
Rapid software development

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.

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 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. 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.
