specification.

What is Software Engineering?

Various Definitions

Software Engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

Software Engineering (IEEE standard): (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

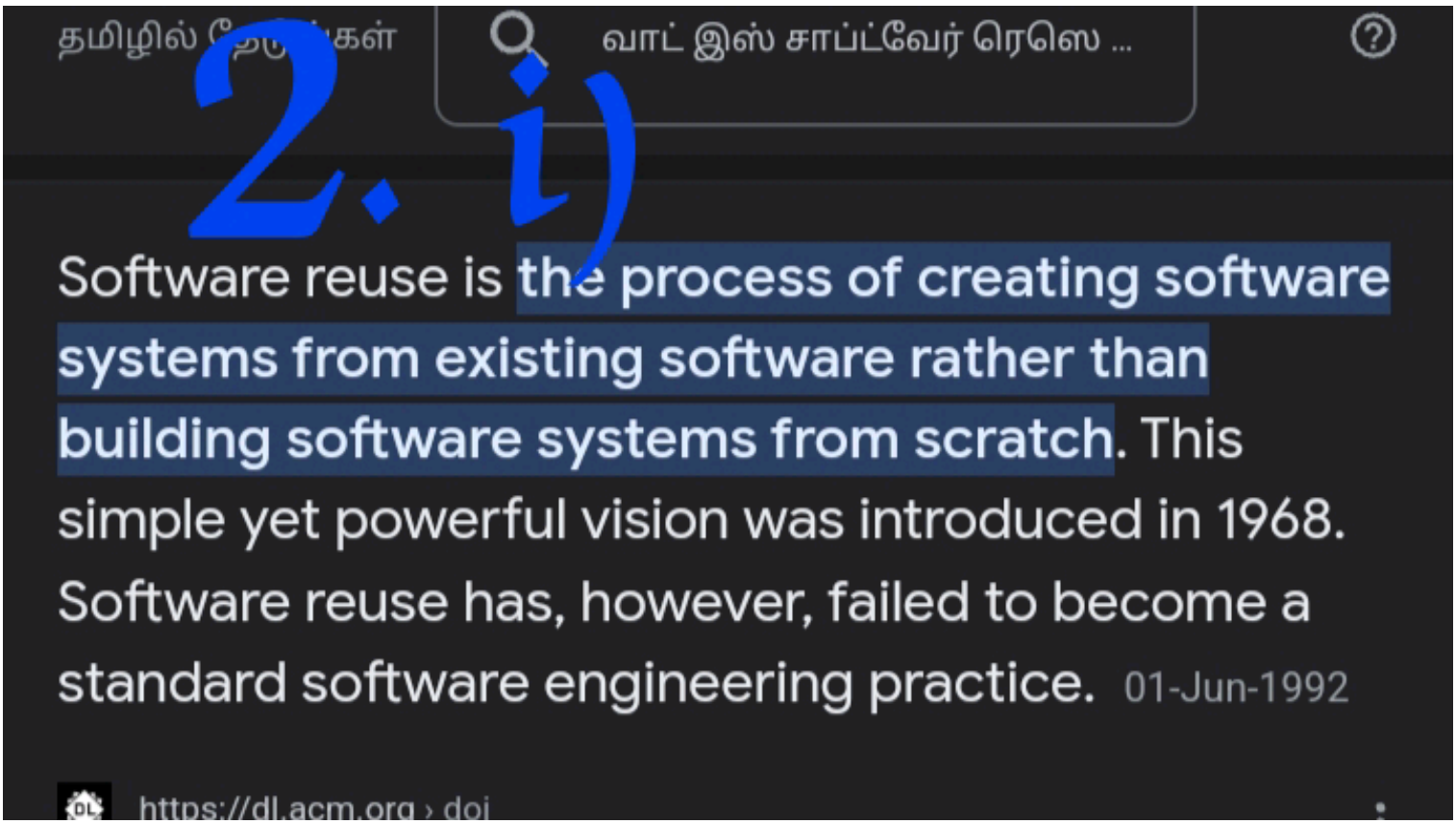
1

Software Engineering is concerned with software systems developed by teams rather than individual programmers, uses engineering principles in the development of this system, and is made up of both technical and non – technical aspects.

Software Engineering is a discipline that integrates methods, tools and procedures for the development of computer software.

Software Engineering is an emerging discipline that focuses on the creation, development, operation, and maintenance of cost-effective, reliably correct, and high-quality solutions to software problems.

Software Engineering is defined as a systematic approach to developing and maintain a software product in a cost-effective and efficient way.


தமிழில் தேடுங்கள்
வாட் இஸ் சாப்ட்வேர் ரெஸெ ...
2. i)
Software reuse is the process of creating software systems from existing software rather than building software systems from scratch. This simple yet powerful vision was introduced in 1968. Software reuse has, however, failed to become a standard software engineering practice. 01-Jun-1992
https://dl.acm.org › doi

Five Benefits Of Reusing Code For Software Development

- Faster Development Periods. ...
- Reduced Development Spending. ...
- Lower Development Risks. ...
- Maintain Product Standards. ...
- Great Product Potential.

03-Aug-2021

1.2 SOFTWARE CHARACTERISTICS

The software is a logical rather than a physical system element. The various software characteristics are:-

1. **The software is developed or engineered, it is not manufactured:** Unlike hardware, software is logical rather than physical. It has to be designed well before producing it. In spite of the availability of many automated software development tools, it is the skill of the individual, creativity of the developers and proper management by the project manager that counts for a good software product.

2. **Software does not "wear out":** As time progresses, the hardware components start deteriorating-they are subjected to environmental maladies such as dust, vibration, temperature etc. and at some point of time, they tend to break down. The defected components can then be traced and replaced. But, software is not susceptible to environmental changes. So, it does not wear out. The software works exactly the same way even after years it was first developed unless any charges are introduced to it. The changes in the software may occur due to the changes in the requirements. And these changes may introduce some defects in it thus, deteriorating the quality of software, So, software needs to maintained properly.

3. **Most software is custom-built, rather than being assembled from existing components:** Most of the engineered products are first designed before they are manufactured, Designing includes identifying various components for the product before they are actually assembled. Here several people can work independently on these components thus making the manufacturing system highly flexible. In software, breading a program into modules is a difficult task, since each module is highly interlinked with other modules. Further, it requires a lot of skill to integrate different modules into one. Nowadays the term component is widely used in the software industry where the object-oriented system is in use.

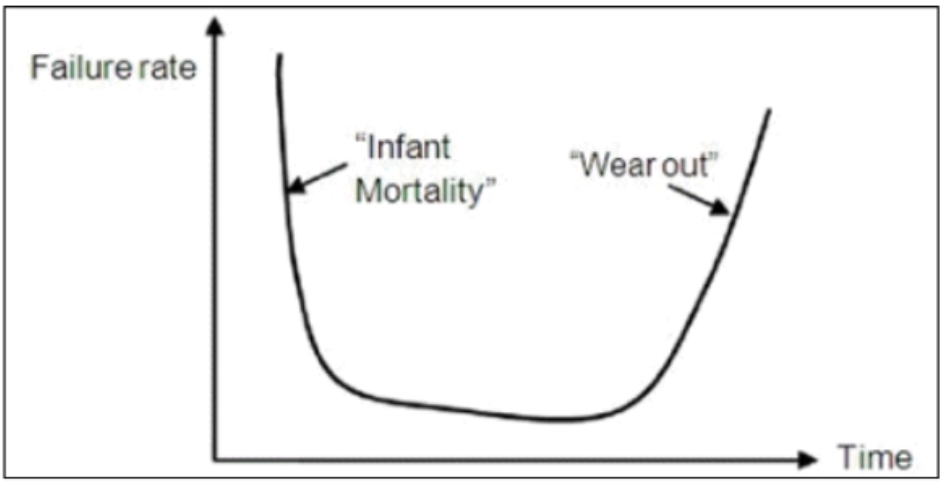


Fig. 1.1 Failure Curves for Hardware

Difference Between

4

AGILE VERSUS WATERFALL

AGILE	WATERFALL
It takes an iterative approach that is derived from Lean thinking and it allows changing the scope of the project.	It takes a sequential approach to software development where a project is divided into different phases.
It works well when the scope of the project is unknown.	It works well when the scope of the project is known beforehand.
It allows changes to be made as it starts with a simple design which is divided into small modules.	The contract terms won't allow changes to be made as the process is sequential.
Customer availability is important through the entire project.	Customer availability is required only at set milestones.
It allows partial success as valuable features are implemented first thereby decreasing risk of failure.	It doesn't allow partial success which increases the risk of failure.
Products are tested thoroughly for bugs and errors during the development cycle.	Testing cannot be done during the development cycle, but only at the end.
It provides flexibility to oversee the development project.	Flexibility is limited as products are based upon documented requirements.

2. **Software does not "wear out":** As time progresses, the hardware components start deteriorating-they are subjected to environmental maladies such as dust, vibration, temperature etc. and at some point of time, they tend to break down. The defected components can then be traced and replaced. But, software is not susceptible to environmental changes. So, it does not wear out. The software works exactly the same way even after years it was first developed unless any charges are introduced to it. The changes in the software may occur due to the changes in the requirements. And these changes may introduce some defects in it thus, deteriorating the quality of software, So, software needs to maintained properly.

Functional Requirements	Non Functional Requirements
A functional requirement defines a system or its component.	A non-functional requirement defines the quality attribute of a software system.
It specifies “What should the software system do?”	It places constraints on “How should the software system fulfill the functional requirements?”
Functional requirement is specified by User.	Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers.
It is mandatory.	It is not mandatory.
It is captured in use case.	It is captured as a quality attribute.
Defined at a component level.	Applied to a system as a whole.
Helps you verify the functionality of the software.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.

What is a feasibility study in software engineering?

In software engineering, the feasibility study is **a project done to determine software's technical and commercial viability before its development**. The project evaluates the system's impact, functionality, and performance before investing in its development. 21-Feb-2023

7

People also ask

8

What is the purpose of system requirements?

System requirements is a statement that **identifies the functionality that is needed by a system in order to satisfy the customer's requirements**. [1] System requirements are a broad and also narrow subject that could be implemented to many items.

8. Traceability: The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

There are two types of Traceability:

1. Backward Traceability: This depends upon each requirement explicitly referencing its source in earlier documents.

2. Forward Traceability: This depends upon each element in the SRS having a unique name or reference number.

2.4 Data Dictionary

The data dictionary is an organized testing of all data elements that are relevant to the system so that both the user and system analyst will have a common understanding inputs, outputs, components of stores and intermediate calculations. It is proposed as quasi- formal grammar during structured analysis and may contain the following information.

- Name of the data item.
- Aliases (Other names for items)
- Description/Purpose – a notation for representing its goal or content.
- Related data items
- Range of value

10

- Data structure definition / form
- Where used / how-used (eg.) input to the process, output from the process, as a store and external entity.
- Supplementary information – data types, preset values, restrictions or limitations.

A data dictionary is simplistically an alphabetic list of names which

People also ask

11. i)

What are the 7 steps in design process?

The 7 steps of the design process

- Define the problem. Crucial to solving any design problem is to begin by asking the right questions. ...

11. ii)

People also ask

What is the purpose of design process software engineering?

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

Software Engineering | Architectural Design

Introduction: The software needs the architectural design to represents the design of software. IEEE defines architectural design as "the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system." The software that is built for computer-based systems can exhibit one of these many architectural styles.
Each style will describe a system category that consists of :