

CHAPTER TWO

SOFTWARE PROCESSES

DR. HOSAM ELDIN

CONTENTS

- 2.1 Software process models
- 2.2 Process activities
- 2.4 The Rational Unified Process

SOFTWARE PROCESS

- A software process is a set of related activities that leads to the production of a software product. These activities may involve the development of software from scratch in a standard programming language like Java or C.

THE SOFTWARE PROCESS

4

❓ *Software process*: a structured set of activities required to develop a software system.

❓ Many different software processes but all involve:

- *Specification* – defining what the system should do;
- *Design and implementation* – defining the organization of the system and implementing the system;
- *Validation* – checking that it does what the customer wants;
- *Evolution* – changing the system in response to changing customer needs.

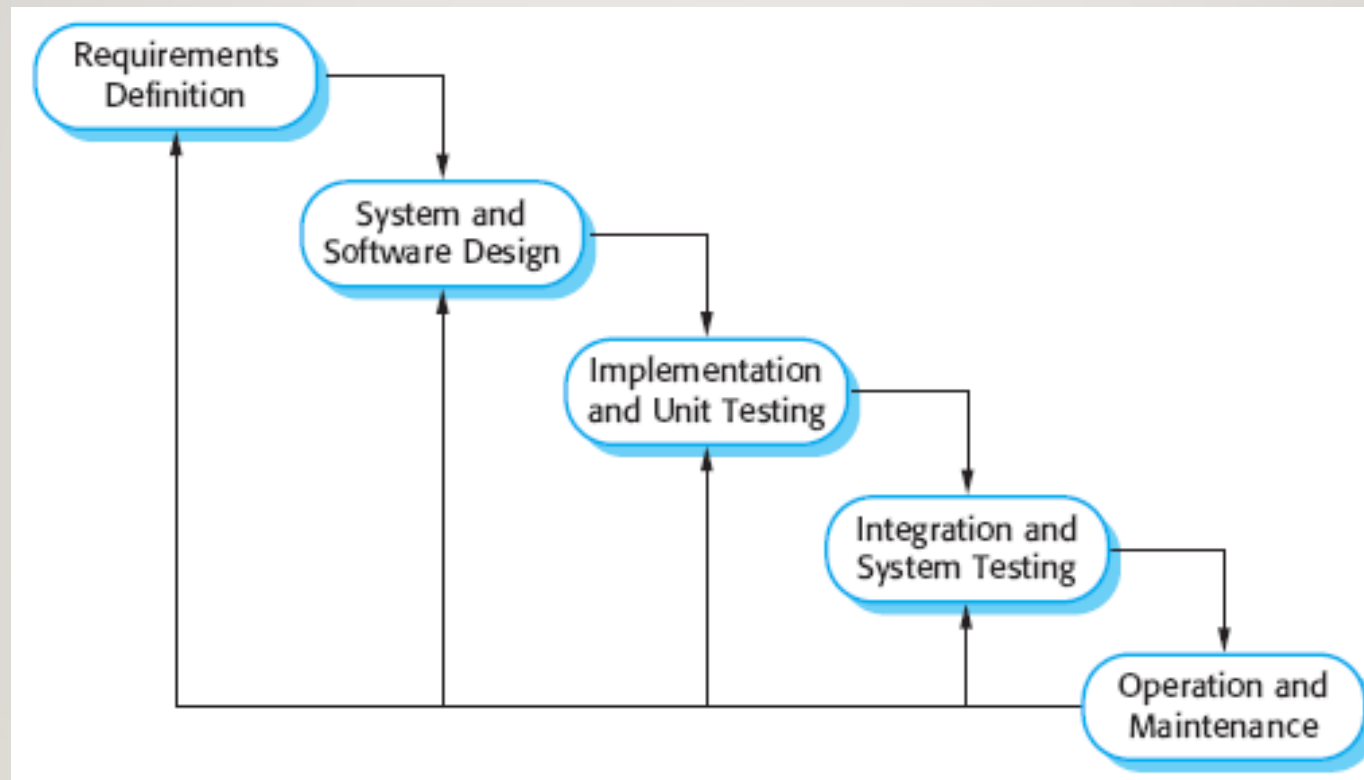
❓ A *software process model* is an abstract representation of a process. It presents a description of a process from some particular perspective.

Software Process Models

- The waterfall model
- Incremental development
- Reuse-oriented software engineering
- V-MODEL
- Boehm's spiral model
- Agile model

I - The Waterfall Model

The waterfall model This takes the fundamental process activities of specification, development, validation, and evolution and represents them as separate process phases such as requirements specification, software design, implementation, testing, and so on.



Stages Of Waterfall Model

- *1. Requirements analysis and definition* The system's services, constraints, and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.
- *2. System and software design* The systems design process allocates the requirements to either hardware or software systems by establishing an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.
- *3. Implementation and unit testing* During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

Stages Of Waterfall Model

- 4. *Integration and system testing* The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.
- 5. *Operation and maintenance* Normally (although not necessarily), this is the longest life cycle phase. The system is installed and put into practical use.

Advantages Of Waterfall Model:

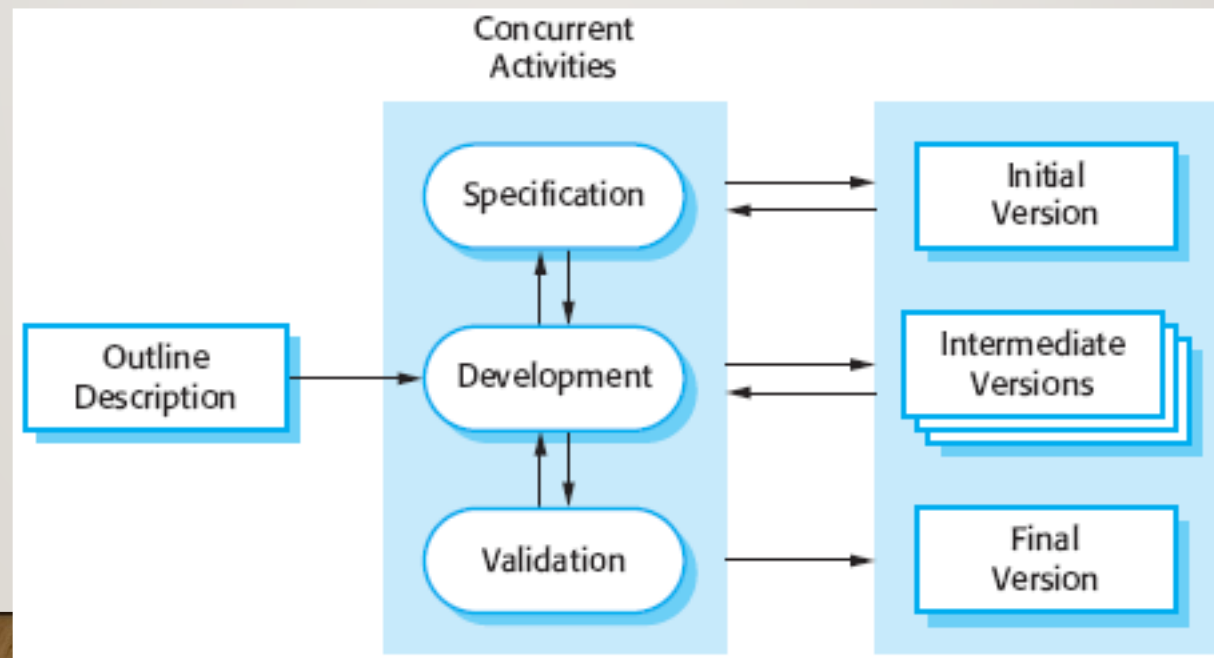
1. It is easy to understand, and easy to use
2. It provide structure to inexperienced staff
3. Works well on mature products and weak teams
4. It is document driven process model, based on published documentation standards.
5. Works well when quality is more important than cost or timing.
6. Sets requirements stability.
7. Identifies deliverables and milestones.

Disadvantages Of Waterfall Model

1. All requirements must be known upfront and these are stable however in evolutionary models requirements evolve over the time .
2. Software is delivered late in project, discovery of serious errors delayed.
3. Unlike to spiral model , difficult to integrate risk Management
4. Difficult and expensive to make changes to documents.
5. Due to administration, costly for small teams and projects.

2- Incremental Development

- This approach interleaves the activities of specification, development, and validation. The system is developed as a series of versions (increments), with each version adding functionality to the previous version.



Incremental Development Has Three Important Benefits, Compared To The Waterfall Model:

1. **The cost of accommodating changing customer requirements is reduced.** The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.
2. **It is easier to get customer feedback** on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. Customers find it difficult to judge progress from software design documents.
3. **More rapid delivery** and deployment of useful software to the customer is possible, even if all of the functionality has not been included. Customers are able to use and gain value from the software earlier than is possible with a waterfall process.

Incremental Approach Has Two Problems

- 1. The process is not visible. Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system.
- 2. System structure tends to degrade as new increments are added. Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.

When To Use The Incremental Model?

- Risk, funding, timing, program complexity, or need for early realization of benefits.
- Unlike sequential models most of the requirements are known earlier but are expected to evolve over time
- A need to get basic functionality to the market early
- long time development projects
- Unlike sequential models it must be used on a project with new technology.

Advantages Of Incremental Models

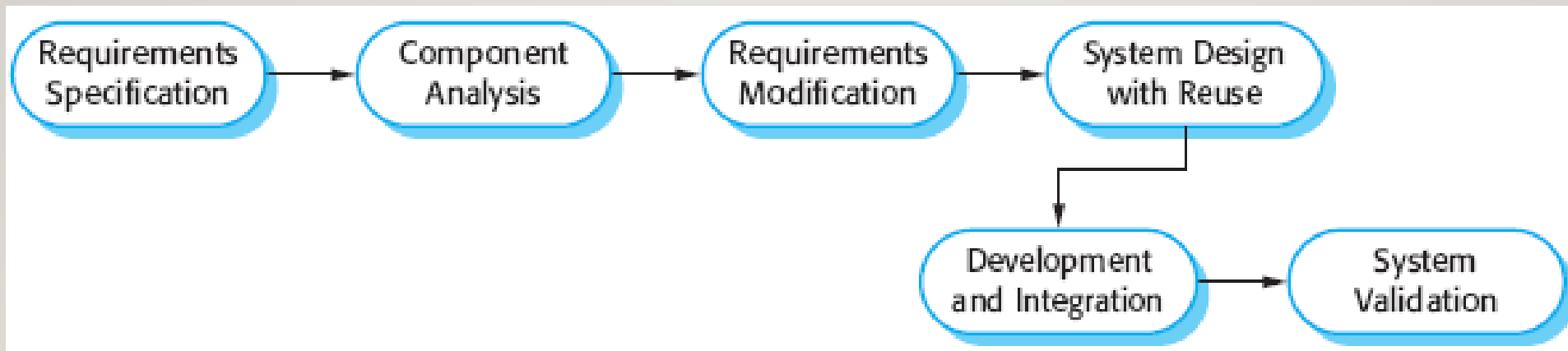
1. The customer receives working software before the end of the project.
2. satisfying some of the stakeholder's early, and this provides visible signs of progress but it is not possible in the sequential models.
3. The cost expended before some functionality is delivered to the customer is less compared to the Waterfall model
4. As functionality is incremented in steps, testing also becomes easy.
5. Risk of failure of a product is decreased as users start using the product early.

Disadvantages Of Incremental Model :

1. Requires good planning and design
2. Requires early definition of a complete and fully functional system (to allow for the definition of Increments)
3. Total cost of the complete system is not lower
4. Requires careful management to ensure quality of each increment.
5. Resources also need to be carefully managed to ensure that each increment is delivered on time

3- Reuse-oriented Software Engineering

In the majority of software projects, there is some software reuse. This often happens informally when people working on the project know of designs or code that are similar to what is required. They look for these, modify them as needed, and incorporate them into their system.

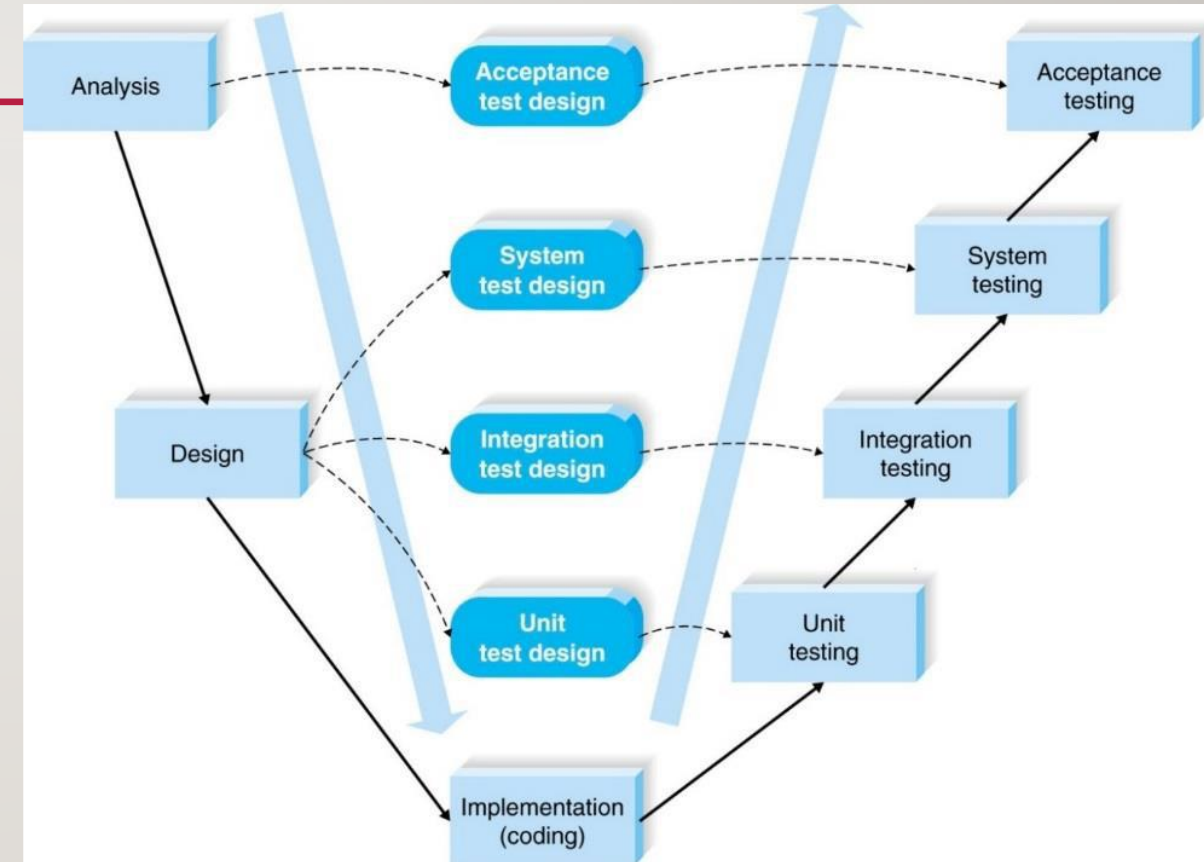


Types Of Software Component That May Be Used In A Reuse-oriented Process:

1. Web services that are developed according to service standards and which are available for remote invocation.
2. Collections of objects that are developed as a package to be integrated with a component framework such as .NET or J2EE.
3. Stand-alone software systems that are configured for use in a particular environment.

4-V-MODEL

- The **V-model** is a type of SDLC model where process executes in a sequential manner in **V-shape**.
- It is also known as Verification and Validation model.
- It is based on the association of a testing phase for each corresponding development stage.



V-MODEL

- A **key concept** of this model is that as requirements are specified and components designed, testing for those elements is also defined.

Testing Phases:

- Unit Testing
- Integration testing
- System Testing
- User Acceptance Testing (UAT):

Advantages Of V-model:

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Avoids the downward flow of the defects.

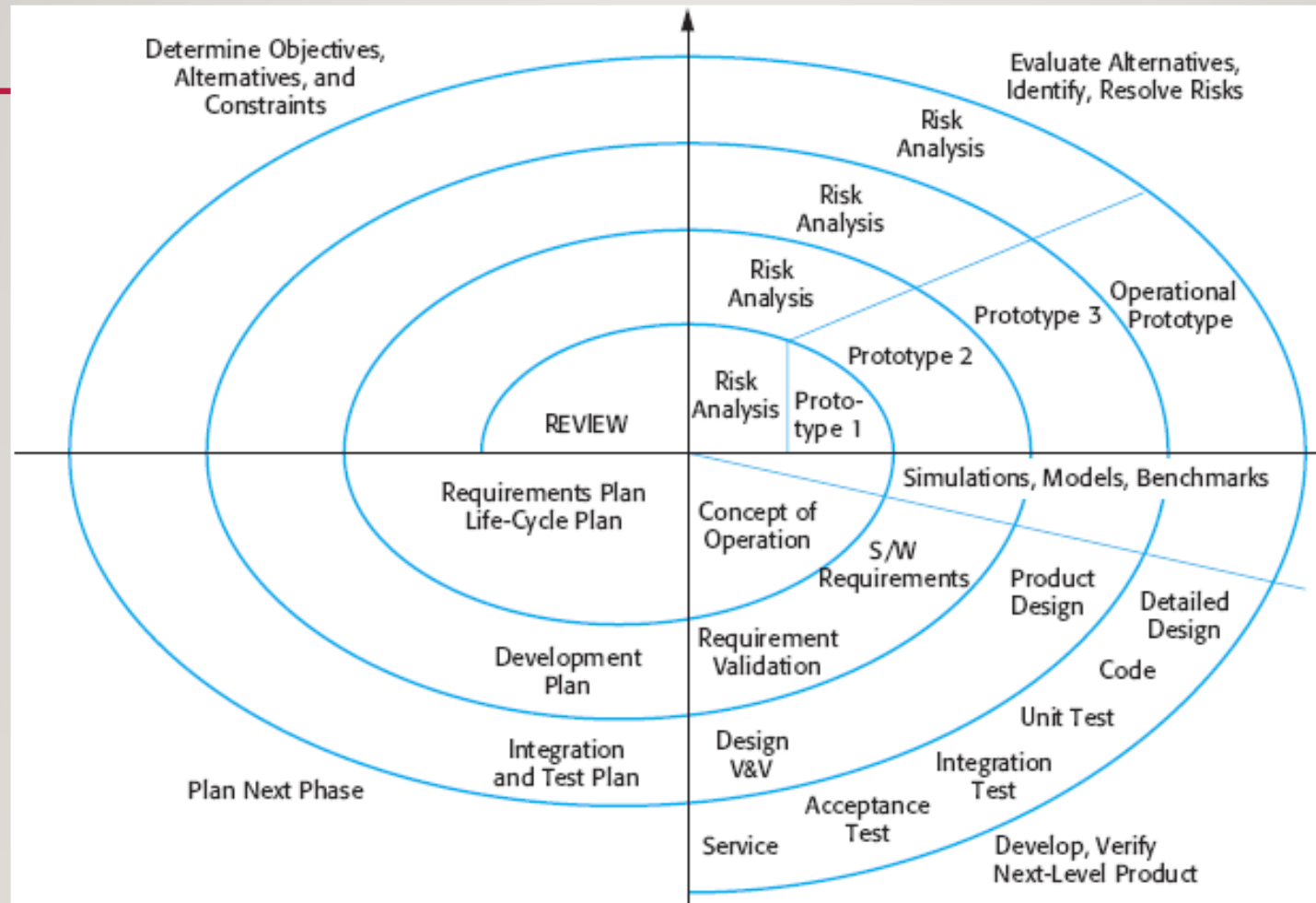
Disadvantages Of V-model:

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

5-Boehm's Spiral Model

- spiral model was proposed by Boehm (1988) the software process is represented as a spiral, rather than a sequence of activities with some backtracking from one activity to another. Each loop in the spiral represents a phase of the software process.

Phases Of Spiral Model



Advantages Of Spiral Model

- High amount of risk analysis
- Good for large and mission-critical projects.
- Software is produced early in the software life cycle.

Disadvantages Of Spiral Model

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects.

Agile Software Development

- Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

Agile Methods Have Been Very Successful For Some Types Of System Development:

- 1. Product development where a software company is developing a small or medium-sized product for sale.
- 2. Custom system development within an organization, 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.

– Agile Software Development



Topics covered

- ✧ Agile methods
- ✧ Plan-driven and agile Plan-driven and agile development
- ✧ Extreme programming
- ✧ 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 and it is practically impossible to produce a set of stable software requirements
- Software has to evolve quickly to reflect changing business needs.

✧ Rapid software development

- Specification, design and implementation are inter-leaved, There is no detailed system specification, and design documentation is minimized
- System is developed as a series of versions with stakeholders involved in version evaluation
- User interfaces are often developed using an interactive development system(IDE) and graphical toolset.

AGILE METHODS

- ✧ ~~Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s 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.
- ✧ 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.

AGILE MANIFESTO

The **Agile Manifesto** describes a customer who is engaged and collaborates throughout the development process, making. This makes it far easier for development to meet their needs of the customer. The Agile Manifesto is a document that sets out the key values and principles behind the Agile philosophy and serves to help development teams work more efficiently and sustainably.

The four core values of Agile software development as stated by the Agile Manifesto are:

- ✧ **Individuals and interactions** over processes and tools
- ✧ **Working software** over comprehensive documentation
- ✧ **Customer collaboration** over contract negotiation
- ✧ **Responding to change** over following a plan

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

- ~~✧ Product development where a software company is developing a small or medium-sized product for sale.~~
- ✧ Custom system development within an organization, 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.

Problems with agile methods

- ✧ It can be difficult to keep the interest of customers who 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.

PLAN-DRIVEN AND AGILE DEVELOPMENT

✧ Plan-driven 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.
- Not necessarily waterfall model – plan-driven, incremental development is possible
- Iteration occurs within activities.

✧ Agile development

- Specification, design, implementation and testing are interleaved and the outputs from the development process are decided through a process of negotiation during the software development process.

PLAN-DRIVEN AND AGILE SPECIFICATION

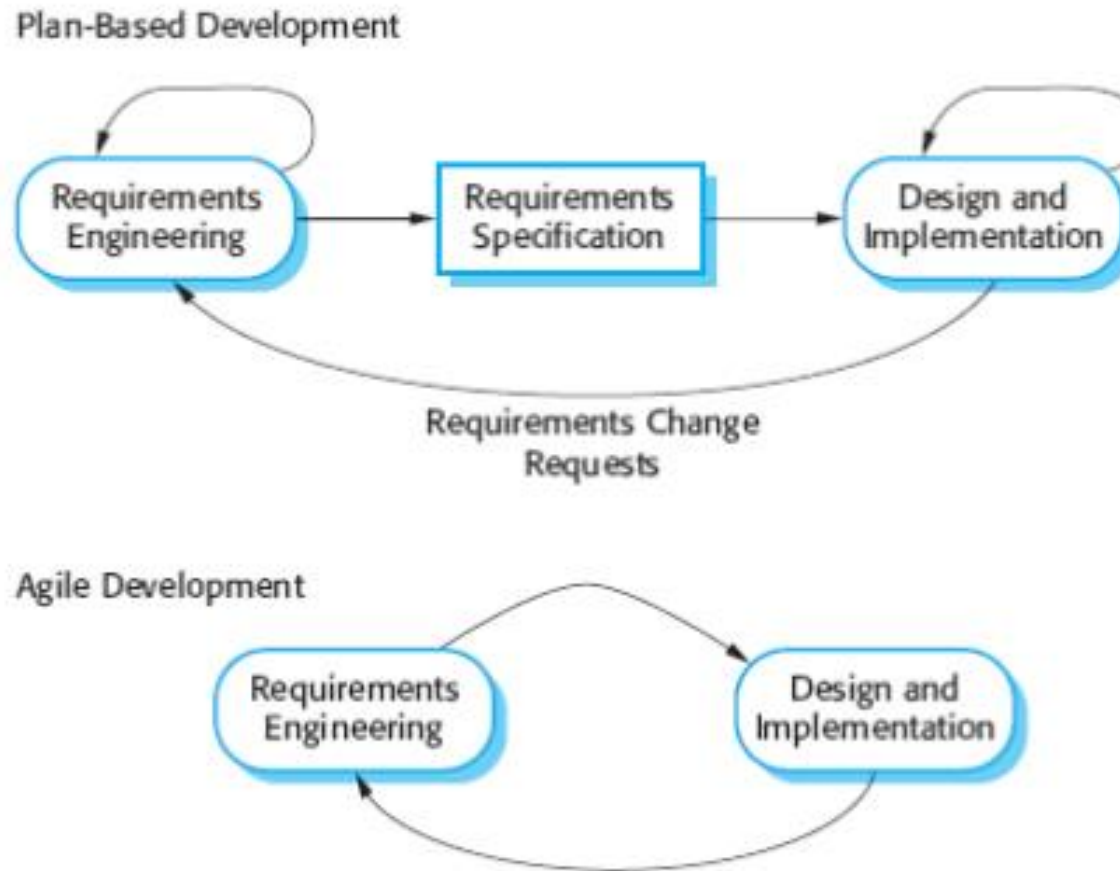


Figure 3.2 Plan-driven and agile specification

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.

XP and agile principles

- ✧ Incremental development is 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.
- ✧ Maintaining simplicity through constant refactoring of code.

THE EXTREME PROGRAMMING RELEASE CYCLE

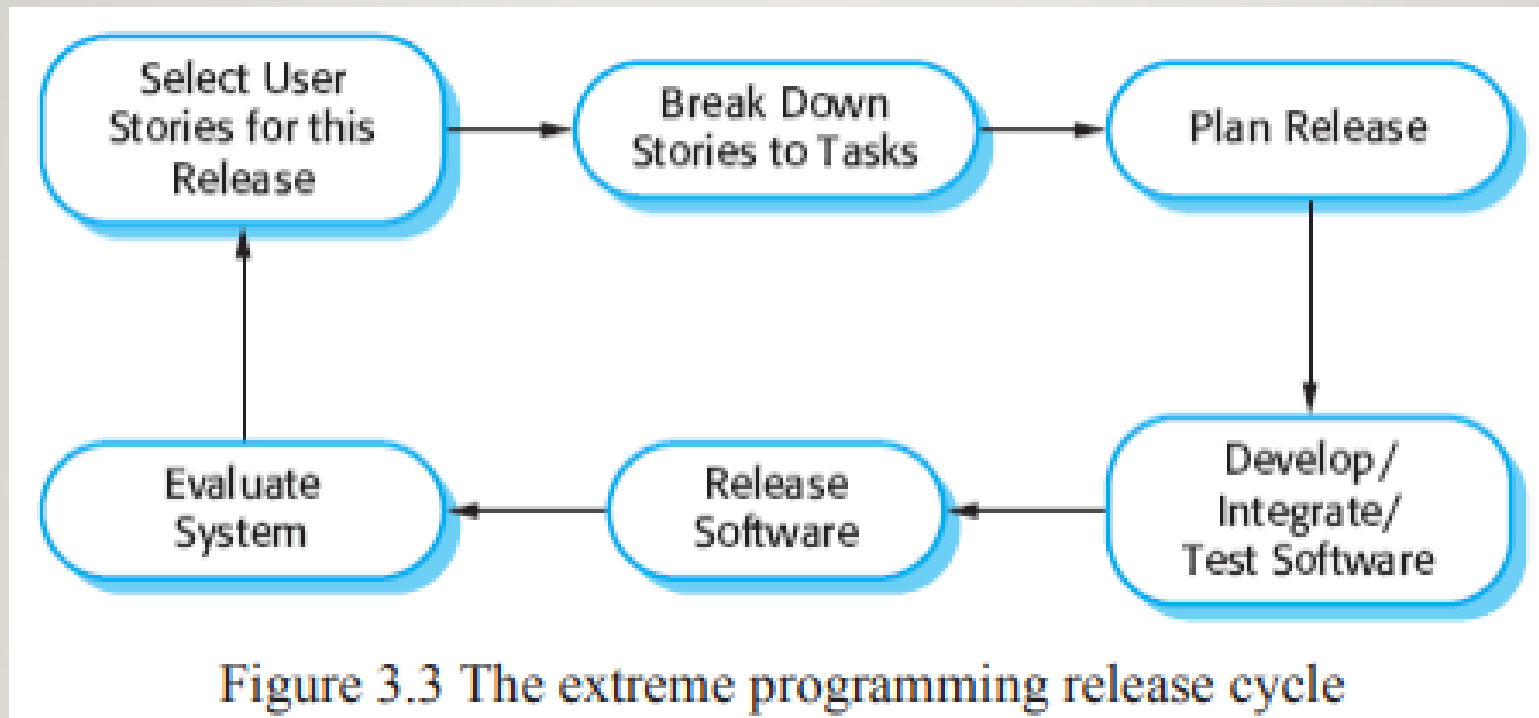


Figure 3.3 The extreme programming release cycle

Requirements scenarios

- ✧ In XP, a customer or user is part of the XP team and is responsible for making decisions on requirements.
- ✧ User requirements are expressed as scenarios or user stories.
- ✧ These are written on cards and the development team break them down into 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.

A 'PRESCRIBING MEDICATION' STORY

Prescribing medication

The record of the patient must be open for input. Click on the medication field and select either 'current medication', 'new medication' or 'formulary'.

If you select 'current medication', you will be asked to check the dose; If you wish to change the dose, enter the new dose then confirm the prescription.

If you choose, 'new medication', the system assumes that you know which medication you wish to prescribe. Type the first few letters of the drug name. You will then see a list of possible drugs starting with these letters. Choose the required medication. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

If you choose 'formulary', you will be presented with a search box for the approved formulary. Search for the drug required then select it. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

In all cases, the system will check that the dose is within the approved range and will ask you to change it if it is outside the range of recommended doses.

After you have confirmed the prescription, it will be displayed for checking. Either click 'OK' or 'Change'. If you click 'OK', your prescription will be recorded on the audit database. If you click 'Change', you reenter the 'Prescribing medication' process.

EXAMPLES OF TASK CARDS FOR PRESCRIBING

M

Task 1: Change Dose of Prescribed Drug

Task 2: Formulary Selection

Task 3: Dose Checking

Dose checking is a safety precaution to check that the doctor has not prescribed a dangerously small or large dose.

Using the formulary ID for the generic drug name, look up the formulary and retrieve the recommended maximum and minimum dose.

Check the prescribed dose against the minimum and maximum. If outside the range, issue an error message saying that the dose is too high or too low. If within the range, enable the 'Confirm' button.

Figure 3.6 Examples of task cards for prescribing medication.

SCRUM

- ✧ ~~The Scrum approach is a general agile method is~~ but focus is 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 SCRUM PROCESS

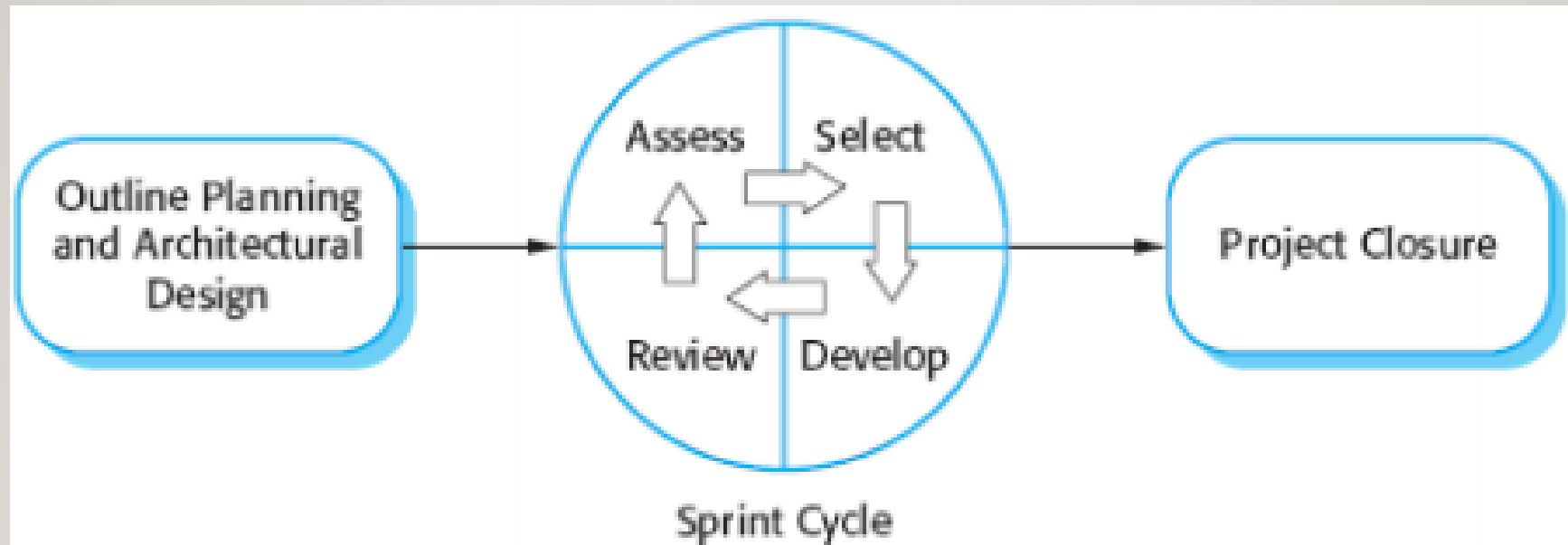


Figure 3.8 The Scrum process

The Sprint cycle

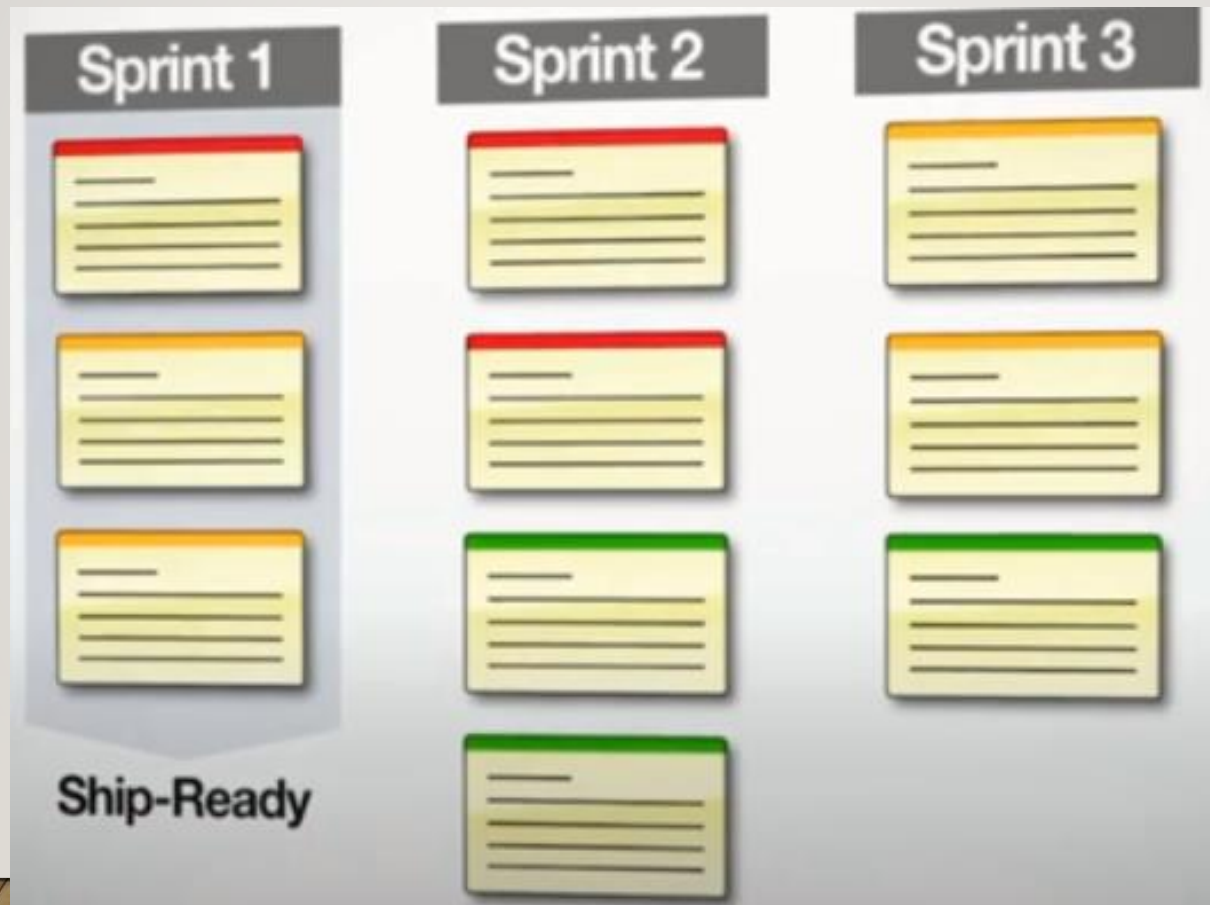
- ✧ Sprints are fixed length, normally 2–4 weeks. They correspond to the development of a release of the system in XP.
- ✧ The starting point for planning is the product backlog, which is the list of work to be done on the project.
- ✧ The selection phase involves all of the project team who work with the customer to select the features and functionality to be developed during the sprint.

THE SPRINT CYCLE

- ✧ ~~Once these are agreed, the team organize themselves to develop the software. During this stage the team is isolated from the customer and the organization, with all communications channelled through the so-called 'Scrum master'.~~
- ✧ The role of the Scrum master is to protect the development team from external distractions.
- ✧ At the end of the sprint, the work done is reviewed and presented to stakeholders. The next sprint cycle then begins.

PRODUCT BACKLOG





Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... DC 8 Test the... SC 8 Test the... SC 8 Test the... SC 6
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC 8 Test the... SC 6

TEAMWORK IN SCRUM

- ✧ The 'Scrum master' is a facilitator who arranges daily meetings, tracks the backlog of work to be done, records decisions, measures progress against the backlog and communicates with customers and management outside of the team.
- ✧ The whole team attends short daily meetings where all team members share information, describe their progress since the last meeting, problems that have arisen and what is planned for the following day.
 - This means that everyone on the team knows what is going on and, if problems arise, can re-plan short-term work to cope with them.

Scrum benefits

- ✧ The product is broken down into a set of manageable and understandable chunks.
- ✧ Unstable requirements do not hold up progress.
- ✧ The whole team have visibility of everything and consequently team communication is improved.
- ✧ Customers see on-time delivery of increments and gain feedback on how the product works.
- ✧ Trust between customers and developers is established and a positive culture is created in which everyone expects the project to succeed.

Key points

- ✧ A particular strength of extreme programming is the development of automated tests before a program feature is created. All tests must successfully execute when an increment is integrated into a system.
- ✧ The Scrum method 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.
- ✧ Scaling agile methods for large systems is difficult. Large systems need up-front design and some documentation.