Rapid software development

Objectives

- To explain how an iterative, incremental development process leads to faster delivery of more useful software
- To discuss the essence of agile development methods
- To explain the principles and practices of extreme programming
- To explain the roles of prototyping in the software process

Topics covered

- Agile methods
- Extreme programming
- Rapid application development
- Software prototyping

Rapid software development

- Because of rapidly changing business environments, businesses have to respond to new opportunities and competition.
- This requires software and rapid development and delivery is often the most critical requirement for software systems.
- Businesses may be willing to accept lower quality software if rapid delivery of essential functionality is possible.

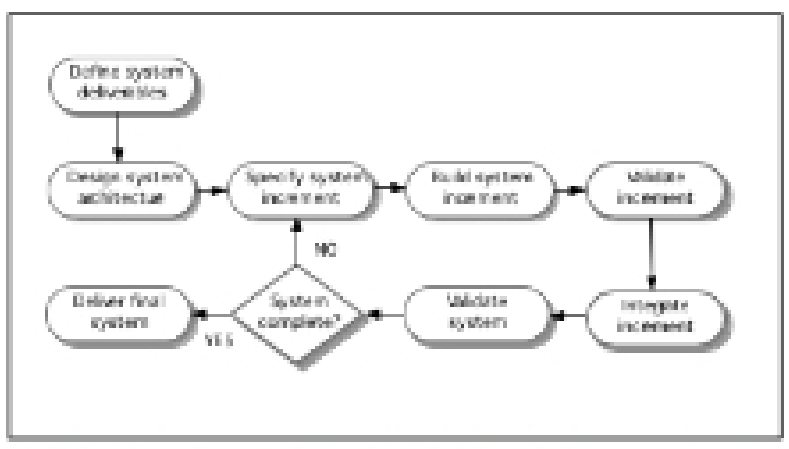
Requirements

- Because of the changing environment, it is often impossible to arrive at a stable, consistent set of system requirements.
- Therefore a waterfall model of development is impractical and an approach to development based on iterative specification and delivery is the only way to deliver software quickly.

Characteristics of RAD processes

- The processes of specification, design and implementation are concurrent. There is no detailed specification and design documentation is minimised.
- The system is developed in a series of increments. End users evaluate each increment and make proposals for later increments.
- System user interfaces are usually developed using an interactive development system.

An iterative development process



Advantages of incremental development

- Accelerated delivery of customer services. Each increment delivers the highest priority functionality to the customer.
- User engagement with the system. Users have to be involved in the development which means the system is more likely to meet their requirements and the users are more committed to the system.

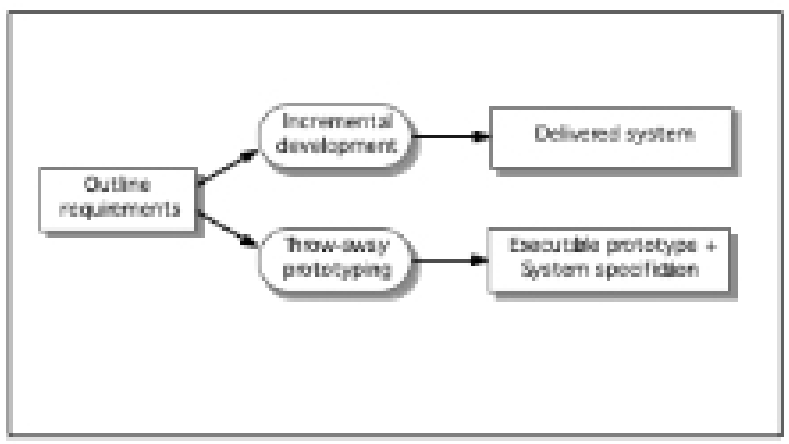
Problems with incremental development

- Management problems
  - Progress can be hard to judge and problems hard to find because there is no documentation to demonstrate what has been done.
- Contractual problems
  - The normal contract may include a specification; without a specification, different forms of contract have to be used.
- Validation problems
  - Without a specification, what is the system being tested against?
- Maintenance problems
  - Continual change tends to corrupt software structure making it more expensive to change and evolve to meet new requirements.

Prototyping

- For some large systems, incremental iterative development and delivery may be impractical; this is especially true when multiple teams are working on different sites.
- Prototyping, where an experimental system is developed as a basis for formulating the requirements may be used. This system is thrown away when the system specification has been agreed.

Incremental development and prototyping



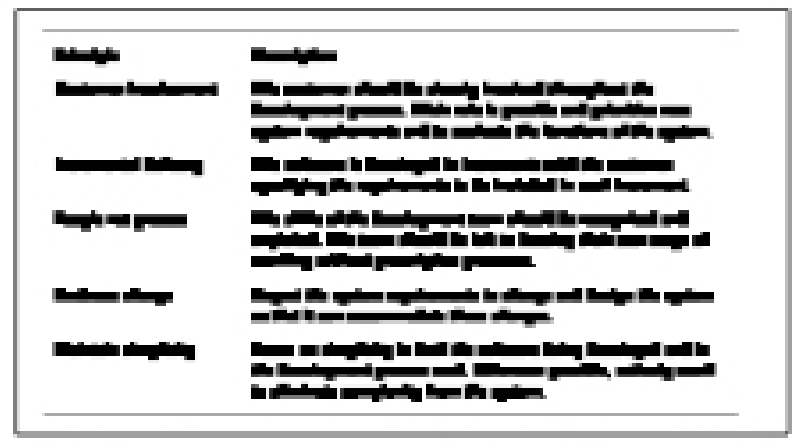
Conflicting objectives

- The objective of incremental development is to deliver a working system to end-users. The development starts with those requirements which are best understood.
- The objective of throw-away prototyping is to validate or derive the system requirements. The prototyping process starts with those requirements which are poorly understood.

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



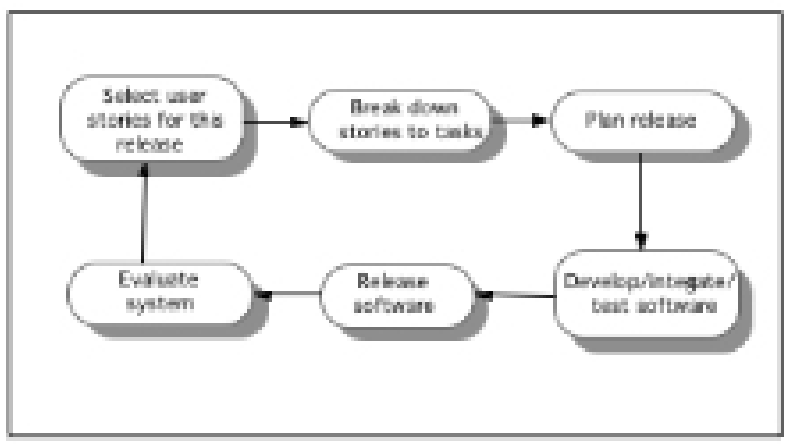
Problems with agile methods

- It can be difficult to keep the interest of customers who are involved in the process.
- Team members may be unsuited to the intense involvement that characterises agile methods.
- Prioritising 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

- Perhaps the best-known and most widely used agile method.
- Extreme Programming (XP) takes 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.

The XP release cycle



Extreme programming practices 1