23

Web Engineering

CHAPTER COVERAGE

1. Introduction
2. Introduction to Web
3. Categories
4. Web Applications Categories
5. How Web Application Work?
6. Advantages of Web Applications
7. Drawbacks of Web Applications
8. Web Engineering

23.1 INTRODUCTION

Web (World Wide Web) is not only the way to browse today, but many large, sophisticated, and mission critical applications are built on web platform now-a-days. As you can understand, few characteristics of the web applications are very unique over the traditional applications such as usability, performance, security, and scalability. But yet many web applications are developed in an ad hoc fashion still now. The web engineering section of the software engineering deals with the systematic, disciplined, and quantifiable approach toward the web development.

23.2 INTRODUCTION TO WEB

Literal meaning of web is "a network of Thread constructed by Spider." In software engineering, the term web (World Wide Web) is a collection of computers connected by a network (Figure 23.1). Web is an indispensable and an evolving and transformative technology. It helps virtually in every aspect of modern living. It changes the way of doing things and the way to acquire and disseminate information.

Web helps us to place any information and documents in web in electronic formats and retrieve the same from web at later point of time from any part of the world. Known electronic formats in which we can store information are text file, graphics file, sound files, and movie files. Web also helps to access other people's information from web, thereby enhancing our knowledge. Web has search facility to search and retrieve only the required information from web. Web not only helps to store and retrieve the

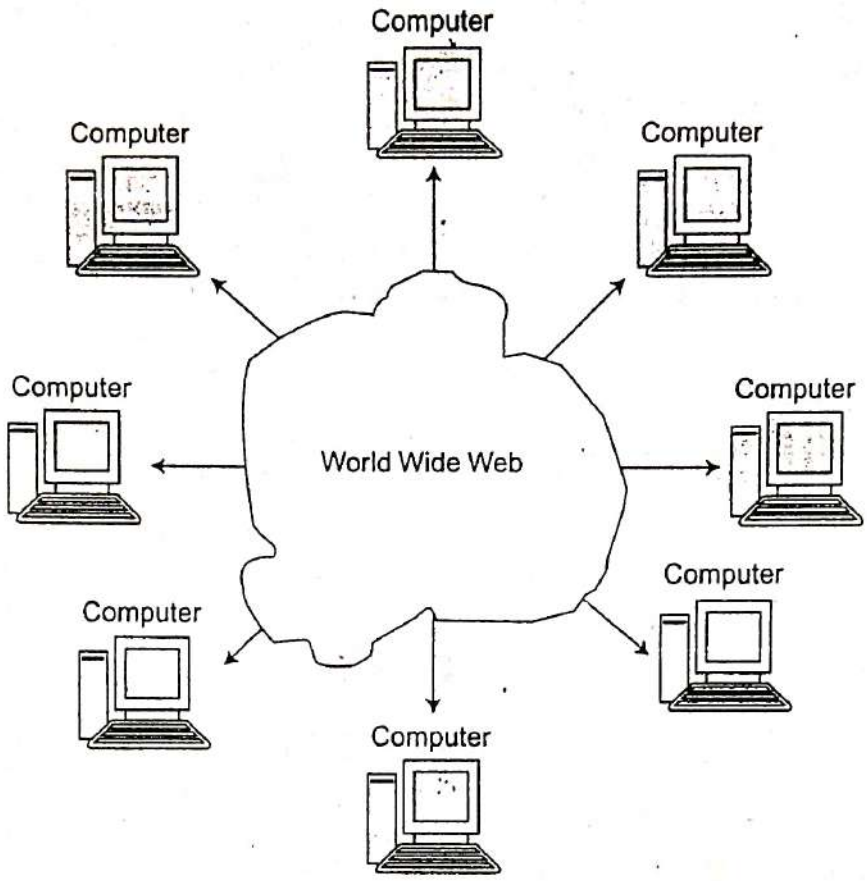


Figure 23.1 World Wide Web

information but also helps to place the whole application (web-based application) in the web which caters the users accessing the application through web. Web also has authorization concepts wherein only authorized user can access those information.

23.3 GENERAL WEB CHARACTERISTICS

Client server is the basic concept under which World Wide Web system acts. Client machine or client is the computer that uses the applications of the web and retrieves and uses data (Figure 23.2). Client machine has a software called as web browser(s) for this purpose. A web browser software which runs on a client machine helps to access the World Wide Web.

Popular web browsers are:

1. Microsoft Internet explorer
2. Google chrome web browser

Server machine or server is the computer in which the actual application and information are stored. Server also has web application that communicates with the client browsers. Information can also be stored in the server using database (Figure 23.3).

Note: A single client can access the information resided in multiple servers through web applications. A client can also act as a server for some other application and vice versa. (A Server can also act as a client for some other application).

Examples of web applications include: (web application can be accessed using www)

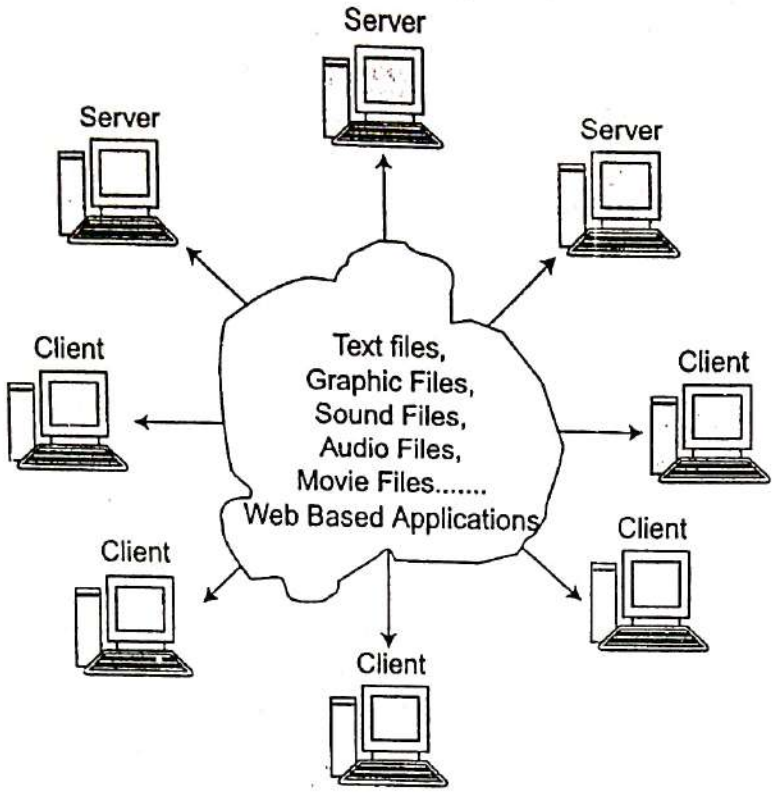


Figure 23.2 World Wide Web Characteristics

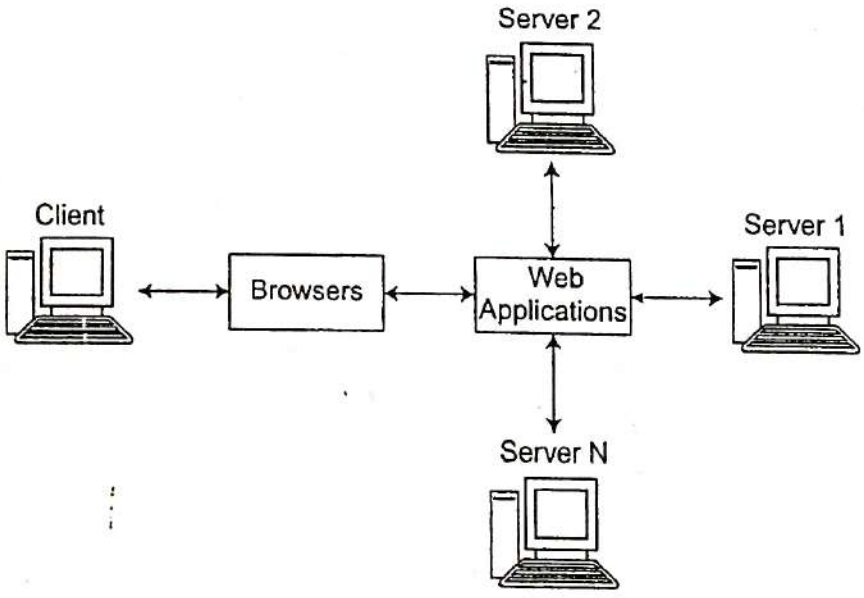


Figure 23.3 Web Browsers and Web Applications

1. www.thehindu.com, www.indiatimes.com (for latest news)
2. www.amazon.com, www.flipkart.com (for electronic shopping)
3. www.india.gov.in (Indian government portal)
4. www.Naukri.com (job portal)
5. www.Irctc.co.in (for travel—Indian railway)

23.4 WEB APPLICATIONS CATEGORIES

Web applications are categorized into three based on where it runs:

1. Client-side web applications
2. Server-side web applications
3. Middleware web applications

Client-side web applications:

This web application runs at the client side in browser. Resources (pages) are loaded at the client side. Client side applications are coded using technologies such as HTML, CSS, and JAVA Scripts.

Server-side web applications:

This web application runs at the server only. Server-side applications are coded using technologies such as JAVA, ASP, and PHP.

Middleware web applications:

This web application acts in between the client and server. Technologies include enterprise Java Beans, Java Servlet, etc.

Web applications can also be categorized based on its usage:

1. Online news provider (www.indiatimes.com)
2. Online web e-mail provider (www.webmail.com)
3. Document-sharing web sites (downloadable documents) (www.4shared.com)
4. Online chat sites (www.yahoo.com)
5. Blog-based sites (www.wordpress.com)
6. Social networking sites (www.Facebook.com, www.twitter.com, etc.)
7. Online Search Engines—to find and download documents
8. Portals (for specific purpose)

23.5 HOW WEB APPLICATION WORK?

Web application has three layers namely presentation layer, application layer (logical layer) and data layer (Storage layer). Presentation layer consists of a basic browser which runs HTML kind of pages. Application layer does basic calculation and business logic, and it is also called as logical layer. Data layer consists of database and other storage devices. Database software is used to access the database (Figure 23.4).

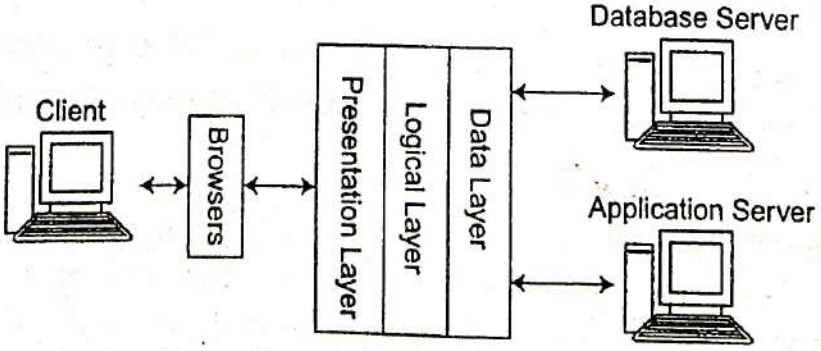


Figure 23.4 Layers of Web Application

23.6 ADVANTAGES OF WEB APPLICATIONS

1. Application can be accessed from anywhere in the world from any machine.
2. Only browser is needed at the client side to connect with the web application. No other special software is required at the client side.
3. Operation system (Windows, Linux, and Mac) at the client side is not a bottleneck to access the application. Any operating system will run the application.
4. Database and objects are running at the server end, and so load at the client side is always low.
5. Upgrading the web application is easy as it required the application to be updated only at a single place (server side). In a stand-alone system, the application will be running at different machines, and upgrading is a tedious task.

23.7 DRAWBACKS OF WEB APPLICATIONS

- Security is the main drawback of any web application. Without proper security, the web applications are prone to get hacked easily by hackers: We can practice ethical hacking, thereby finding all the possibilities of hacking the developed web application and addressing it through counter mechanism. We need to ensure all the security engineering aspects are taken care of even from the beginning stage of the web development, thereby ensuring the development of a secured web application.
- Data theft is possible easily in web application: We can ensure security engineering aspects are taken care of even from the beginning stage of the development, thereby avoiding data theft. We can also ensure proper backup mechanism of the data such as cold backup and hot backup systems to reduce the impact of the theft. We can also use data-masking mechanism to avoid data theft.
- Application also can be copied (duplicated) easily in the web: We need to follow the engineering principles and best practices to avoid this. Security engineering is a specialized field of engineering that focuses on the security aspects of a system. It ensures the security of the system is taken care in the entire life cycle of the system including requirements and design of the system.

23.8 WEB ENGINEERING

> **Web Engineering** - *is the application of systematic, disciplined, and quantifiable approches to the design, production, deploymemt, operation, maintenance and evolution of Web - based software products.*
>
> — Gaedke, 2000

Web engineering is also defined as the process of creating high-quality web-based applications.

Web engineering in addition to software engineering also consists of the following engineering aspects related to the development of web-based application (Figure 23.5).

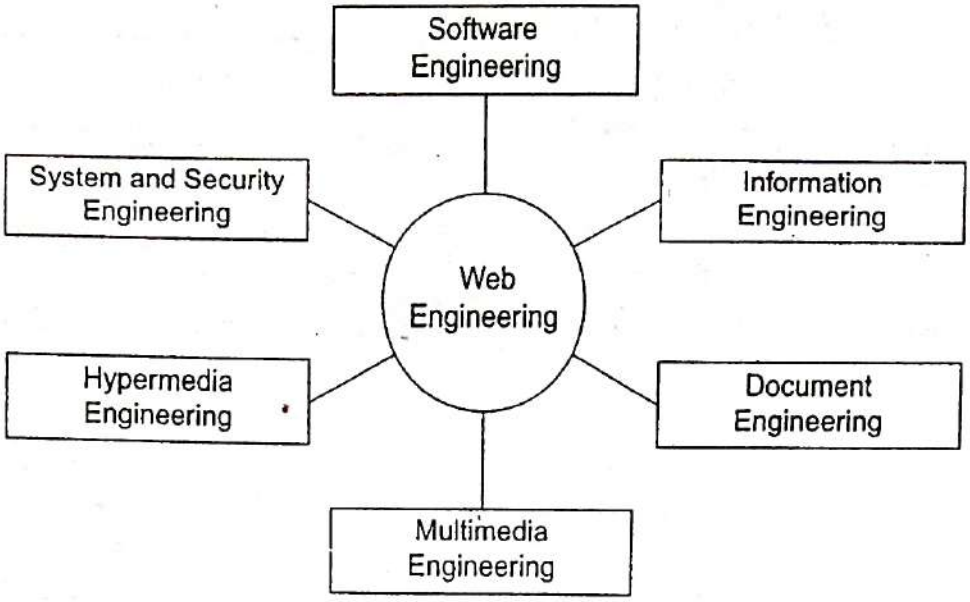


Figure 23.5 Web Engineering

- Information engineering: Information engineering can be defined as an approach of designing and developing information systems. It can also be defined as the generation, distribution, interpretation and use of information in systems. It also takes care of the availability of the information, accessibility of the information, security of the information, and authenticity of the information.
- Document engineering: Document engineering can be defined as an approach of designing and developing document-based system which helps to get ideas from various information and systems. This is the next level of information engineering. In addition to the attributes of information engineering, the documentation engineering takes care of the look and feel of the documents, performance (time to create and retrieve the documents), and concurrency of the documents.
- Multimedia engineering: Multimedia engineering can be defined as an approach of designing and developing a system based on multimedia concepts such as animations, voice, and audio. It can be defined as the generation, distribution, and interpretation and use of multimedia concepts in a system.
- Hypermedia engineering: Hypermedia is a term which describes the form and structure of online information connecting electronic data. It is mainly used for interactive information. It analyzes the purpose for which a web site or CDROM is created and thereby defines a process of creating it.
- System engineering: System engineering is an interdisciplinary field of engineering that focuses on integrating all the aspects of a system such as project management, control engineering, and industrial engineering including human resource interaction of the system.
- Security engineering: Security engineering is a specialized field of engineering that focuses on the security aspects of a system. It ensures the security of the system is taken care in the entire life cycle of the system including requirements and design of the system. Security engineering also takes care of passing the message in a secured way between systems by various mechanisms such as cryptography and ethical hacking.

23.8.1 Goals of Web Engineering

Following are the goals of web engineering:

1. Develop high-quality web application as per the requirement
2. Develop effective and efficient web application
3. Continuously adapt to the requirement and environment change
4. Use web engineering process model
5. Continuously follow security and administrative regulations

23.8.2 Web Engineering versus Software Engineering

Table 23.1 differentiates software engineering and web engineering.

Table 23.1 Software Engineering versus Web Engineering

	Software Engineering	Web Engineering
User base	Small	Large
User Requirements	Specific	Changes fast
Cost of Development	Varies	Usually Low
Infrastructuew Cose	High	Comparatively Low
Release Cycle	Longer	Shorter
Security	Not so Important	Highly important
Architecture Complexity	Low	High

23.8.3 Attributes of Web Engineering

Attributes of web engineering can be easily derived from the web engineering and its associated aspects (related engineering fields). Web engineering attributes are the combination of associated aspects.

Attributes of information engineering:

1. Security of information
2. Availability of information
3. Accessibility of information
4. Authenticity of information

Attributes of document engineering:

In addition to the attributes of information engineering, the documentation engineering has the following attributes:

1. Type of documents
2. Concurrency
3. Performance (time to create)
4. Look and feel of the document

Attributes of multimedia engineering:

In addition to the attributes of document engineering, the multimedia engineering has the following attributes:

1. Loading and refreshment time
2. Sensitive contents (age restrictions)

Attributes of hypermedia engineering:

In addition to the attributes of multimedia engineering, the hypermedia engineering has the following attributes:

1. Compatibility (browser compatibility)
2. CPU utilization at customer side
3. CPU utilization at server side

Attributes of system and security engineering:

1. Server size
2. System network type
3. System (server and client) distribution
4. Server load
5. Authority level
6. Authenticity level

23.8.4 Principles of Web Design

Usability: Usability is the degree of user interface design consideration, which takes care of making the system effective and efficient and satisfying the users by taking into account the human psychology and physiology of the users. Any user interface has four elements, namely users, tasks, contents, and environment. All these factors need to be analyzed. Refer three golden rules of Mandel which we discussed in Chapter 7. In addition to Mandel's rule, the following principles can also be followed:

1. Target for simple model always.
2. Similar components (elements) can be placed at same place (together) in a page.
3. Proper and uniform distribution of heavy and light elements within a page is necessary.
4. More emphasis can be given to important components (elements) in a page.
5. Consistencies need to be maintained between pages (colors, menus, etc.).

23.8.5 Web Engineering Process

We already discussed the difference between software engineering and web engineering, and it indicates separate process required for web engineering catering its unique need. The web engineering process must also be agile (incremental and iterative) as user's requirements evolve over time (user base is usually huge for web-based applications) and changes occur frequently.

When it comes to web engineering, each phase needs to be iterative and incremental in nature. (Figure 23.6).

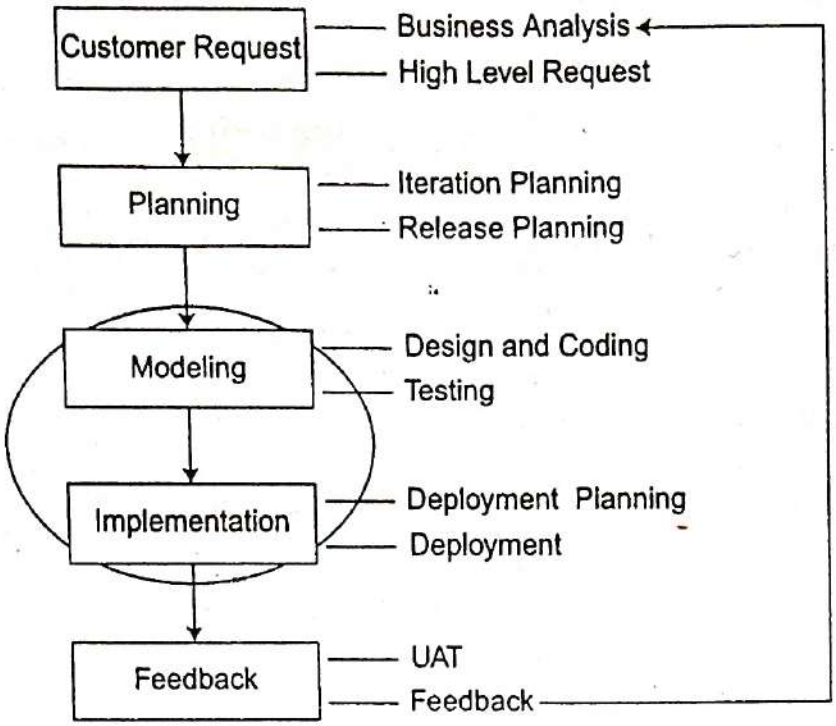


Figure 23.6 Web Engineering Process

Customer request phase

Customer request phase is the first phase of web software development in which the customer and the business analysts (usually developer) interact with each other and document the necessities of the outcome of the web application.

Business need: Business need of the web site (web application) is being discussed here. Business outcome is defined in the form of business statements or vision statements.

Requirement collection: Customers would state their requirements (business statements, vision statements)—how and what the web application should perform. The requirements may also include the appearances (interface requirements), the functionalities (functional requirements) and the web application output expectations (non-functional requirements).

The steps in this phase include:

- Feasibility study if required
- Eliciting requirements at high level
- Requirement analysis
- Specify and validate requirements

Some of the factors to be considered in this phase include:

1. Degree to which the organization's success is directly dependent on the success of the web application
2. Relationship between business need and requirements
3. Technical feasibility of the web application to be developed
4. Number of users of the web application. This is a critical parameter while designing the web page.

5. Number of stakeholders (analysis can help to identify conflicting requirements coming from different sources)
6. Size of the web engineering team
7. Degree to which members of the WebE team have worked together before (analysis can help develop a common understanding of the project)

Output of this phase: Business case document, requirement specification document, and feasibility study output

Planning phase

Planning phase is the second phase of web development in which the feasibility study outcome and requirement specifications are analyzed and a decision is taken to move forward or not.

Iteration planning and release planning: Planning phase sets the stage for both iteration planning and release planning. The entire set of requirements are divided into release buckets (release backlog) based on the release cycle. A release will have its own release backlog. Now release backlog are further subdivided into iteration backlog based on the iteration cycle. A release will have multiple iterations. Iterations are time-boxed and ranges from 1 to 4 weeks and follow a strict schedule.

Team will start coding the web application by going through iteration by iteration. Iterations are the heartbeat of an agile project. It is inside the iteration that the actual work is achieved. Iteration plan consists of iteration backlog (requirements of the iteration), assumptions, risks, actions, dependencies, and communication. The team will determine what features can be accommodated in the current iterative cycle (can also be called as sprint). It also orders (priorities) the features of the current immediate iterative cycle. Iterations follow a consistent, unchanging schedule in which each activity is time-boxed:

- Demonstrate previous iteration (up to an hour)
- Hold retrospective on previous iteration (1 h)
- Plan iteration (half an hour to 4 h)
- Commit to delivering stories (15 min)
- Develop stories (remainder of iteration)
- Prepare release (1 h)

This helps the team keep focused on the main activities and avoid spending time on the low-priority tasks. Modeling and implementations are execution part of the iteration planning.

Output of this phase: Iteration backlog (iteration requirements) and release backlog (release requirements).

Modeling and implementation phase

"Modeling and implementations" phases together form the execution part of the iteration planning. In modeling phase designing, coding and testing of the iteration-related tasks happens. This is called as modeling phase as the entire application is getting refined and changed multiple times until it is getting implemented. During the feedback stage, the customer accepts part of the web application (iteration outcome) after demo.

We keep the basic design framework and keeps on adding and refining the design framework when we move iteration to iteration. Modeling includes interface design and architectural design.

Design of the web application depends on the complexity level of the web site, the graphical interfaces required, and simplicity needed in usage of the end users.

Design of the web application should be:

1. Precise and specific
2. Conform to budget and requirements
3. A direct solution to the problem

Coding and testing: The coding is defined as the phase where the document format is transformed into lines of codes in web programming language (HTML, ASP, JAVA, etc.). Writing standardized coding for software not only helps to maintain it properly once it is built but also helps to enhance the code easily at later point of time; it also helps to identify and fix the bugs quickly and easily, and so writing the proper working code alone is not enough and focus should be on writing a standard code. After the coding phase is completed, the lines of code have to be tested for ensuring its right functionality in the platform. Testing is carried out by a separate team of experts (software testers) specialized in generating sample cases and subjecting the product to various tests. Only after the product has been proved to be reliable over the tests, it enters into the implementation phase.

Following aspects are tested typically for a web application:

- Testing web application at different operating systems
- Testing web application at different browsers (or other interface devices such as set-top boxes, personal digital assistants (PDAs), and mobile phones)
- Testing web application at different hardware platforms
- Communications protocols checking
- "Backroom" applications checking

In implementation phase, the deployment planning and deployment happens which will make the web application ready for demo, feedback, and customer acceptance.

Feedback phase

In feedback phase, the user acceptance testing happens for the particular iteration. Any further refinement and changes are carried out in the next iteration which make the web site (web application) getting evolved over a period of time. Feedback phase is vital for the web site development as it is used to implement the changes at the next upcoming iterations. Normally, the iteration demo marks the end of the iteration followed by the team retrospective. Key elements of the retrospective meeting are:

- Process improvements are made at the end of every iteration—ensures that the project team is always in self-improving mode.
- The retrospective is a collaborative process between all members of the team.
- All team members identify what went well and what can be improved.
- The actions and lessons learnt are prioritized based on the team direction
- Team works out a solution to most prominent problems—helps to build the team ownership and self management.
- Helps the team formation and bonding as the areas of conflict can be identified and dealt with.

23.8.6 Web Engineering Best Practices

- Understand the business need of the web application
- Always follow iterative and incremental steps for web engineering

- Use as many re-usable components as possible
- Give more importance to security (encryption, firewalls, etc.)
- Always concentrate on consistency and quality
- Always follow Internet standards
- Use good web programming tools
- Prepare a project schedule and continuously update it

23.8.7 Web Metrics

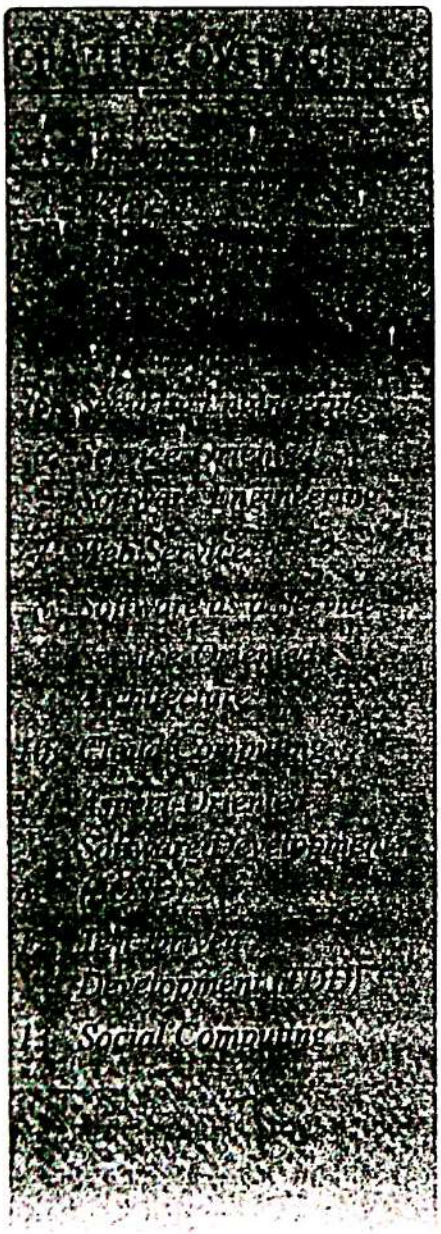
1. *Web traffic*: How many users are using the web site on average per day?
2. *Popularity of web application (web pages)*: This is the comparative rating of web traffic in comparison to similar kind of web applications.
3. User accessibility of the web application.
4. *Loading time of the web application*: How much time it takes the application to display on the client browser?
5. *Business realization percentage*: Any web application starts with a business case and how much of the business objective got realized because of the web application.

Summary

- Web-based information system (WIS) development process is different and unique.
- Web engineering is multi-disciplinary; no single discipline (such as software engineering) can provide complete theory basis, body of knowledge and practices to guide WIS development.
- Issues of evolution and lifecycle management when compared to more "traditional" applications.
- WIS and applications are pervasive and non-trivial. The prospect of web as a platform will continue to grow and it is worth being treated specifically.
- Literal meaning of web is "a network of thread constructed by spider". In software engineering, the term web is a collection of computers connected by a network.
- Client server is the basic concept under which World Wide Web system acts. Client machine or client is the computer that uses the applications of the web and retrieves and uses the data. Client machine has a software called as web browser(s) for this purpose.
- Server machine or server is the computer in which the actual application and information are stored. Server also has a web application that communicates with the client browsers. Information can also be stored in the server using database.
- Web applications are categorized into three based on where it runs:
  1. Client-side web applications
  2. Server-side web applications
  3. Middleware web applications
- Web applications can also be categorized based on its usage.

24

Emerging Trends in Software Engineering



24.1 INTRODUCTION

Software engineering field is evolving constantly as well as rapidly; so adapting to new changes is imperative for success of this field. Unlike other fields, software engineering undergoes several changes with the invention of new technology and innovation awareness. Various factors determine the trend in software engineering field, such as

1. Technology
2. Business environment
3. Economy situation
4. Innovation
5. Increase in business complexity
6. Competitive world

In this chapter, let us look at some of the emerging trends in software engineering.

24.2 WEB 2.0

Web 1.0 is considered as web pages created with static pages (HTML) and is completely content-driven; and user of the website has no role to play other than just reading the website and it is just one-way communication. Web 2.0 is (Figure 24.1) considered as a collaborative technology, which is more than the static web pages in which users take part collaboratively and contribute to the content; websites are created using dynamic web pages. Users can also edit and update the already existing website content.

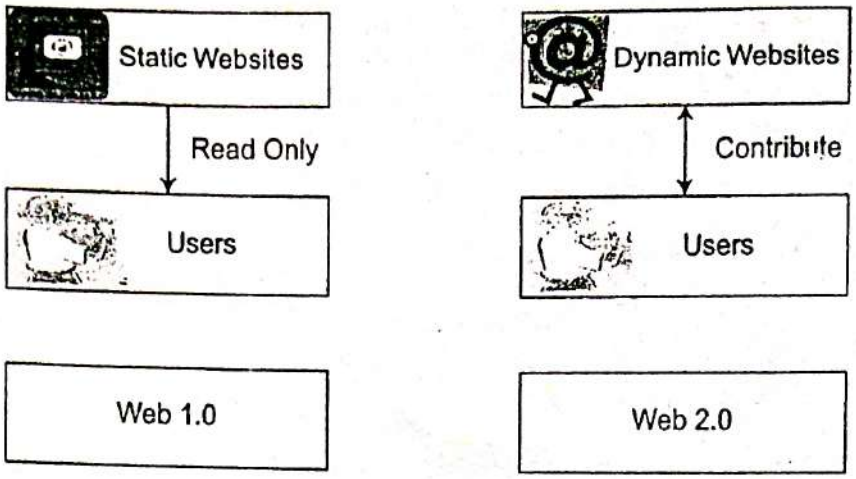


Figure 24.1 Representation of Web 2.0

Web 2.0 uses social media for interactions. Blogging is the best example for Web 2.0 principle; Gmail, Google maps, wikies, and Flickr are a few examples of Web 2.0.

Web 2.0 websites include following features and techniques, which are referred as SLATES (Figure 24.2) by Andrew McAfee (Source: http://en.wikipedia.org/wiki/Web_2.0).

Search: Obtaining information through keyword search.

Links: Connects information together into a meaningful information ecosystem using the model of the Web, and provides low-barrier social tools.

Authoring: The ability to create and update the content leads to the collaborative work of many rather than just a few web authors. In wikis, users may extend, undo, and redo each other's work. In blogs, posts and comments of individuals build up over time.

Tags: Tags categorization of content by users adding "tags"—short, usually one-word descriptions—to facilitate searching, without depending on pre-made categories. Collection of tags created by many users within a single system is referred as "folksonomies" (i.e., folk taxonomies).

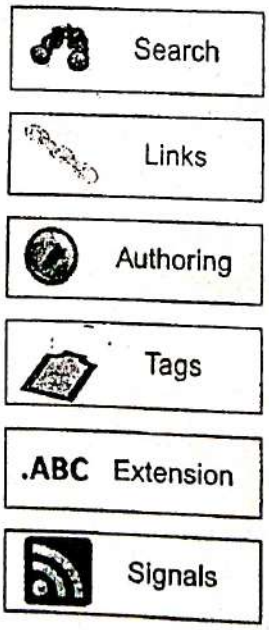


Figure 24.2 SLATES–Web 2.0

Extensions: Software that makes the Web an application platform as well as a document server, which include softwares such as Adobe Reader, Adobe Flash player, Microsoft Silverlight, ActiveX, Oracle Java, QuickTime, and Windows Media.

Signals: The use of syndication technology such as RSS- Rich Site Summary to notify users of content changes.

These days, web 3.0 is talked about; the machines (computers) generate the data and their associated new information on their own based on the need. In Web 3.0, computers start interacting with humans. Now the trend is moving from Web 2.0 to Web 3.0

24.3 RAPID DELIVERY

Software engineering principles (life cycles) are being followed in parallel rather than in sequence, as the customer wants to see the outcome quickly. Agile software development is the best example for rapid delivery. As we have already discussed in the previous chapter, agile is iterative and incremental (Figure 24.3) way of developing software.

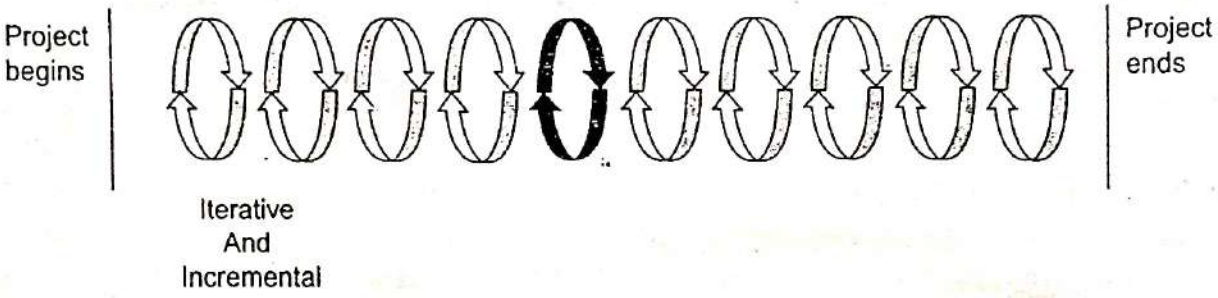


Figure 24.3 Iterative and Incremental Delivery

A time box (usually 2–4 weeks of time frame) is fixed and whatever possible during this boxing period will be delivered as a workable incremental software product to the customer. Changes in the interim product as well as addition will be carried out in the subsequent iteration and this repeats until all the requirements are fulfilled. This takes the software to the customer quickly and thereby customer gets early value from the product.

Examples for Rapid Delivery include:

1. SCRUM Methodology
2. XP Methodology
3. Dynamic Software Development Methodology (DSDM)
4. KANBAN Methodology

24.4 OPEN SOURCE SOFTWARE DEVELOPMENT

As the name suggests, open source not only gives the executable code to the users but also the source code to the users (Free license), which has a lot of advantages (Figure 24.4). The users can customize the code further based on his needs. He can also contribute his changed code to others (Open). This way of development of code is cost-effective and also ensures that software is being used as an

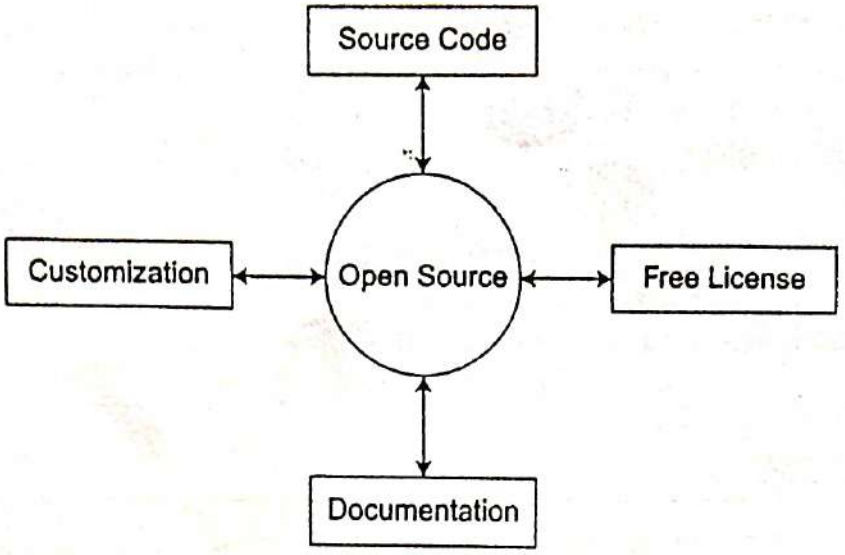


Figure 24.4 Open Source Software Development

enabler for life and business rather than controlling the life of the people in the name of warranty, support, and maintenance. Documentation of the entire source code is also available free of cost.

A few examples of Open Source software products are as follows:

1. Eclipse—Integrated Development Environment
2. Mozilla Firefox—Famous Web browser
3. Word press—Web publishing platform (website creation)
4. PHP—Scripting language of Web
5. Linux—famous Operating system (like UNIX)

24.5 SECURITY ENGINEERING

Security engineering is a specialized field of engineering that focuses on the security aspects of the software system. Security is given high priority during the requirement analysis and designing stage of the system itself so that we ensure nobody else hacks (intrude) into the system and modifies the code maliciously. It also aims to prevent misuse of data in the system. Owing to technological advances, the software system becomes complex and we need to give a lot of attention to security aspects of the system. Common coding errors lead to security vulnerability and developers need to follow security-related best practices while writing the code. Default deny and Default permit are two fundamental principles (Figure 24.5) of security engineering aspects.

Default deny indicates that the call to the source code is denied except special permission calls. In other way, special permitted code can only be called from outside and all others are denied. Default permit is dangerous, which is opposite to Default deny. In default permit mode, everything is default permitted, which is a big security threat to the code.

While gathering the functionality requirements, we also need to gather the security requirement of the system. For example, security needs to be given a lot of attention while developing core banking applications dealing with sensitive data. Data hiding and information hiding principles need to be worked on. Common software programming errors lead to security lapses.

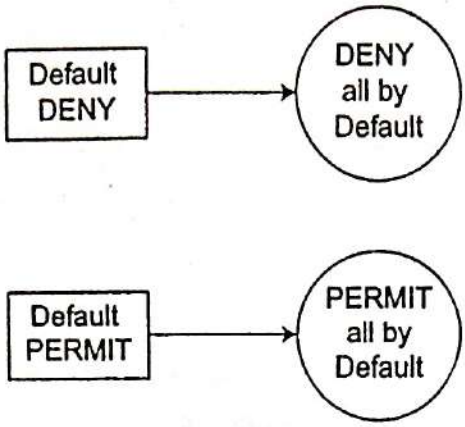


Figure 24.5 Default Deny and Default Permit

24.6 SERVICE-ORIENTED SOFTWARE ENGINEERING

Service-oriented software engineering (SOSE) is the development of software system using reusable components and services from other providers (Figure 24.6). Parts of these systems are called as components, but it differs from component-based software engineering, as the service outcome of those components are focused on and it has the ability to locate the services at run time.

SOSE acts as a middle layer between the business and system developers (IT). Business users will only look into the services rather than the implementation of those services in the system, and IT provider will take care of it. Iterative development is mostly suitable in this case.

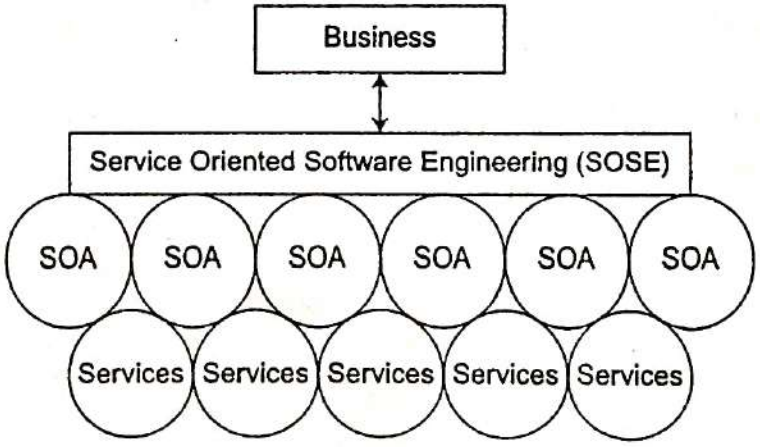


Figure 24.6 Service-oriented Software Engineering (SOSE)

24.7 WEB SERVICE

Web Service concepts are famous because of SOSE (Figure 24.7). Services usually do not have interfaces and are independent of each other. Services depend solely on the input messages that are being passed on to it, which are usually in the XML format. Web Service Description Language (WSDL), which is also XML based, is used to describe the services provided, Format of the service, accessibility of the service. It provides Introduction, Interface, Implementation related to the service, and helps locate the services.

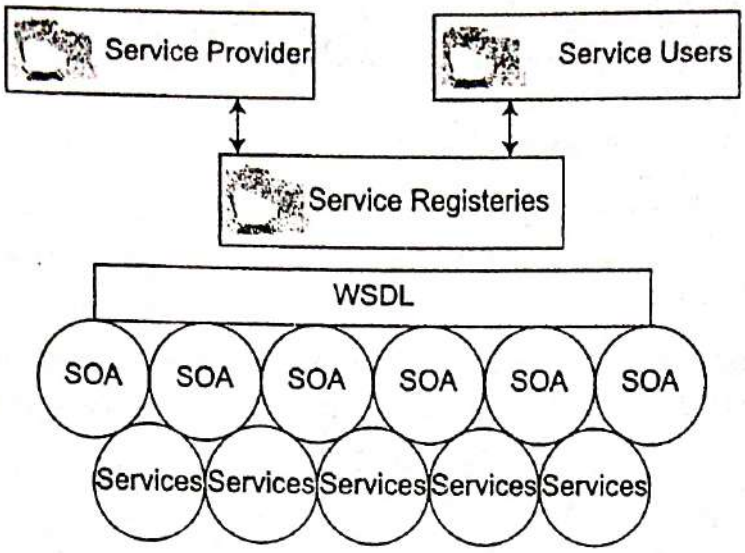


Figure 24.7 Web Services

There are three types of Players in any service-oriented interaction: service providers (who provide the service), service users (who actually use the services provided) and service registries (who act as intermediary between service provider and service users). Service providers register their service in service registries. Service users will search the service registries for suitable service and if found, use those service according to the terms and conditions.

24.8 SOFTWARE AS A SERVICE

Software as a Service (SAAS) is a model of software delivery (Figure 24.8), where the software company (Software Provider) provides maintenance, daily technical operation, and support for the software provided to their client (Source: Wikipedia).

SAAS may use Service-oriented architecture (SOA). Salesforce.com, who is a CRM service provider, is the best example of SAAS.

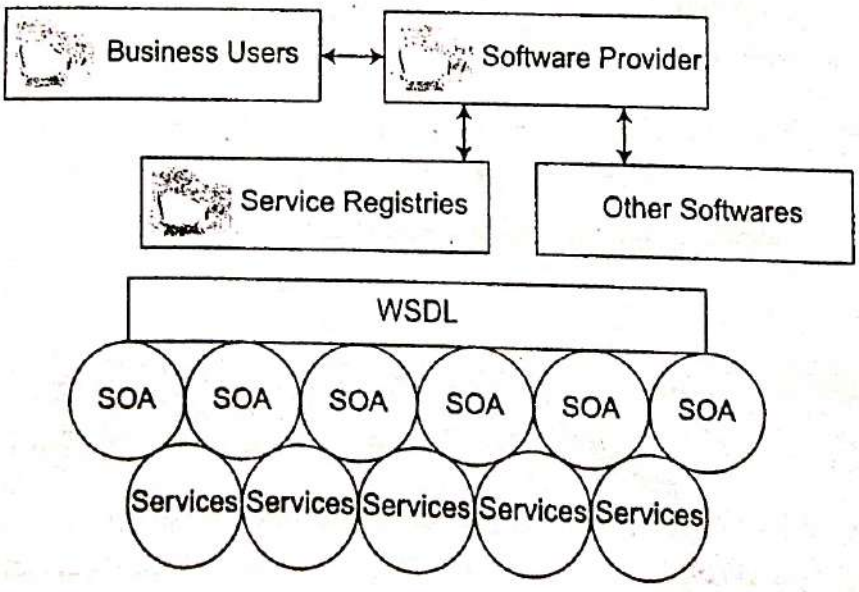


Figure 24.8 Software as a Service (SAAS)

Key characteristics of SAAS are as follows:

1. Software is available through internet
2. Services charged on pay per user basis
3. Activities are managed at central location
4. Upgradation of services happens to all at the same time
5. Installation not required and free maintenance
6. No upfront licensing and usage-based cost

Disadvantages of SAAS are as follows:

1. It fully depends on broadband connection
2. Customization is little
3. Not feasible for high volume of data
4. Security is still an issue.

24.9 SERVICE-ORIENTED ARCHITECTURE

SOA does not provide full business as service, instead a small process (part of the overall service). It is a manufacturing hub, where it allows applications to exchange data with one another. SOA offers services to other applications, but does not provide services to the end users; Whereas SAAS offers services directly to the end users. SOA is an architectural style of building SAAS. Only industry-proven technology is used in building SOA, which helps communicate with another application easily. Services are cataloged and are also discoverable. SOA may service SAAS.

24.10 CLOUD COMPUTING

IT infrastructure as a service is called as cloud computing (Figure 24.9). Cloud application may be delivered using SAAS. In cloud computing, infrastructure itself is programmable (customizable). Dynamic scalability is easily achieved in cloud computing. Basic components of client computing

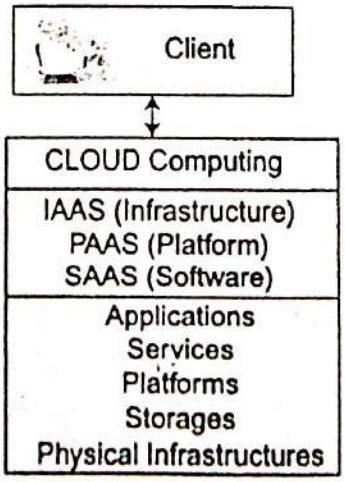


Figure 24.9 Cloud Computing

are (1) clients; (2) services; (3) application; (4) platform (internet infrastructure); (5) storage; and (6) physical infrastructure.

IT infrastructure as a service (IAAS), IT platform as a service (PAAS) and IT SAAS form the architecture of cloud computing (Cloud computing Layers). SAAS is accessed by end users; PAAS is accessed by developers; and IAAS is accessed by infrastructure engineers.

24.11 ASPECT-ORIENTED SOFTWARE DEVELOPMENT (AOSD)

Aspect means concerns. Concerns of the system are separately addressed as an Object (in addition to functional Objects) and are being addressed throughout the software life cycle. The entire system is modularized into Objects based on functionality, and a special object called as Aspect is also created, which represents the aspects of the entire system (in turn the aspects of the objects). It addresses each concern separately so that overall concerns of the system are being addressed. When all the concerns are addressed, the system is developed automatically.

24.12 TEST-DRIVEN DEVELOPMENT (TDD)

In this concept, a testing code is written separately in order to test a software Code (Figure 24.10). The testing code is written even before the actual code is written. But when the testing code is compiled, it fails (Test Fail) as the calling code (actual code) is not available.

Now, the developer writes the basic skeleton of the code (Write Code) and then compiles the testing Code, which will now get compiled. But when it is invoked, it will not give out the correct result, as the actual code is not written completely. Now, the actual code is strengthened (Refactor Coding) for actual implementation. Now, when we invoke the testing code, it will not only run but also gives the results correctly. Testing reusability is the main advantage of TDD, and it also ensures that all the code snippets are thoroughly tested before the delivery.

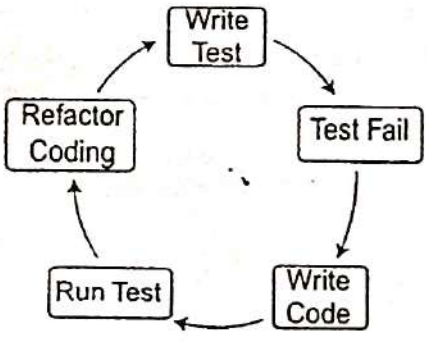


Figure 24.10 Test-driven Development (TDD).

24.13 SOCIAL COMPUTING

Social computing means using the social software to connect with society. There are two main components of social computing: (1) Contents and (2) Connections. Contents are getting shared through (by) Connections, for example, Blog, Face book, Twitter, LinkedIn, wikis, instant messaging (Skype, Google talk), etc. Online identity is the identity of a person in social media (it refers to the profile of the users). It is also possible to use pseudo names here. Social computing is usually used for marketing, branding, collaboration, and communication purposes.

- Web 2.0 is considered as a collaborative technology, which is more than the static web pages where users take part collaboratively and contribute to the content; websites are created using dynamic web pages.
- Rapid Delivery: Software engineering principles (life cycles) are being followed in parallel rather than in sequence, as the customer wants to see the outcome quickly. Agile software development is the best example for Rapid delivery.
- Open Source Software Development: As the name suggests, Open source not only gives the executable code to the users but also the source code to the users (Free license), which has a lot of advantages.
- Security engineering is a specialized field of engineering that focuses on the security aspects of the software system. Security is given high priority during the requirement analysis and designing the stage of the system itself, so that we ensure nobody else hacks (intrude) into the system and modifies the code maliciously.
- Service-oriented Software Engineering (SOSE) is the development of software system using reusable components and services provided by other providers.
- Web Service concepts are famous because of SOSE. Services usually do not have interfaces and are independent of each other; and depend solely on the input messages that are being passed on to it, which are usually in the form of XML.
- Software as a Service (SAAS) is a model of Software delivery, where the software company (Software Provider) provides maintenance, daily technical operation and support for the software provided to their client (Source: Wikipedia).
- IT infrastructure as a service is called as cloud computing. Cloud application may be delivered using SAAS. In Cloud computing, infrastructure itself is programmable (Customizable). Dynamic scalability is easily achieved in Cloud computing.
- Aspect-Oriented Software Development (AOSD): Aspect means concerns. Concerns of the system are separately addressed as an Object (in addition to functional Objects) and are being addressed throughout the software life cycle.
- Test-Driven Development (TDD): In this concept, a testing code is written separately in order to test a software code. The testing code is written even before the actual code is written.
- Social Computing: It means using the social software to connect with society. There are two main components of social computing: (1) Contents and (2) Connections. Contents are getting shared through (by) Connections.

1. Which of the following is not an example of Rapid Delivery?

   A. SCRUM Methodology
   B. XP Methodology
   C. DSDM (Dynamic Software Development Methodology)
   D. Iterative

Answer: D