Program

- **A program :** is a set of instructions (written in form of human-readable code) that performs a specific task.
- **Application:** A program or group of programs that is designed for the end user. Application software (an application) is a set of computer programs designed to permit the user to perform a group of coordinated functions, tasks, or activities.

What is software?

Computer Programs

+

associated

Documentation

Documentation Types

- user Documentation
- Technical Documentation
- Marketing Documentation

Software types

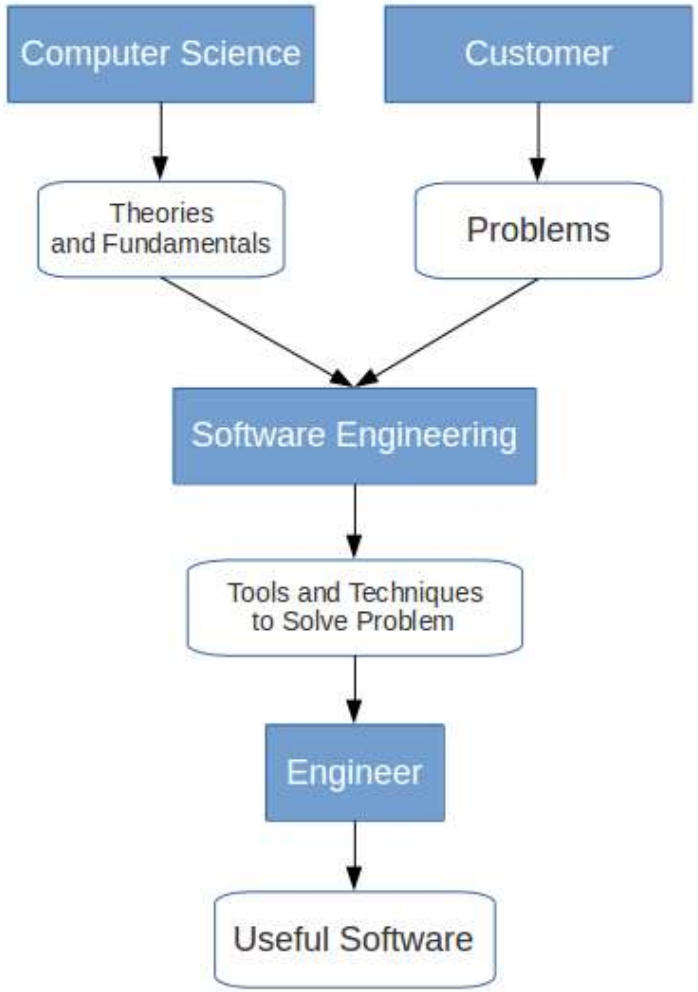
- Generic
- Customized

Attributes of Good Software

Software engineering vs computer science

- Computer science focuses on theory and fundamentals.

- Software engineering is concerned with the practicalities of developing and delivering useful software.



The Team


Onsite PM
TEST Team
T Lead
P Manager
SOLID
BORN BRAVE
Developer
BUG

Daily Software Engineering Jobs/ Roles

1. Systems Analyst
2. Designer
3. UI/UX Developer
4. Software Programmer
5. Software Systems Administrator
6. Software Database Administrator

7 Requirement Engineer

8 Software Security Engineer

9 Software Tester

10 Software Project Manager

11 Software Configuration Engineer

12 IT Help Desk

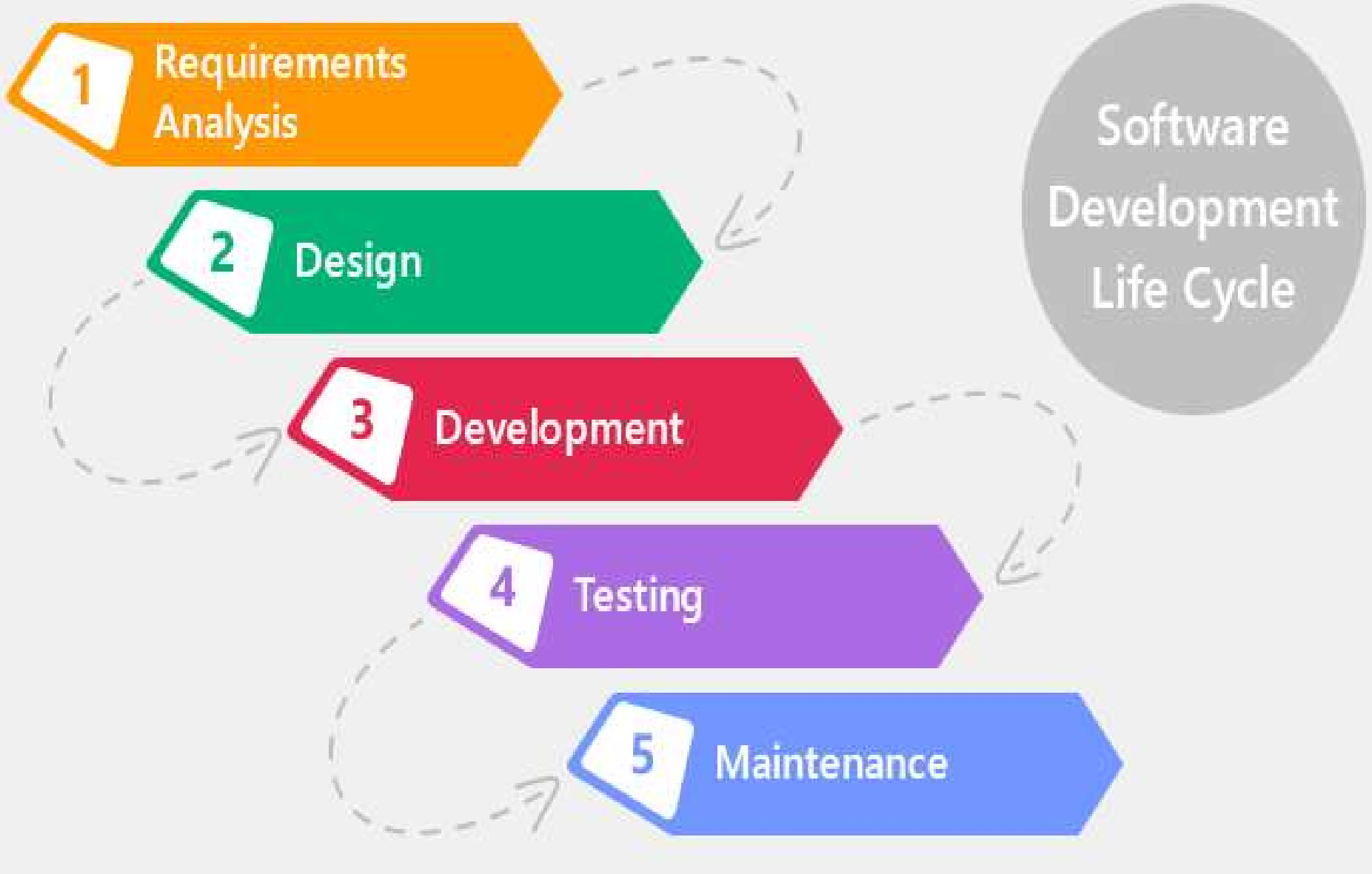
Position Levels

- trainee
- Junior
- Senior
- Team Leader
- Project Manager

Software Process

- A software process (also knows as software methodology) is a set of related activities that leads to the production of the software. These activities may involve the development of the software from the scratch, or, modifying an existing system.

Stages for software development



Requirement

- descriptions of what the system should do

the services that it provides and the constraints on its operation.

- These requirements reflect the needs of customers for a system that serves a certain purpose

Problems of requirements elicitation

- Stakeholders don’t know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- The requirements change during the analysis process. New stakeholders may emerge and the business environment change.

Levels of description Requirements

- **User requirements**： are statements, in a natural language, of what services the system is expected to provide to system users.

"The System shall generate monthly management reports showing the Profit for each product prescribed by each Branch during that month."

Levels of description Requirements

- **System requirements :** are more detailed descriptions of the software system's

functions, services, and operational constraints.

- should define exactly what is to be implemented.
- It may be part of the contract between the system buyer and the software developers.

System requirements

“1.1 On the last working day of each month, a summary of the products , their profit , and the branch shall be generated.

1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.

1.3 A report shall be created for each branch and shall list the individual Product names, the total number of this product , the number of sold, and the total Profit of this product .

1.4 If Product are available in Different sizes(e.g., 10 mg, 20 mg) separate reports shall be created for each Size unit.

1.5 Access to all profit reports shall be restricted to authorized users listed on a management access control list.”

System requirements classification

- ***Functional requirements*** *:* These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
- ***Non-functional requirements*** *:* These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process

Types of non-functional requirement

- Access
- Usability
- Cost
- Capacity
- Documentation
- recovery
- Efficiency
- Backup
- licensing
- Performance
- Platform compatibility
- Privacy
- Quality
- Response time
- Security
- Support
- Test
- Speed

software requirements document

- sometimes called the software requirements specification or SRS
- is an official statement of what the system developers should implement.
- It should include both the user requirements for a system and a detailed specification of the system requirements.

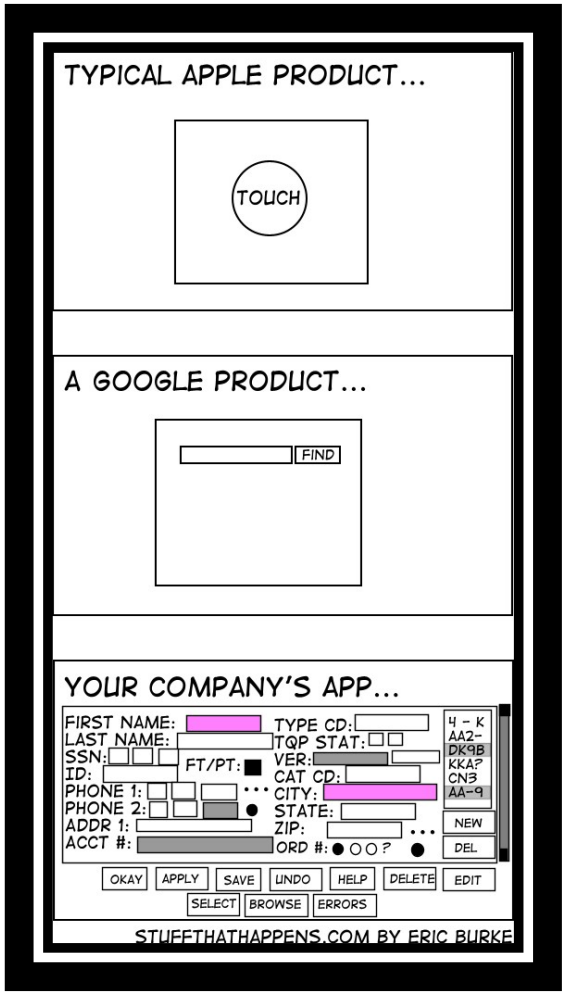
The structure of a requirements document

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.

The structure of a requirements document

Chapter	Description
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

2. Design



Planning the software solution

Modularity

- In this concept, software is divided into separately named and addressable components called modules
- Follows “divide and conquer” concept, a complex problem is broken down into several manageable pieces
- Easier to change
- Easier to build
- Easier to maintain

Refactoring

Refactoring is the process of changing a software system in
such a way that it does not alter the external behavior of the
code [design] yet improves its internal structure.
When software is refactored, the existing design is examined for

- redundancy
- unused design elements
- inefficient or unnecessary algorithms
- poorly constructed or inappropriate data structures
- or any other design failure that can be corrected to yield a better design.

Interface Design

- UI is the user interface. This comprises everything a user can see and touch, such as menu options, buttons, text, layouts, navigation elements, sharing options, etc

- UX is why you made that change to affect how the user feels and behaves. The user experience is an umbrella term for the user's overall experience with the product: what they liked about it and how easily they accomplished their goals

Interface Design

UI

UX

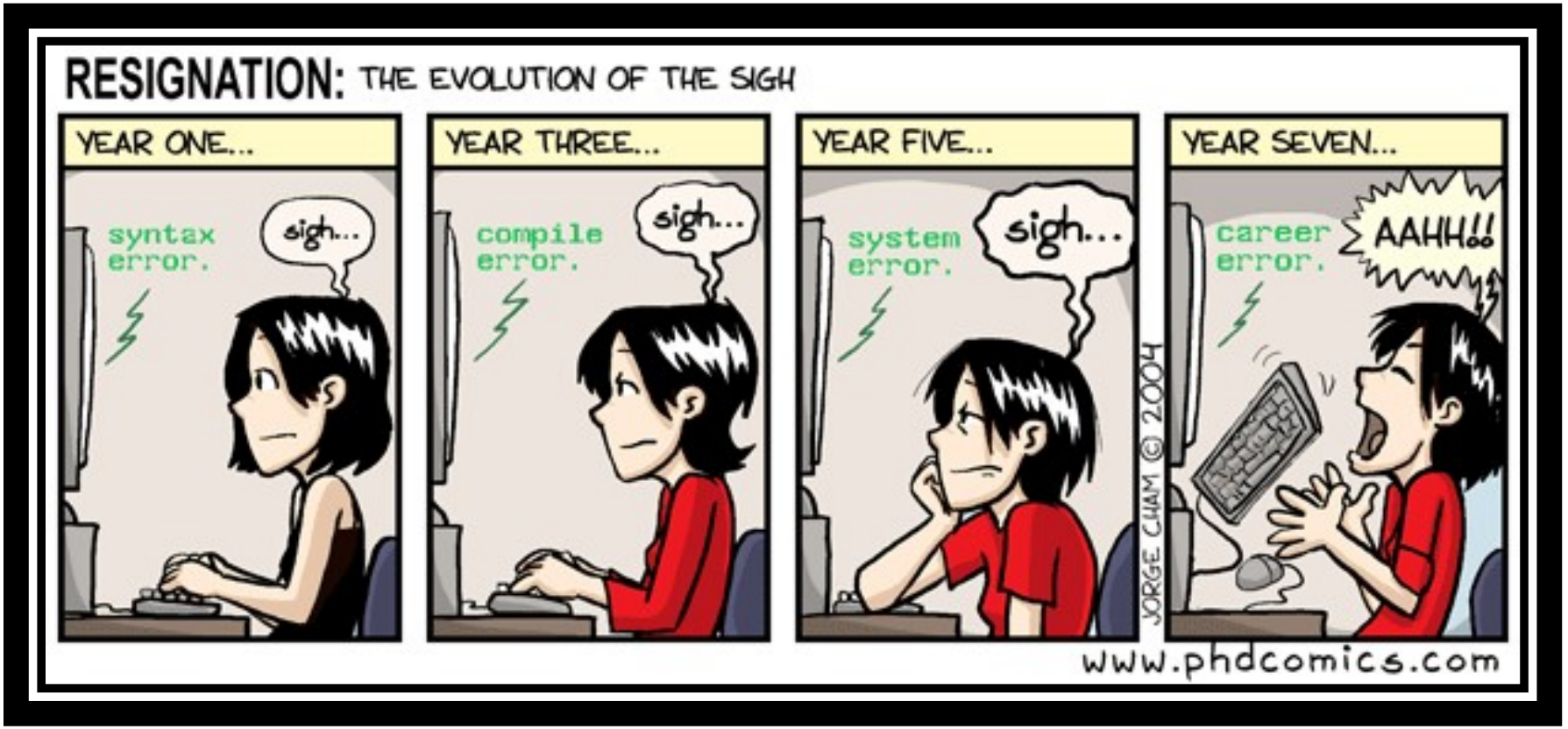
Open top, shake vigorously

Pray it doesn't get all over you

Open top

Squeeze out desired amount

3. Implementation



Code!!!

4. Testing

Executing the application trying to find software bugs

Verification vs Validation

✧ Verification:

"Are we building the product right".

The software should conform to its specification.

✧ Validation:

"Are we building the right product".

The software should do what the user really requires.

Software Testing

Software Testing

- Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use.
- When you test software, you execute a program using artificial data.
- Testing is part of a more general verification and validation process, which also includes static validation techniques

Why Software Testing ?

Software Testing is important as it may cause mission failure, impact on operational performance and reliability if not done properly.

Effective software testing delivers quality software products satisfying user's requirements, needs and expectations.

Who Should Test?

- Developer
  - Understands the system
  - But, will test gently
  - And, is driven by deadlines

- Independent tester
  - Must learn system
  - But, will attempt to break it
  - And, is driven by “quality”


ERROR,
Bug,
Defect,
Fault & Failure

Bug, Fault & Failure

Error: **An error is a human action that produces the incorrect result that results in a fault.**

Bug: An Error found in the development environment before the product is shipped to the customer.

Defect : is the difference between expected and actual result in the context of testing**.**

Fault: A wrong or mistaken step, process or Data definition in a computed program which causes the program to perform in an unintended or unanticipated manner**.**

Failure: Deviation of the software from its expected result. It is an event.

Test Case

- A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.
- The process of developing test cases can also help find problems in the requirements or design of an application.

Try to Write test Case

Stages of testing

- Development testing, where the system is tested during development to discover bugs and defects.
- Release testing, where a separate testing team test a complete version of the system before it is released to users.
- User testing, where users or potential users of a system test the system in their own environment.

Development testing

- Development testing includes all testing activities that are carried out by the team developing the system.
  - Unit testing, where individual program units or object classes are tested. Unit testing should focus on testing the functionality of objects or methods.
  - Component testing, where several individual units are integrated to create composite components. Component testing should focus on testing component interfaces.
  - System testing, where some or all of the components in a system are integrated and the system is tested as a whole. System testing should focus on testing component interactions.

Release testing

- Release testing is the process of testing a particular release of a system that is intended for use outside of the development team.
- The primary goal of the release testing process is to convince the supplier of the system that it is good enough for use.
  - Release testing, therefore, has to show that the system delivers its specified functionality, performance and dependability, and that it does not fail during normal use.
- Release testing is usually a black-box testing process where tests are only derived from the system specification.

Requirements based testing

- Requirements-based testing involves examining each requirement and developing a test or tests for it.
- MHC-PMS requirements:
  - If a patient is known to be allergic to any particular medication, then prescription of that medication shall result in a warning message being issued to the system user.
  - If a prescriber chooses to ignore an allergy warning, they shall provide a reason why this has been ignored.

Scenario testing

- Scenario testing is an approach to release testing where you devise typical scenarios of use and use these to develop test cases for the system.
- A scenario is a story that describes one way in which the system might be used.
- Scenarios should be realistic and real system users should be able to relate to them.
- If you have used scenarios as part of the requirements engineering process , then you may be able to reuse these as testing scenarios.

Performance testing

- Part of release testing may involve testing the emergent properties of a system, such as performance and reliability.
- Performance tests usually involve planning a series of tests where the load is steadily increased until the system performance becomes unacceptable.
- Stress testing is a form of performance testing where the system is deliberately overloaded to test its failure behaviour.

Acceptance testing

A formal test conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

It is the final test action before deploying the software. The goal of acceptance testing is to verify that the software is ready and can be used by the end user to perform the functions for which the software was built.

Alpha testing

1. The application is tested by the users who doesn't know about the application.
2. Done at developer's site under controlled conditions
3. Under the supervision of the developers.

Beta testing

1. This Testing is done before the final release of the software to end-users.
2. Before the final release of the software is released to users for testing where there will be no controlled conditions and the user here is free enough to do what ever he wants to do on the system to find errors.

Bug Reporting Format

Bug Id	:	Sys_def_xyz_o1
Test case Id	:	Sys_xyz_o1
Bug Description	:	Specify the defect.
Bug Priority	:	Major or Minor
Identified by	:	ABC
Module	:	Database Operations
Project	:	XYZ Corporation
Dated	:	22/9/03