

GANPAT UNIVERSITY  
U. V. PATEL COLLEGE OF ENGINEERING

# 2CEIT502 SOFTWARE ENGINEERING

## UNIT 10

### **ADVANCED TOPICS IN SOFTWARE ENGINEERING**

Prepared by: Prof. Y. J. Prajapati (Asst. Prof in IT Dept. , UVPCE)

# Outline

---

- ❑ Component-Based Software Engineering
- ❑ Client/Server Software Engineering
- ❑ Web Engineering, Reengineering
- ❑ Computer-Aided Software Engineering
- ❑ Software Process Improvement
- ❑ Emerging Trends in software Engineering

# Component-Based Software Engineering

- **Component Based Software Engineering (CBSE)** is a process that focuses on the design and development of computer-based systems with the use of reusable software components.

## **CBSE Framework Activities:**

### **Component Qualification:**

- This activity ensures that the system architecture define the requirements of the components for becoming a reusable component.
- Reusable components are generally identified through the traits in their interfaces.
- It means “the services that are given, and the means by which customers or consumers access these services” are defined as a part of the component interface.

### **Component Adaptation:**

This activity ensures that the architecture defines the design conditions for all component and identifying their modes of connection.

In some of the cases, existing reusable components may not be allowed to get used due to the architecture’s design rules and conditions.

These components should adapt and meet the requirements of the architecture or refused and replaced by other, more suitable components.



## **Component Composition:**

This activity ensures that the Architectural style of the system integrates the software components and form a working system. By identifying connection and coordination mechanisms of the system, the architecture describes the composition of the end product.

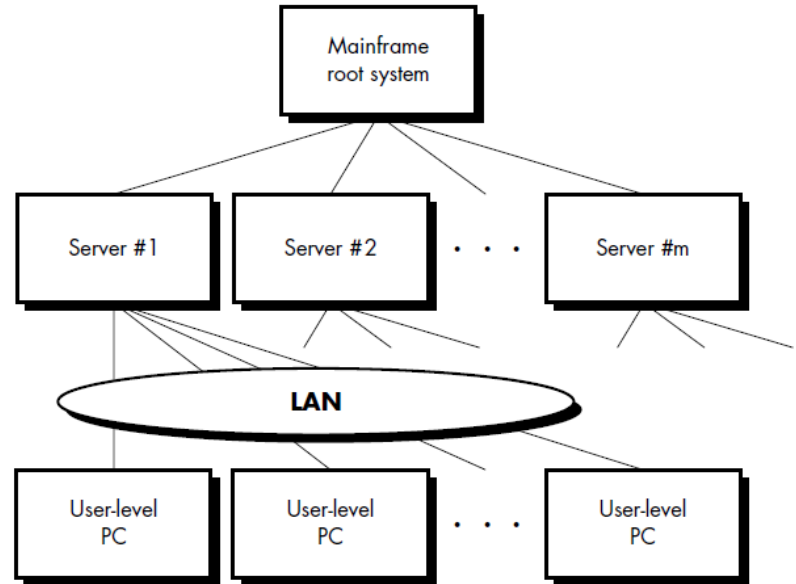
## **Component Update:**

This activity ensures the updation of reusable components. Sometimes, updates are complicated due to inclusion of third party (the organization that developed the reusable component may be outside the immediate control of the software engineering organization accessing the component currently.).

# Client/Server Software Engineering

- ❑ **Client-Server Architecture** is a distributed system architecture where the workload of client server are separated.
- ❑ **Clients** are those who request for the services or resources and **Server** means the resource provider.
- ❑ The server host several programs at its end for sharing resources to its clients whenever requested.
- ❑ Client and server can be on the same system or may be in a network.

## THE STRUCTURE OF CLIENT/SERVER SYSTEMS



### File servers:

- The client requests specific records from a file. The server transmits these records to the client across the network.

### Database servers:

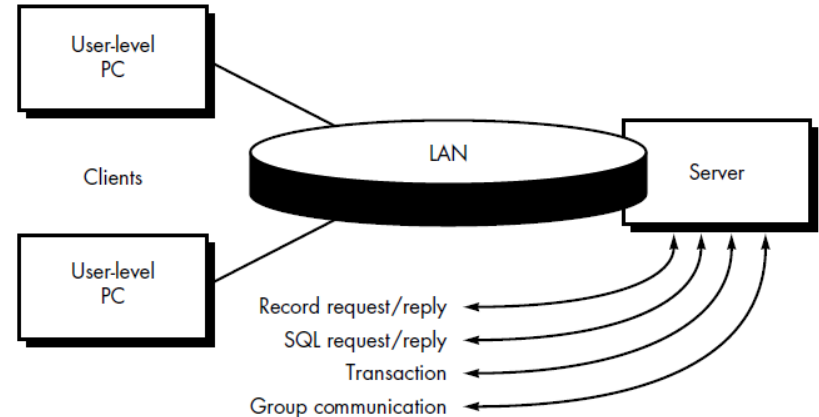
- The client sends structured query language (SQL) requests to the server. These are transmitted as messages across the network.
- The server processes the SQL request and finds the requested information, passing back the results only to the client.

### Transaction servers:

- The client sends a request that invokes remote procedures at the server site. The remote procedures are a set of SQL statements.
- A transaction occurs when a request results in the execution of the remote procedure with the result transmitted back to the client.

### Groupware servers:

- When the server provides a set of applications that enable communication among clients (and the people using them) using text, images, bulletin boards, video, and other representations, groupware architecture exists.



## Software Components for c/s Systems

### User interaction/presentation subsystem:

- This subsystem implements all functions that are typically associated with a graphical user interface

### Application subsystem

- This subsystem implements the requirements defined by the application within the context of the domain in which the application operates.
- For example, a business application might produce a variety of printed reports based on numeric input, calculations, database information, and other considerations.

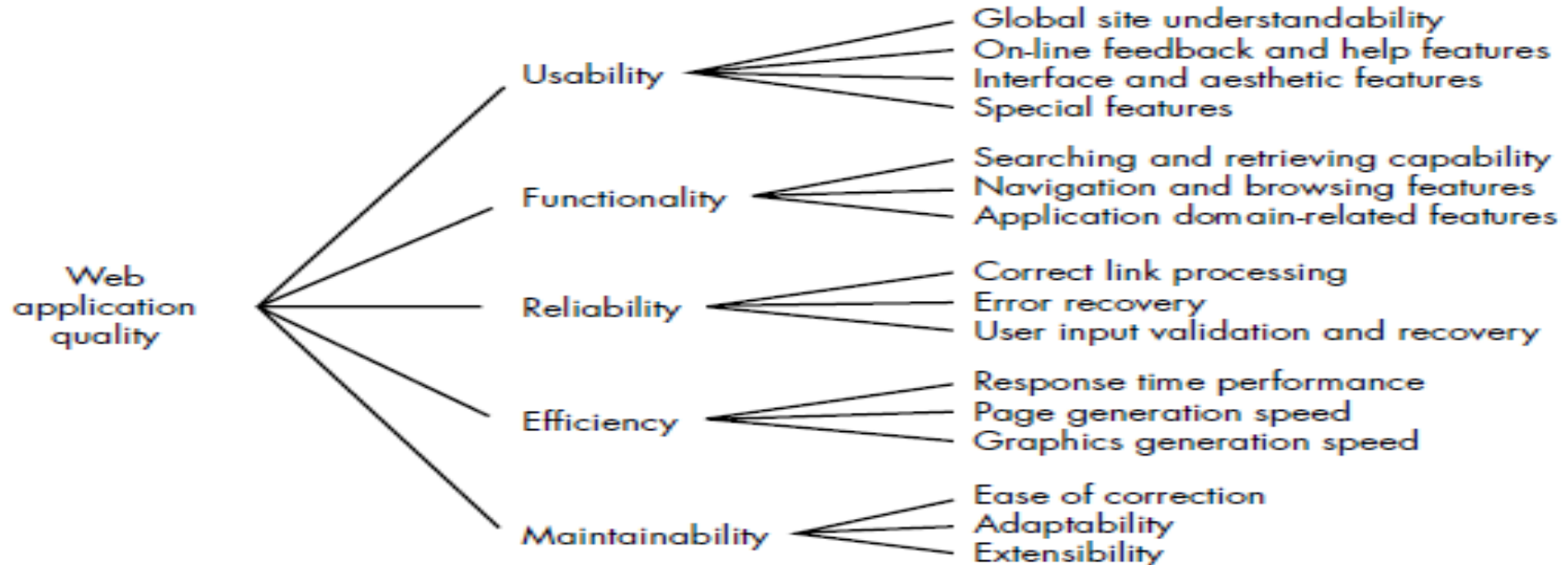
### Database management subsystem

- This subsystem performs the data manipulation and management required by an application. Data manipulation and management may be as simple as the transfer of a record or as complex as the processing of sophisticated SQL transactions

# Web Engineering, Reengineering

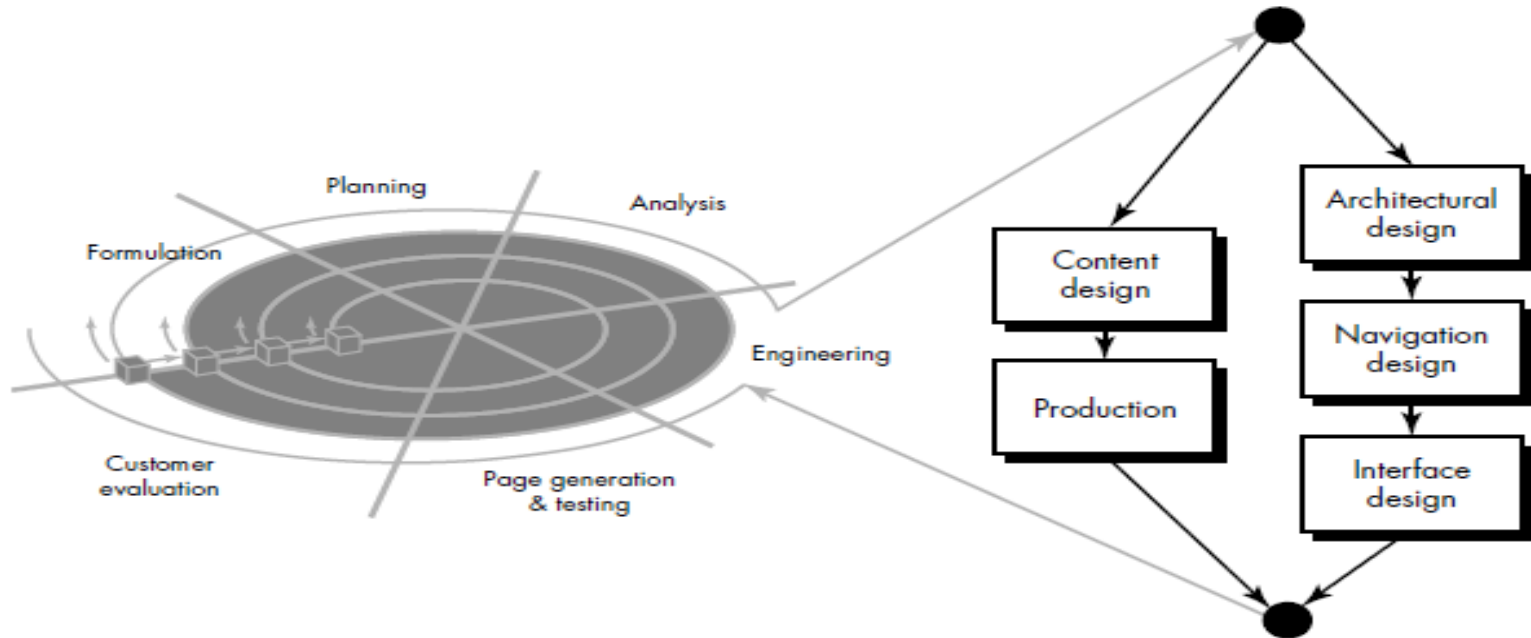
## THE ATTRIBUTES OF WEB-BASED APPLICATIONS

Quality Attributes:





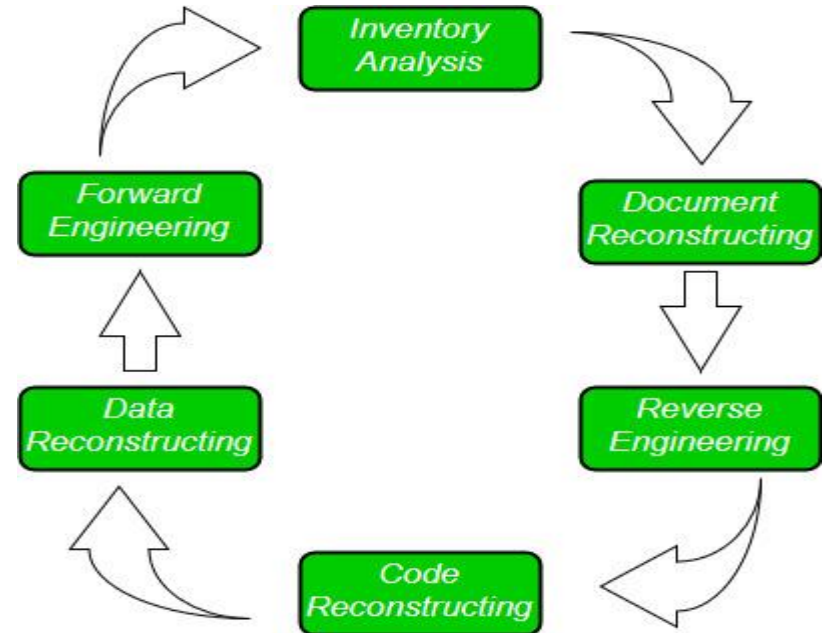
# A FRAMEWORK FOR WEBE



# Software Re-Engineering

- ❑ **Software Re-Engineering** is the examination and alteration of a system to reconstitute it in a new form.
- ❑ The principles of Re-Engineering when applied to the software development process is called software re-engineering.
- ❑ It affects positively at software cost, quality, service to the customer and speed of delivery.
- ❑ In Software Re-engineering, we are improving the software to make it more efficient and effective.

## Software Re-Engineering Activities:



### **Inventory Analysis:**

- Inventory can be nothing more than a spreadsheet model containing information that provides a detailed description of every active application.
- Resource can then be allocated to candidate application for re-engineering work.

### **Document reconstructing:**

- Documentation must be updated.
- It may not be necessary to fully document an application.
- The system is business critical and must be fully re-documented.

### **Reverse Engineering:**

- Reverse engineering is a process of design recovery. Reverse engineering tools extracts data, architectural and procedural design information from an existing program.

### **Code Reconstructing:**

- To accomplish code reconstructing, the source code is analysed using a reconstructing tool. Violations of structured programming construct are noted and code is then reconstruct.
- The resultant restructured code is reviewed and tested to ensure that no anomalies have been introduced.

### **Data Restructuring:**

- Data restructuring begins with the reverse engineering activity.
- Current data architecture is dissected, and necessary data models are defined.
- Data objects and attributes are identified, and existing data structure are reviewed for quality.

### **Forward Engineering:**

- Forward Engineering also called as renovation or reclamation not only for recovers design information from existing software but uses this information to alter or reconstitute the existing system in an effort to improve its overall quality.

# Computer-Aided Software Engineering(CASE)

CASE stands for **Computer Aided Software Engineering**. It means, development and maintenance of software projects with help of various automated software tools.

## Components of CASE Tools

### Central Repository:

- CASE tools require a central repository, which can serve as a source of common, integrated and consistent information.
- Central repository is a central place of storage where product specifications, requirement documents, related reports and diagrams, other useful information regarding management is stored. Central repository also serves as data dictionary.

### Diagram tools

- These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form.

### Process Modeling Tools

- Process modelling is method to create software process model, which is used to develop the software. Process modelling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

### Project Management Tools

- These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Managers have to strictly comply project execution with every mentioned step in software project management.
- For example, Creative Pro Office, Trac Project, Basecamp.

## Documentation Tools

- Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.
- Documentation tools generate documents for technical users and end users.
- For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

## Analysis Tools

- These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions.
- For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

## Design Tools

- These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques.
- These tools provides detailing of each module and interconnections among modules.
- For example, Animated Software Design

## Configuration Management Tools

Configuration Management tools deal with –

- Version and revision management
- Baseline configuration management
- Change control management

For example, Fossil, Git, Accu REV.

## **Programming Tools**

- These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools.
- These tools provide comprehensive aid in building software product and include features for simulation and testing.

## **Prototyping Tools**

- Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product.
- For example, Serena prototype composer, Mockup Builder.

## **Web Development Tools**

- These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on.
- Web tools also provide live preview of what is being developed and how will it look after completion.

## **Quality Assurance Tools**

- Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards.
- For example, SoapTest, AppsWatch, JMeter.

## **Maintenance Tools**

- Software maintenance includes modifications in the software product after it is delivered.
- For example, Bugzilla for defect tracking, HP Quality Center.

# Software Process Improvement

- ❑ Software Process Improvement (SPI) methodology is defined as a sequence of tasks, tools, and techniques to plan and implement improvement activities to achieve specific goals such as increasing development speed, achieving higher product quality or reducing costs.





## **Current Situation Evaluation**

This step is the initial phase of the process and it is mainly to assess the current situation of the software process by eliciting the requirements from the stakeholders, analyzing the current artifacts and deliverables, and identifying the inefficiencies from the software process.

## **Improvement Planning**

After analyzing the current situation and the improvement goals, the findings should be categorized and prioritized according to which one is the most important or have the most severity.

## **Improvement Implementation**

In this step, the planned activities are executed and it puts the improvements into practice and spreads it across the organization.

## **Improvement Evaluation**

The before improvement measures, after the improvement measures, and the target improvement measure. Measurement, in general, permits an organization to compare the rate of actual change against its planned change and allocate resources based on the gaps between actual and expected progress.



# Emerging Trends in software Engineering

- ❑ AI based application
- ❑ Robotic process Automation
- ❑ Progressive Web Apps
- ❑ Rapid prototyping and Innovation
- ❑ Digital Transformation Enablers
- ❑ Low-Code Development
- ❑ Cloud computing
- ❑ Python will dominate the demand

