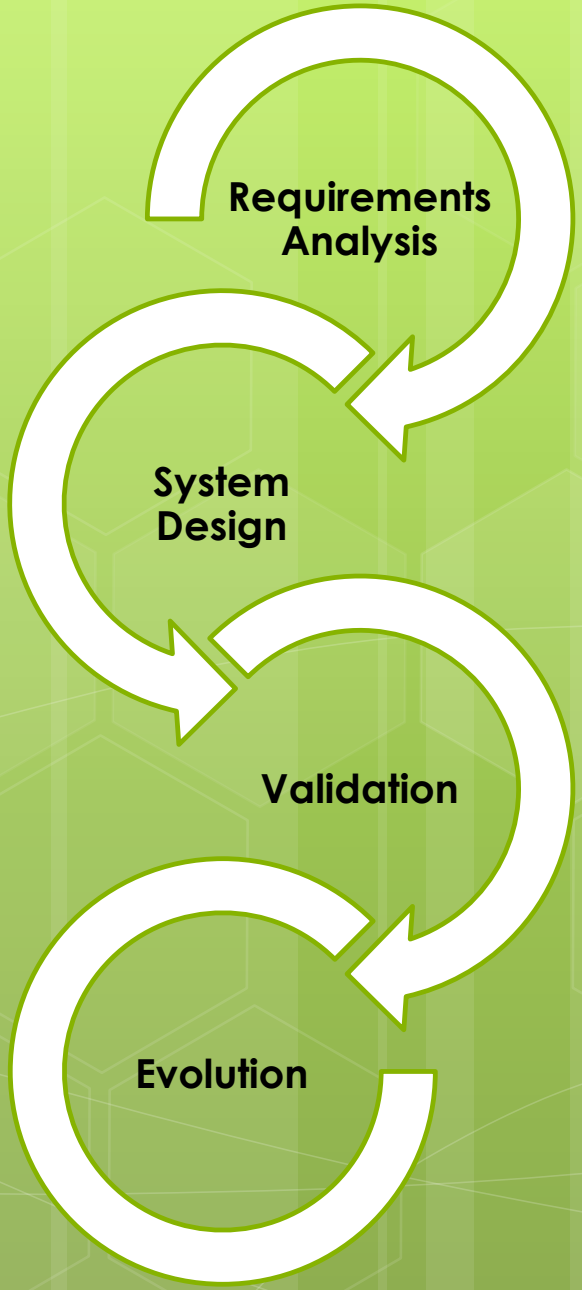


Phases of Software Engineering

REQUIREMENTS ANALYSIS

- Tasks determining the needs or conditions to meet for a new or altered product, taking account of the possibility conflicting requirements of the various stakeholders, such as beneficiaries or users.

- It includes three types of activity:

  - **Eliciting Requirements**
    - The task communicating with customers and users to determine what their requirements are. This is sometimes called as *Requirements Gathering*.

  - **Analyzing Requirements**
    - Determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolved these issues.

  - **Recording Requirements**
    - Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

REQUIREMENTS ANALYSIS

- **Feasibility Study**
  - The study considers whether the proposed system will be cost effective from a business point of view.
  - An estimate of whether the identified user needs may be satisfied using current software and hardware technologies.

Functional and Non-functional Requirements

Functional Requirements

- describe what the system should do.

Example:

1. A user shall be able to search the appointments lists for all clinics.

2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

Non-functional Requirements

- Specify the characteristics of the system as a whole.
- More often more critical than individual functional requirements.

Example:

Reliability, response time

Seatwork

- Identify the functional and non-functional requirements of your proposed system.

SYSTEM DESIGN

- It is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.
- It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system
- Systems design implies a systematic approach to the design of a system.
- It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system.
- Systems design then overlaps with systems analysis, systems engineering and systems architecture.

VALIDATION

- Also called as “Software Validation and Verification”
- **Validation** refers to the process of evaluating software at the end of its development to ensure that it is free from failures and complies with its requirements.
  - A **failure** is defined as an incorrect product behaviour.
- **Verification** refers to the process of determining whether or not the products of a given phase of a software development process fulfil the process established during the previous phase.
- **Software V&V Approaches**
  - Software Technical Reviews
  - Software Testing
  - Proof of Correctness
  - Simulation and Prototyping
  - Requirements Tracing

EVOLUTION

- The process of developing a software product using software engineering principles and methods is referred to as *software evolution*.
- This includes the initial development of software and its maintenance and updates, till desired software product is developed, which satisfies the expected requirements.

- This processes include:
  - Change Request
  - Impact Analysis
  - Release Planning
  - System Update
  - System Release

- Evolution starts from the requirement gathering process.
- After which developers create a prototype and show it to the users for their feedback at the early stage of software development.
- The users suggest changes, on which several consecutive updates and maintenance keep on changing.
- This process changes to the original software, till the desired software is accomplished.

People in the Development Effort

People in the Development Effort

1. **Development Team**
   - responsible in making the software
   - Build/create the software.

Members of the Development Team

- **Analyst**
  - Understands the system
- **System Designer**
  - Defines the software architecture, components, modules, interfaces.
- **Programmer**
  - write codes of the system in a specific programming language
- **Tester**
  - reviews faults and errors of the system

People in the Development Effort

2. End – users

- used/ utilize the system

- **Directly Involved in the system**
  - ***Operational Job***
    - Performs the daily task/operation of the organization

    - *Concern: How does it work?*

People in the Development Effort

2. End – users

- **Directly Involved in the system**
  - ***Supervisor***
    - Performs supervisory job
    - *Concern: Physical Interface & Performance*
  - ***Executive***
    - Provides finances and initiatives
    - Outputs/results
    - *Concern: global view of the system*

People in the Development Effort

2. End – users

- **Indirectly Involved in the system**
  - ***Quality Assurance (QA)***
    - part of **quality** management focused on providing confidence that **quality** requirements will be fulfilled.(ISO 9000)
    - groups and external organizations which set standards to be followed

Software Quality and Software Quality Assurance

Software Quality

Quality

- is the total characteristics of an entity to satisfy stated and implied needs .

- **Software Quality**
  - if it is fit for use

Three Perspectives

1. **Quality of the Product**

- relative to the person analyzing quality.

***end-users**

- if it gives what they want and when they want it, all the time.

*** dev't team**

- errors and faults found during the SE process

Three Perspectives

2. **Quality of the Process**
   - failures of task are avoided
   - system development process is improved

3. **Quality in the Context of business Environment**
   - in terms of products/services provided by the business in which the software is used.

   - software adds value to the business

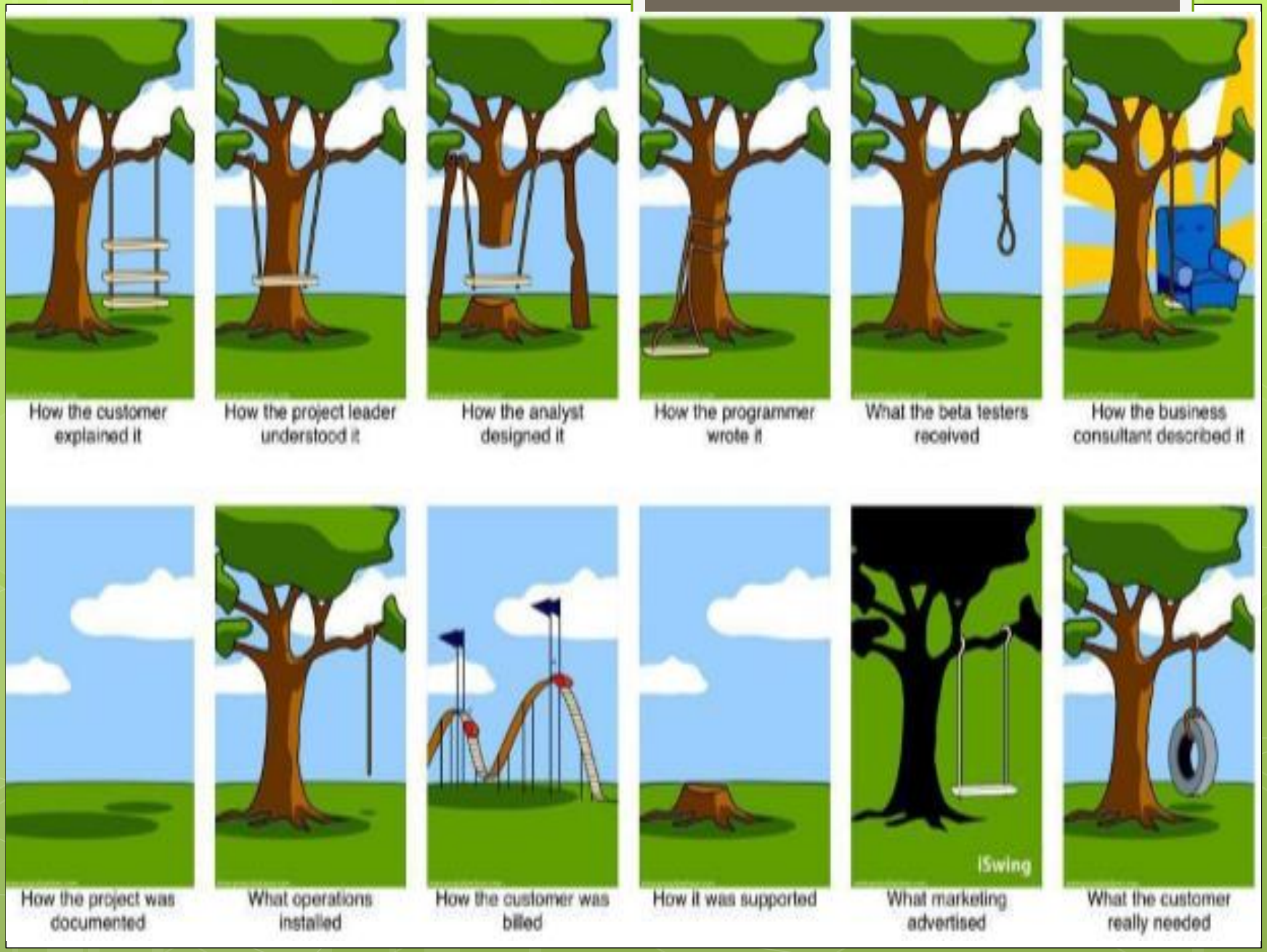
Software Quality Assurance

- ensures that the software product complies with user requirements and standards.

What are the characteristics of a well-engineered software?

- Usability
- Portability
- Reusability
- Maintainability
- Dependability
- Efficiency

Large projects are more of a challenge of organization and communication than they are of programming.


How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
What the beta testers received
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
iSwing
What marketing advertised
What the customer really needed

Interpersonal Communication Rules

1. Respect your teammates, even when you disagree or they are wrong.
2. Accept group decision even when you disagree.
3. You must include all group members in decisions.
4. You should do your best to contribute to your team.
5. Email or text is not a good medium for resolving problems.

Quiz2