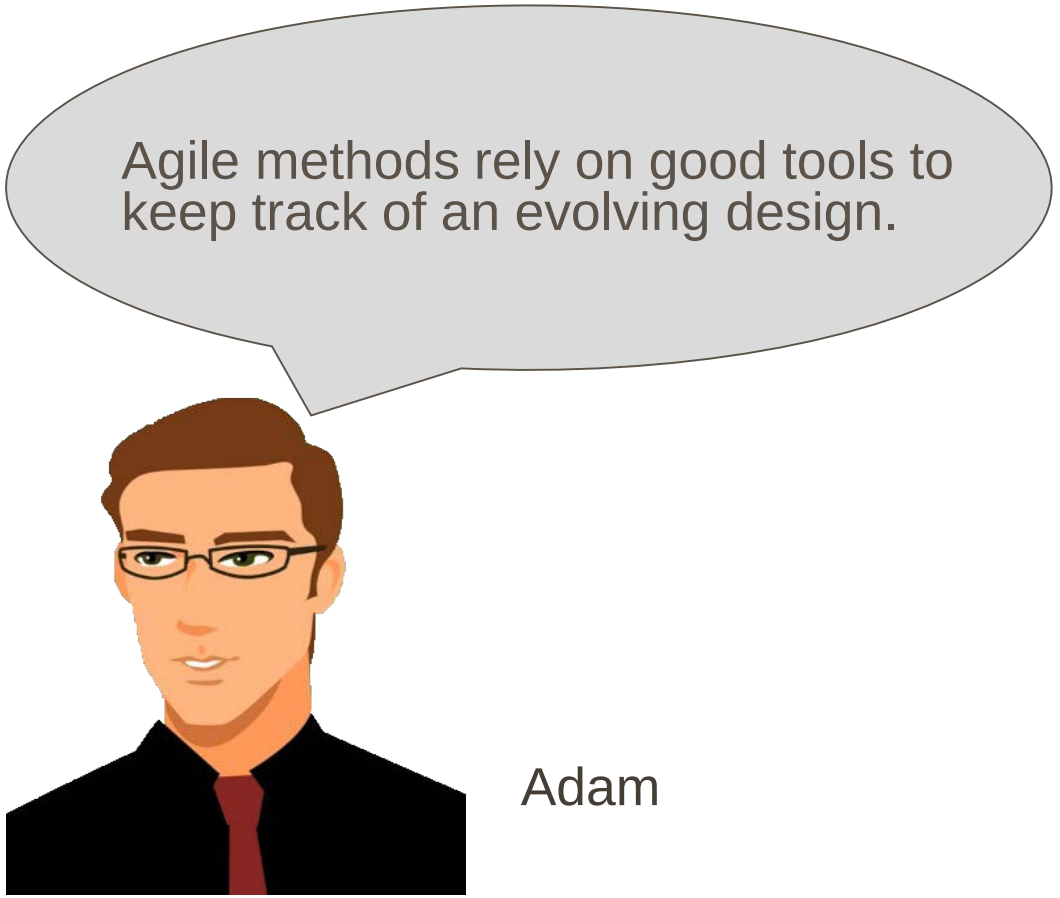
Long-lifetime systems may require more design documentation to communicate the original intentions of the system developers to the support team.

Adam

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

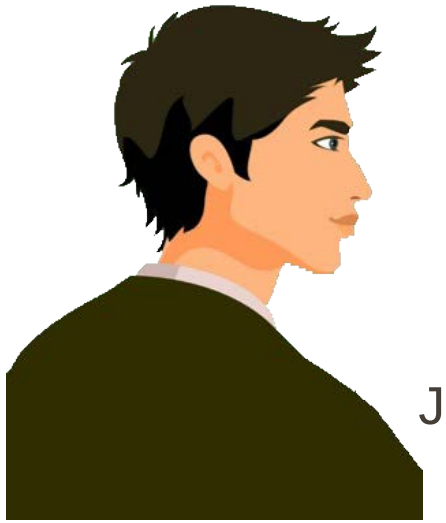




Technical, Human, Organisational Issues

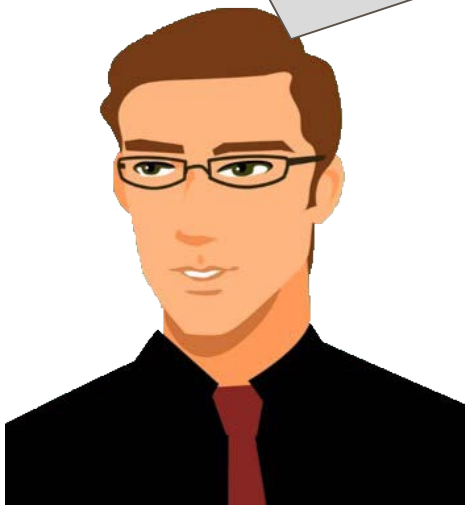
Below is the conversation between two computer science students, James and Adam.

How is the development team organised?



James

If the development team is distributed or if part of the development is being outsourced, then you may need to develop design documents to communicate across the development teams.

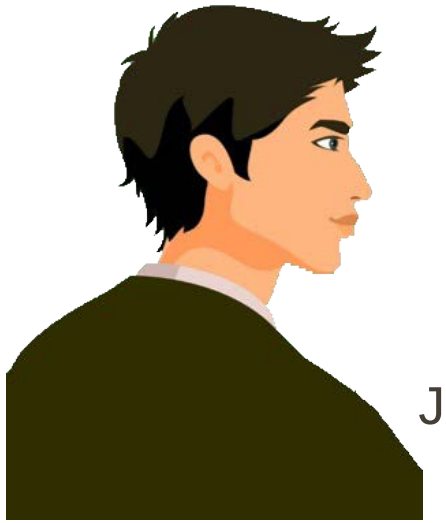


Adam

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

Are there cultural or organisational issues that may affect the system development?



James

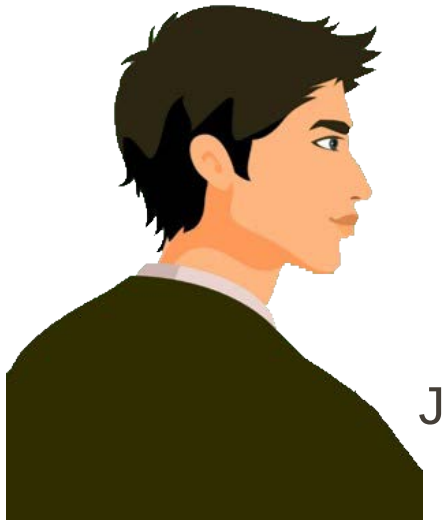
Traditional engineering organisations have a culture of plan-based development, as this is the norm in engineering.

Adam

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

How good are the designers and programmers in the development team?



James

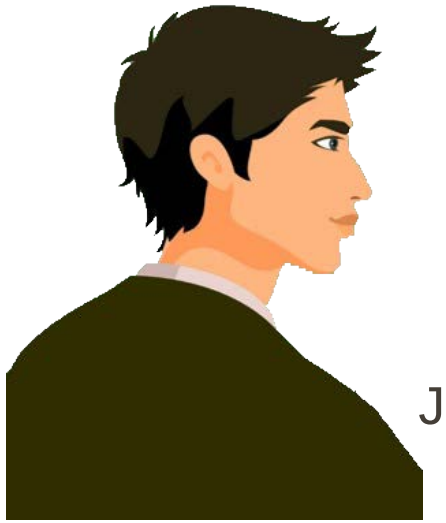
It is sometimes argued that agile methods require higher skill levels than plan-based approaches in which programmers simply translate a detailed design into code.

Adam

Technical, Human, Organisational Issues

Below is the conversation between two computer science students, James and Adam.

Is the system subject to external regulation?



James

If a system has to be approved by an external regulator (e.g. the FAA approve software that is critical to the operation of an aircraft) then you will probably be required to produce detailed documentation as part of the system safety case.

Adam

Agile Methods and Software Maintenance

- Most organisations spend more on maintaining existing software than they do on new software development. So, if agile methods are to be successful, they have to support maintenance as well as original development.
- Two key issues:
  - Are systems that are developed using an agile approach maintainable, given the emphasis in the development process of minimizing formal documentation?
  - Can agile methods be used effectively for evolving a system in response to customer change requests?
- Problems may arise if original development team cannot be maintained.

Highlights

- Agile methods are lightweight software methods.
- Agile development methods are very pragmatic in understanding the fact that requirements in a business environment changes constantly.
- Highly creative people who have understood the shortcomings of normal software management processes are using agile development methods in organisations.
- Many organisations all around the world are trying out the various available Agile development methods. Such as Google, Facebook, IBM, Microsoft, Salesforce.