

Software Engineering

***Summarized & Presented
By Dr.Engineer***

IBRAHIM ESKANDAR





What are the key challenges facing software engineering?

Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.

What are the costs of software engineering?

Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.

What are the best software engineering techniques and methods?

While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. There are no methods and techniques that are good for everything.

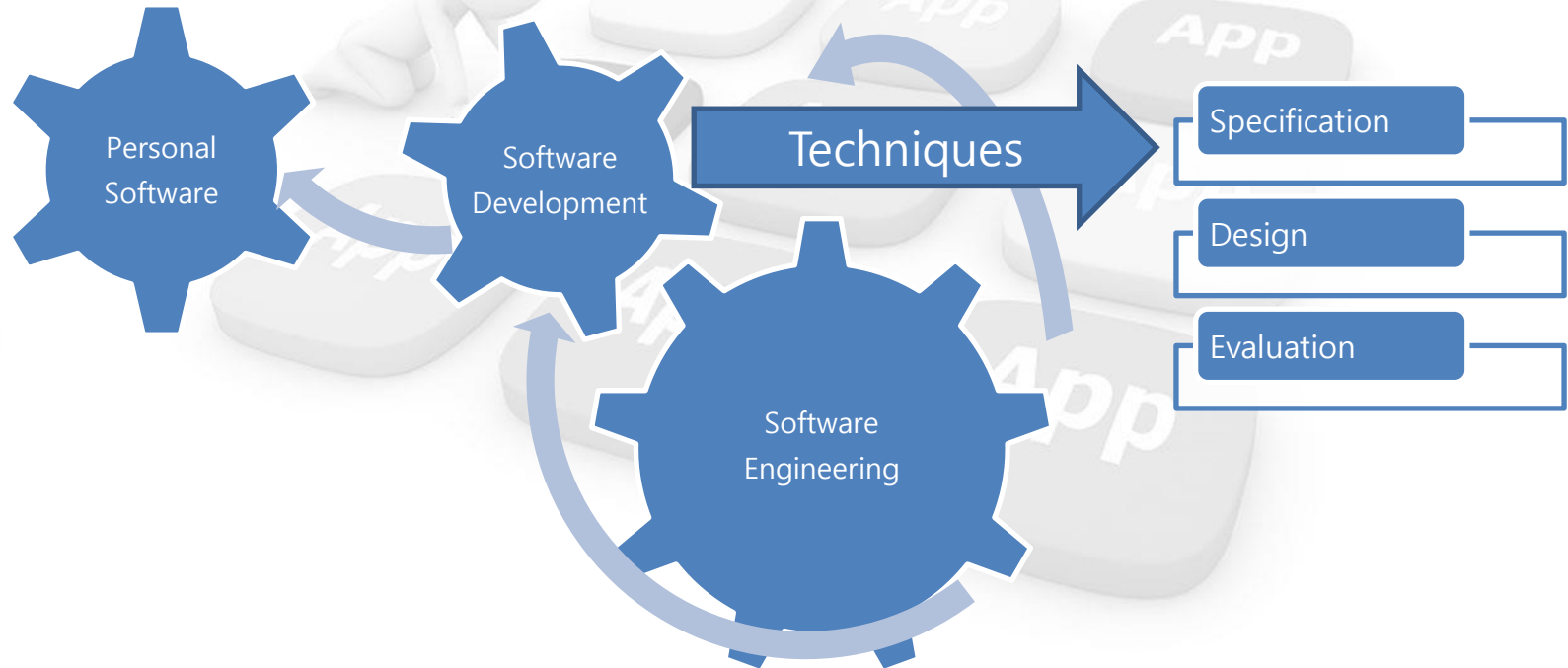
What differences has the Internet made to software engineering?

Not only has the Internet led to the development of massive, highly distributed, service-based systems, it has also supported the creation of an "app" industry for mobile devices which has changed the economics of software.

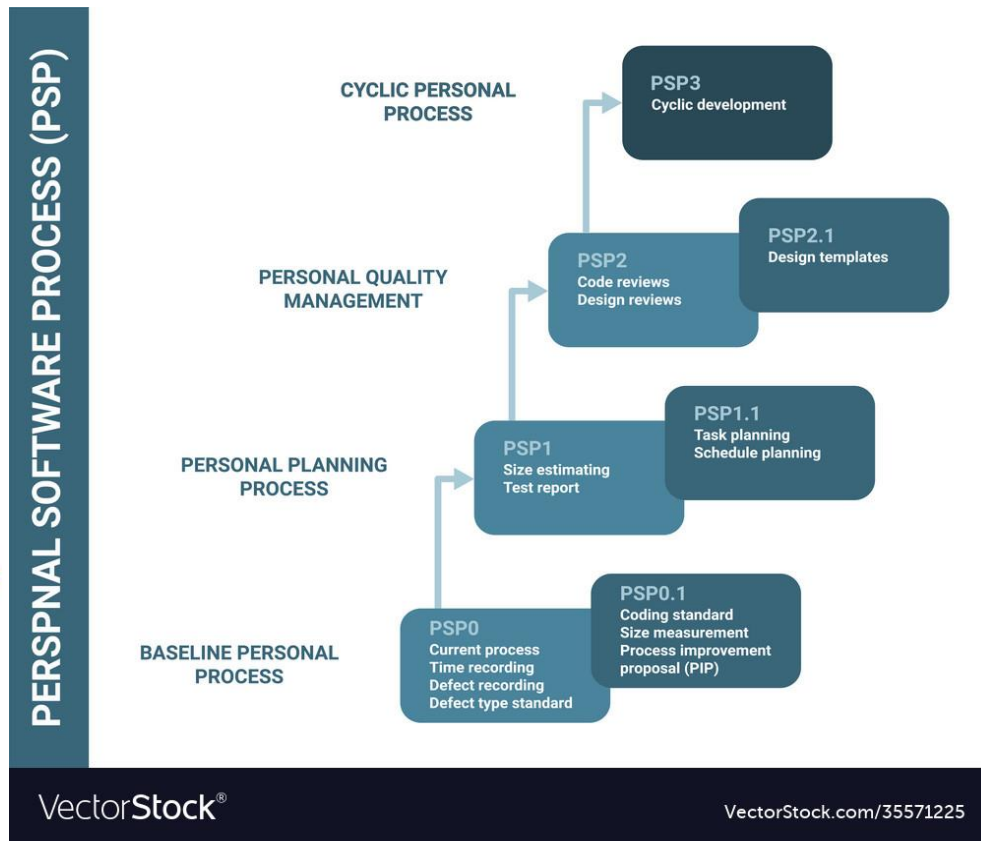


What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

Software development is a complicated process because of the issues and because of the maintained and changed throughout its life.



Personal Software Process (PSP)



PSP Activity
Postmortem
Development
Design Review
Design
Planning

Personal Software Process (PSP)

The Personal Software Process

The PSP process is designed for individual use.

It is based on scaled-down industrial software practice.

The PSP course demonstrates the value of using a defined and measured process.

It helps you and your organization meet the increasing demands for high quality and timely software.



Team Software Process (TSP)

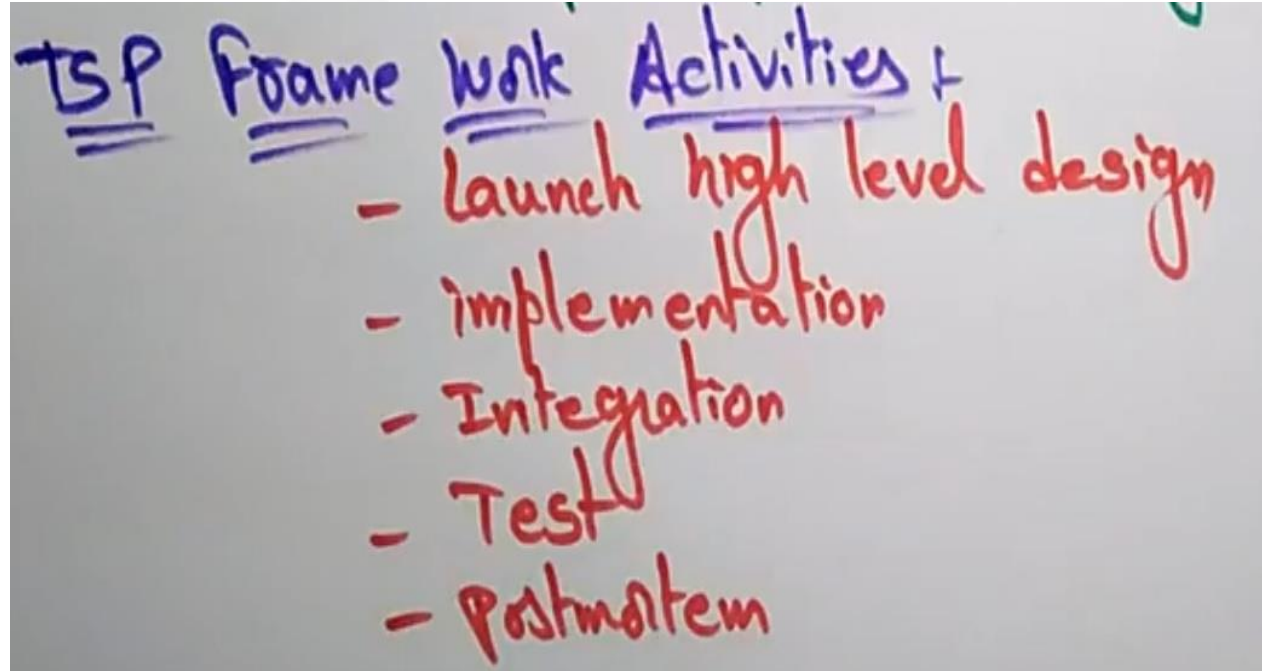


Team Software Process (TSP)

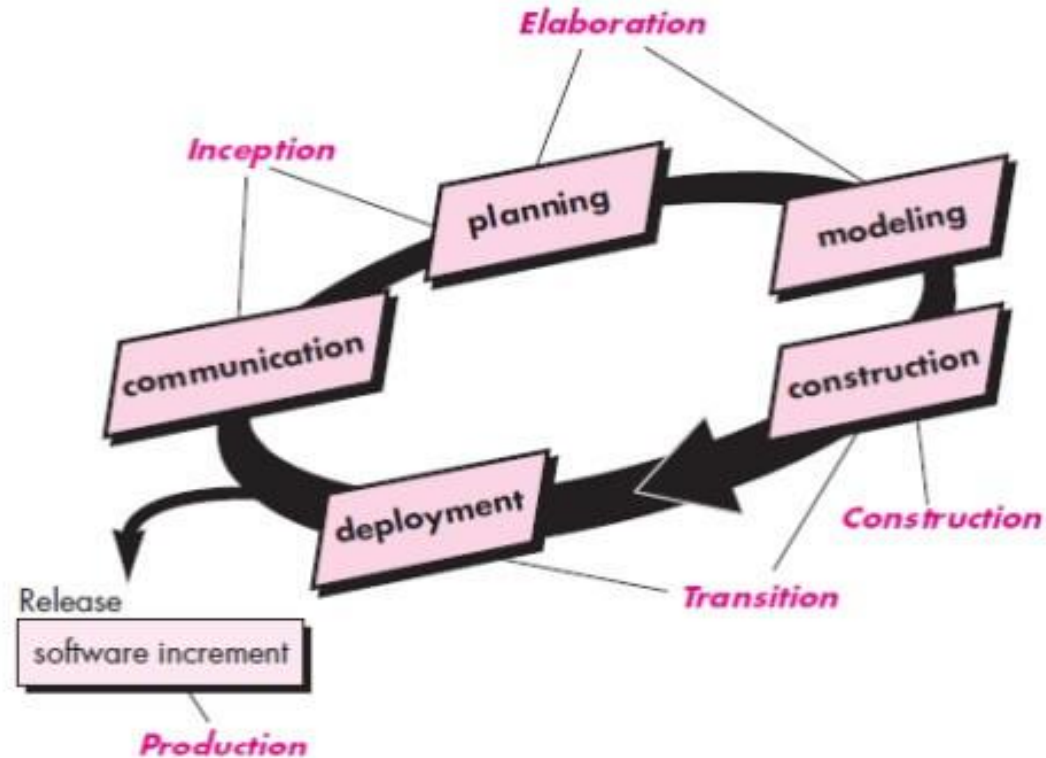
PSP & TSP, helps the high performance engineers to,

- ensure quality s/w product
- create secure s/w product
- improve process management in an organization.

Team Software Process (TSP)



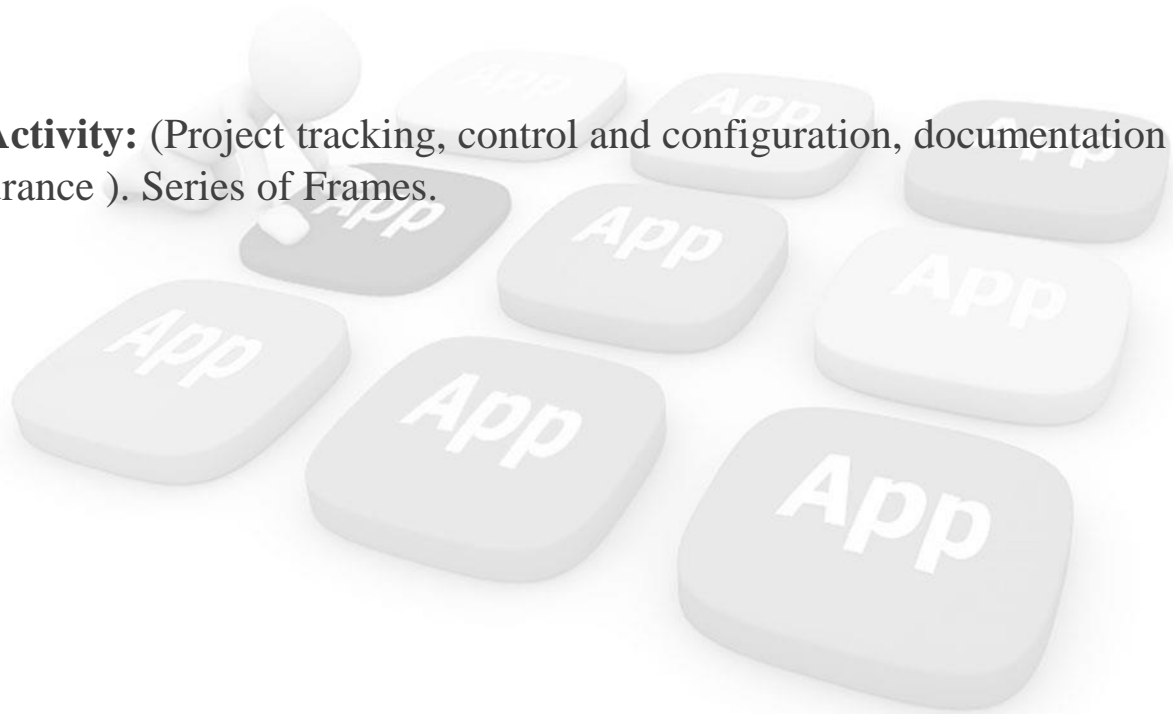
Unified Modeling Process (TSP)



Software Process

Process Framework: Generic activity that are applicable to all projects (size complexity planning modelling construction).

Umbrella Activity: (Project tracking, control and configuration, documentation and quality assurance). Series of Frames.





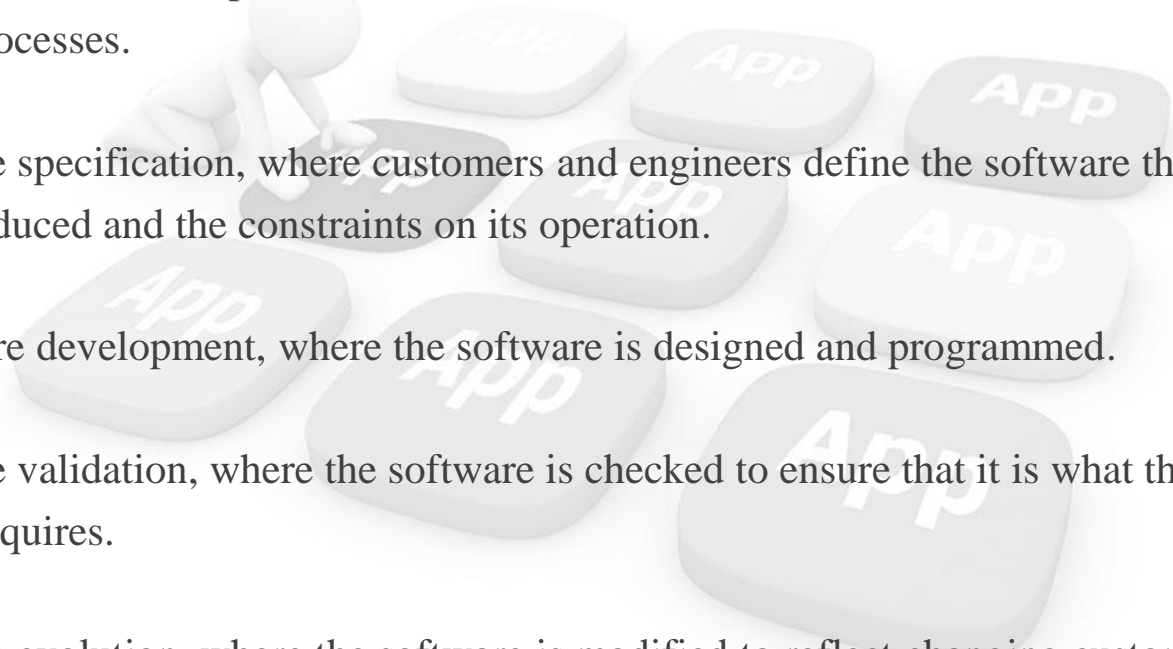
Professional Software Development

- 1- **Generic:** These are stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them. Examples of this type of product include apps for mobile devices, software for PCs such as databases, word processors, drawing packages, and project management tools.
- 2- **Customized:** A software contractor designs and implements the software especially for that customer. Examples of this type of software include control systems for electronic devices, systems written to support a particular business process, and air traffic control systems.

Software Engineering Process

The systematic approach that is used in software engineering is sometimes called a software process. A software process is a sequence of activities that leads to the production of a software product. Four fundamental activities are common to all software processes.

1. Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.
2. Software development, where the software is designed and programmed.
3. Software validation, where the software is checked to ensure that it is what the customer requires.
4. Software evolution, where the software is modified to reflect changing customer and market requirements.



Product characteristic	Description
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use.
Dependability and security	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Software has to be secure so that malicious users cannot access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, resource utilization, etc.
Maintainability	Software should be written in such a way that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.





Software Engineering Ethics

Confidentiality: You should normally respect the confidentiality of your employers or clients regardless of whether or not a formal confidentiality agreement has been signed.

Competence: You should not misrepresent your level of competence. You should not knowingly accept work that is outside your competence.

Intellectual property rights: You should be aware of local laws governing the use of intellectual property such as patents and copyright. You should be careful to ensure that the intellectual property of employers and clients is protected.

Computer misuse: You should not use your technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine) to extremely serious (dissemination of viruses or other malware).

SWE Main Process



Software specification The functionality of the software and constraints on its operation must be defined.

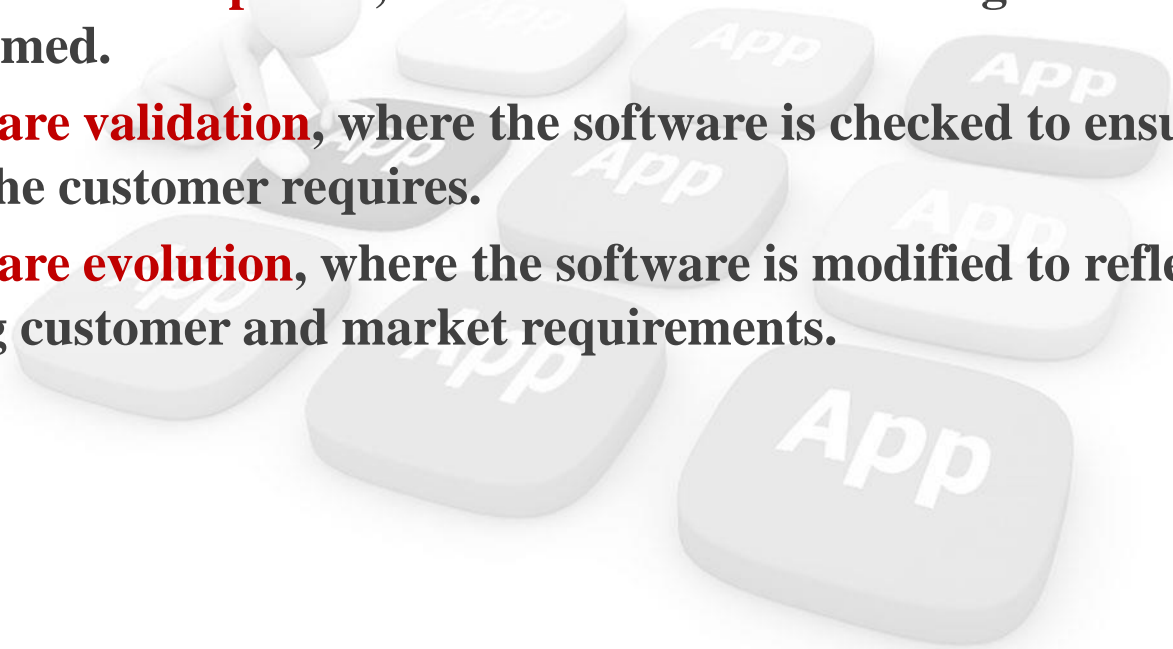
Software development The software to meet the specification must be produced.

Software validation The software must be validated to ensure that it does what the customer wants.

Software evolution The software must evolve to meet changing customer needs.



1. **Software specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
2. **Software development**, where the software is designed and programmed.
3. **Software validation**, where the software is checked to ensure that it is what the customer requires.
4. **Software evolution**, where the software is modified to reflect changing customer and market requirements.



SWE Main Process

When we describe and discuss processes, we usually talk about the activities in these processes, such as specifying a data model and designing a user interface, and the ordering of these activities.

However, when describing processes, it is also important to describe who is involved, what is produced, and conditions that influence the sequence of activities:

- 1. Products or deliverables are the outcomes of a process activity.**
- 2. Roles reflect the responsibilities of the people involved in the process.**
- 3. Pre- and post conditions are conditions that must hold before and after a process activity has been enacted or a product produced.**

Software Process Method

1- 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, and testing.

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.

3- Integration and configuration This approach relies on the availability of reusable components or systems. The system development process focuses on configuring these components for use in a new setting and integrating them in to a system.

3- Agile: Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Software Process Method - Agile

Agile is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like:

Planning

Requirements Analysis

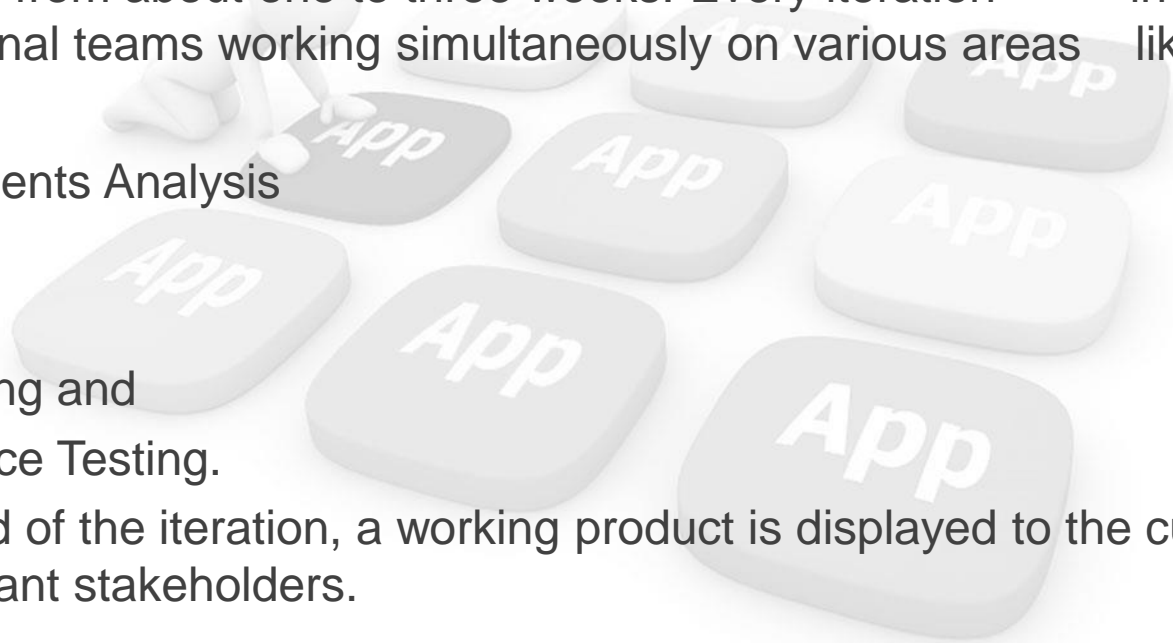
Design

Coding

Unit Testing and

Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

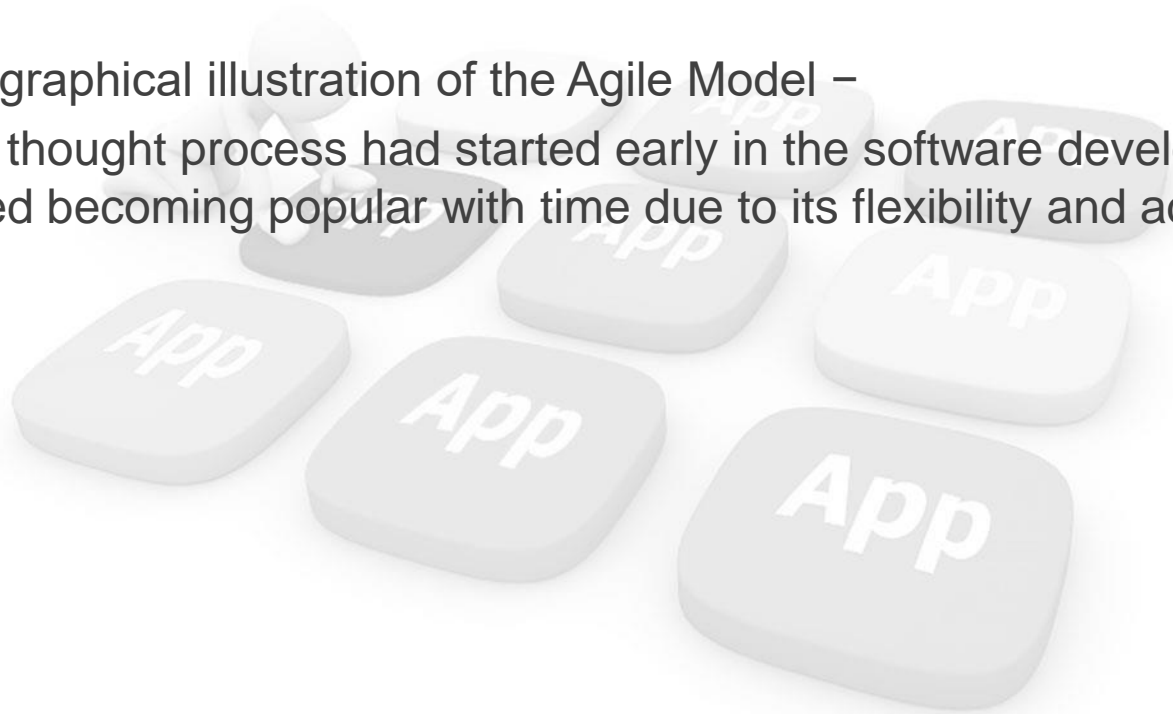


Software Process Method - Agile

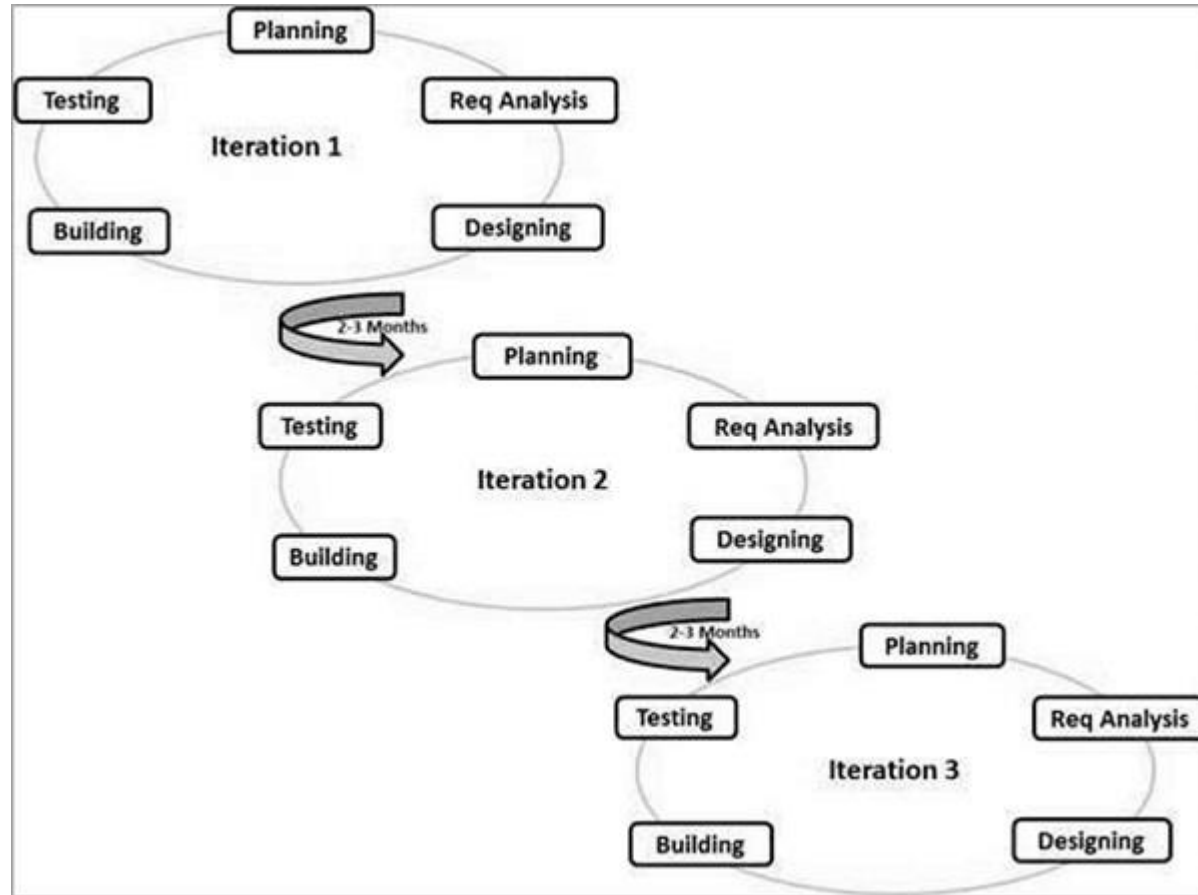
Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model –

The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.



Software Process Method - Agile



Software Process Models

Incremental development has three major advantages over the waterfall model:

- 1. The cost of implementing requirements changes is reduced.** The amount of analysis and documentation that has to be redone is significantly 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. Early 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.

Software Process Models

From a management perspective, the 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.** Regular change leads to messy code as new functionality is added in whatever way is possible. It becomes increasingly difficult and costly to add new features to a system. To reduce structural degradation and general code messiness, agile methods suggest that you should regularly refactor (improve and restructure) the software.

Software Process Models **Integration and configuration**

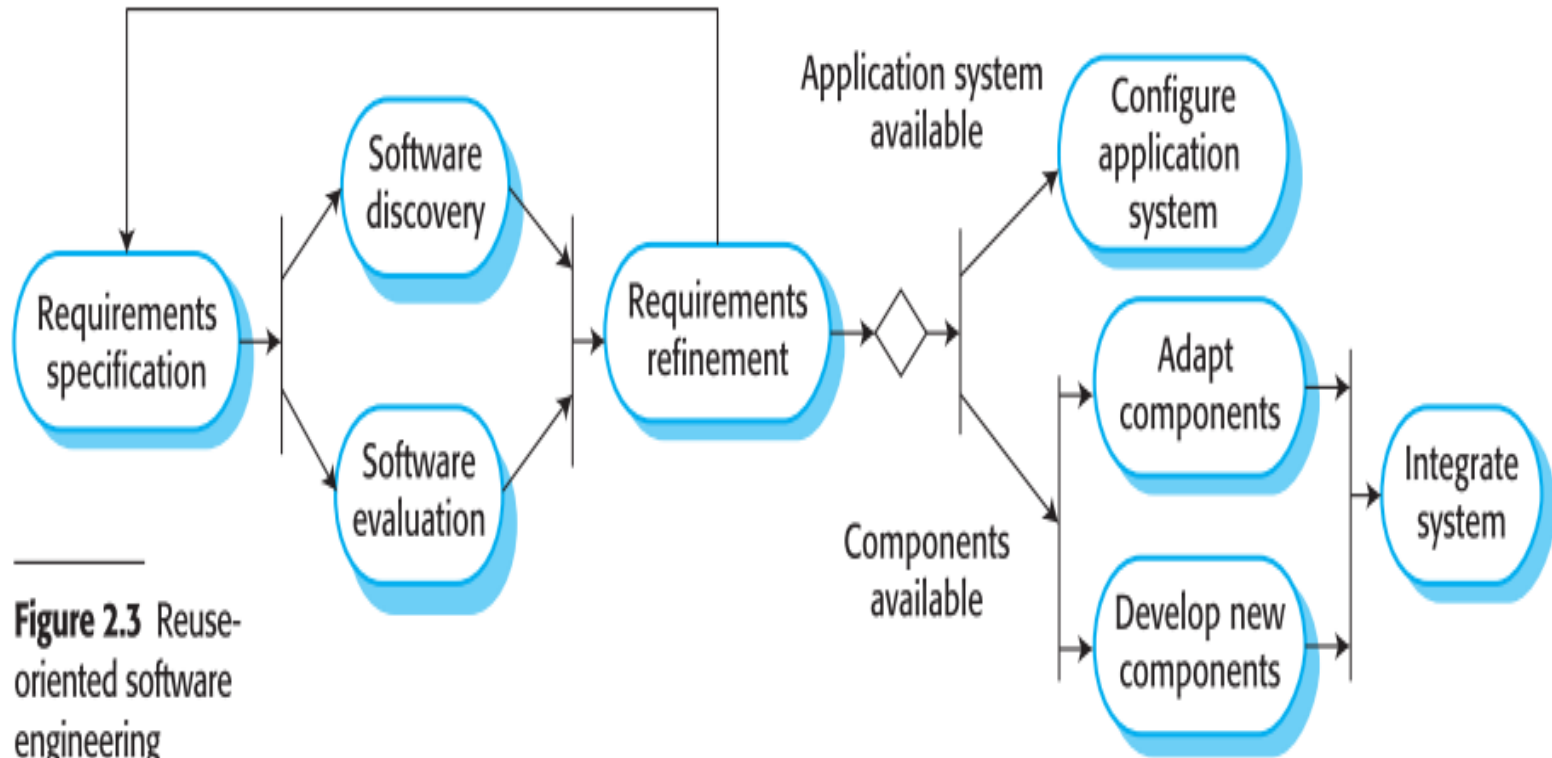


Figure 2.3 Reuse-oriented software engineering