Unit 8: Introduction to Engineering Software trends and Technology [5Hrs]

8.1 Agile development

- The key idea of Agile Programming is to construct software in a series of short (typically two to four weeks) iterations. Each iteration has its own planning, design, coding, testing, and release phases.
- Agile Programming emphasizes quick changes to design and verbal, rather than written, communication among team members.
- Agile Programming is quite similar to the Spiral software development methodology except that Agile has shorter iteration times. Agile Programming is particularly effective on smaller projects (say, 10 or fewer team members), non-critical projects (in other words, not projects such as medical or weapons systems), and non-distributed systems (in other words, when all team members are in one physical location).
- The Agile mindset promotes evolutionary changes throughout the SDLC. It does so by emphasizing:
    - **Rapid development and delivery of small chunks of useable software**
    - **Quick integration of changes in requirements**: changes have to be dealt with appropriately without having to go through the lengthy process of a change management board
    - **Close collaboration between users, analysts, and programmers**: they have to be available all the time at the same location and communicate face-to-face
    - **Empowering team members**: responsibility is based on trust, not on authority
    - **Self-organizing teams**: thanks to trust and close communication, team members can easily step up or down according to the issue at hand. There is no guru, and no fifth wheel.

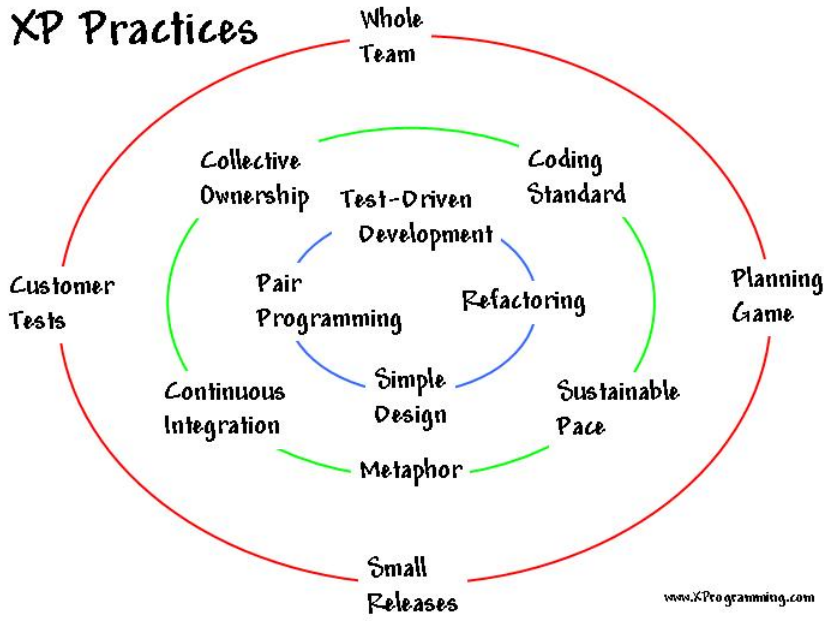
- **The downsides of Agile methods**
    - Even more than with non-Agile methods, the success of the project heavily depend on the motivation, know-how, team abilities, and common sense of team members. This is a direct result of empowering team members to make decisions that can deeply affect the project without going through “committees” and “boards”.
    - The success of agile methods depends on its fit with the organizational culture.
    - Agile methods are often perceived as synonyms to cowboy-style, opening the door to excess of creativity. Although this is (generally) a misconception, it means that team members have to be (re-)educated concerning the agile methods before starting the project.
    - Agile methods lack credibility for senior management. As a result, it may require strong negotiation power and increased risks for the project manager to use agile

methods for strategic projects. Most companies will only allow agile methods for small exploratory projects at first.

- Agile methods may not be appropriate for large-scale projects

8.2 Extreme Programming

- Extreme Programming is most often considered a particular form of Agile Programming, although some engineers prefer to think of Extreme Programming as a separate methodology altogether.
- Extreme Programming is more specific the Agile Programming but in practice no two groups which use Extreme Programming seem to agree on exactly which characteristics are essential.
- From a practical point of view, Extreme Programming encourages (or requires depending on your point of view) "Pair Programming" where developers always work in pairs, and "Test-Driven Development" where unit tests are always created in conjunction with the system under development.
- Extreme Programming may be a good choice for prototype technologies and research projects.
- **Extreme Programming** (or **XP**) is a software engineering methodology (and a form of agile software development) prescribing a set of daily stakeholder practices that embody and encourage particular XP values. Proponents believe that exercising these practices—traditional software engineering practices taken to so-called "extreme" levels—leads to a development process that is more responsive to customer needs ("agile") than traditional methods, while creating software of better quality.



- Proponents of Extreme programming and agile methodologies in general regard ongoing changes to requirements as a natural, inescapable and desirable aspect of software development projects; they believe that adaptability to changing requirements at any point during the project life is a more realistic and better approach than attempting to define all requirements at the beginning of a project and then expending effort to control changes to the requirements.

XP values

- Extreme Programming initially recognized four values in 1999. A new value was added in the second edition of *Extreme Programming Explained*. The five values are:
    - Communication
    - Simplicity
    - Feedback
    - Courage
    - Respect

8.3 Cloud Computing and Grid Computing

- Cloud computing and grid computing are two relatively new concepts in the field of computing.
- Both grid and cloud computing are networks which abstract processing tasks. Abstraction masks the actual complex processes taking place within a system, and presents a user with a simplified interface with which they can interact easily. The idea is to be able to make the system more user-friendly whilst retaining all the benefits of more complicated processes.

Cloud Computing

- Cloud computing is an extension of the object-oriented programming concept of abstraction. Abstraction removes the complex working details from visibility. All that is visible is an interface, which receives inputs and provides outputs. How these outputs are computed is completely hidden.
- For example, a car driver knows that a steering wheel with turn the car in the direction they want to go; or that pressing the accelerator will cause the car to speed up. The driver is usually unconcerned about how the directions of the steering wheel and the accelerator pedal are translated into the actual motion of the car. Therefore, these details are abstracted from the driver.
- A cloud is similar; it applies the concept of abstraction in a physical computing environment, by hiding the true processes from a user. In a cloud computing environment, data can exist on multiple servers, details of network connections are hidden and the user is none the wiser. In fact, cloud computing is so named because a cloud is often used to depict inexact knowledge of inner workings.

- Cloud computing derives heavily from the Unix paradigm of having multiple elements, each excellent at one particular task, rather than have one massive element which isn't as good.

Advantages of Cloud Computing

- **Cost Efficient**
  - Cloud computing is probably the most cost efficient method to use, maintain and upgrade. Traditional desktop software costs companies a lot in terms of finance. Adding up the licensing fees for multiple users can prove to be very expensive for the establishment concerned. The cloud, on the other hand, is available at much cheaper rates and hence, can significantly lower the company's IT expenses. Besides, there are many one-time-payments, pay-as-you-go and other scalable options available, which make it very reasonable for the company in question.
- **Almost Unlimited Storage**
  - Storing information in the cloud gives you almost unlimited storage capacity. Hence, you no more need to worry about running out of storage space or increasing your current storage space availability.
- **Backup and Recovery**
  - Since all your data is stored in the cloud, backing it up and restoring the same is relatively much easier than storing the same on a physical device. Furthermore, most cloud service providers are usually competent enough to handle recovery of information. Hence, this makes the entire process of backup and recovery much simpler than other traditional methods of data storage.
- **Automatic Software Integration**
  - In the cloud, software integration is usually something that occurs automatically. This means that you do not need to take additional efforts to customize and integrate your applications as per your preferences. This aspect usually takes care of itself. Not only that, cloud computing allows you to customize your options with great ease. Hence, you can handpick just those services and software applications that you think will best suit your particular enterprise.
- **Easy Access to Information**

- Once you register yourself in the cloud, you can access the information from anywhere, where there is an Internet connection. This convenient feature lets you move beyond time zone and geographic location issues.

- **Quick Deployment**
  - Lastly and most importantly, cloud computing gives you the advantage of quick deployment. Once you opt (choose) for this method of functioning, your entire system can be fully functional in a matter of a few minutes. Of course, the amount of time taken here will depend on the exact kind of technology that you need for your business.

- **Resiliency and Redundancy**

  - A cloud deployment is usually built on a robust architecture thus providing resiliency and redundancy to its users. The cloud offers automatic failover between hardware platforms out of the box, while disaster recovery services are also often included.

Disadvantages of Cloud Computing

- **Technical Issues**
  - Though it is true that information and data on the cloud can be accessed anytime and from anywhere at all, there are times when this system can have some serious dysfunction (disturbance in function). You should be aware of the fact that this technology is always prone to outages and other technical issues. Even the best cloud service providers run into this kind of trouble, in spite of keeping up high standards of maintenance. Besides, you will need a very good Internet connection to be logged onto the server at all times. You will invariably be stuck in case of network and connectivity problems.
- **Security in the Cloud**
  - The other major issue while in the cloud is that of security issues. Before adopting this technology, you should know that you will be surrendering all your company's sensitive information to a third-party cloud service provider. This could potentially put your company to great risk. Hence, you need to make absolutely sure that you choose the most reliable service provider, who will keep your information totally secure.
- **Prone to Attack**

- Storing information in the cloud could make your company vulnerable to external hack attacks and threats. As you are well aware, nothing on the Internet is completely secure and hence, there is always the lurking possibility of stealth of sensitive data.

- **Dependency and vendor lock-in**

  - One of the major disadvantages of cloud computing is the implicit dependency on the provider. This is what the industry calls "vendor lock-in" since it is difficult, and sometimes impossible, to migrate from a provider once you have rolled with him. If a user wishes to switch to some other provider, then it can be really painful and cumbersome to transfer huge data from the old provider to the new one. This is another reason why you should carefully and thoroughly contemplate all options when picking a vendor.

- Limited control and flexibility

  - Since the applications and services run on remote, third party virtual environments, companies and users have limited control over the function and execution of the hardware and software. Moreover, since remote software is being used, it usually lacks the features of an application running locally.

Grid Computing

- Grid computing has been around for over 12 years now and its advantages are many.
- Grid computing can be defined in many ways but for these discussions let's simply call it a way to execute compute jobs (e.g. perl scripts, database queries, etc.) across a distributed set of resources instead of one central resource.
- Grid computing harnesses the idle processing power of various computing units, and uses that processing power to compute one job. The job itself is controlled by one main computer, and is broken down into multiple tasks which can be executed simultaneously on different machines. These tasks needn't be mutually exclusive, although that is the ideal scenario. As the tasks complete on various computing units, the results are sent back to the controlling unit, which then collates them forming a cohesive output.
- The advantage of grid computing is two-fold: firstly, unused processing power is effectively used, maximizing available resources and, secondly, the time taken to complete the large job is significantly reduced.
- The ability to distribute jobs to many smaller server components using load sharing software that distributes the load evenly based on resource availability and policies. Now instead of having one heavily burdened server the load can be spread evenly across many

smaller computers. The distributed nature of grid computing is transparent to the user. When a user submits a job they don't have to think about which machine their job is going to get executed on. The "grid software" will perform the necessary calculations and decide where to send the job based on policies.

Advantage

- No need to buy large servers for applications that can be split up and farmed out to smaller commodity type servers. Results can then be concatenated and analyzed upon job(s) completion.
- Much more efficient use of idle resources. Jobs can be farmed out to idle servers or even idle desktops. Many of these resources sit idle especially during off business hours. Policies can be in places that allow jobs to only go to servers that are lightly loaded or have the appropriate amount of memory/CPU characteristics for the particular application.
- Grid environments are much more modular and don't have single points of failure. If one of the servers/desktops within the grid fails there are plenty of other resources able to pick the load. Jobs can automatically restart if a failure occurs.
- Policies can be managed by the grid software. The software is really the brains behind the grid. A client will reside on each server which sends information back to the master telling it what type of availability or resources it has to complete incoming jobs.
- This model scales very well. Just plug them in by installing grid client on additional desktops or servers. They can be removed just as easily on the fly. This modular environment really scales well.
- Upgrading can be done on the fly without scheduling downtime. Since there are so many resources some can be taken offline while leaving enough for work to continue. This way upgrades can be cascaded as to not affect ongoing projects.
- Jobs can be executed in parallel speeding performance. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes.

Disadvantages

- For memory hungry applications that can't take advantage of MPI you may be forced to run on a large SMP.
- You may need to have a fast interconnect between compute resources (gigabit Ethernet at a minimum).
- Some applications may need to be tweaked to take full advantage of the new model.
- Licensing across many servers may make it prohibitive for some apps. Vendors are starting to be more flexible with environment like this.
- Political challenges associated with sharing resources (especially across different admin domains). Many groups are reluctant with sharing resources even if it benefits everyone

involved. The benefits for all groups need to be clearly articulated and policies developed that keeps everyone happy.

Areas that already are taking good advantage of grid computing include bioinformatics, cheminformatics, oil & drilling, and financial applications.

Cloud Computing vs. Grid Computing

- In fact, they are both used to economize computing by maximizing existing resources. Additionally, both architectures use abstraction extensively, and both have distinct elements which interact with each other.
- However, the difference between the two lies in the way the tasks are computed in each respective environment.
    - In a computational grid, one large job is divided into many small portions and executed on multiple machines. This characteristic is fundamental to a grid; not so in a cloud.
    - The computing cloud is intended to allow the user to avail (benefit) of various services without investing in the underlying architecture.
    - While grid computing also offers a similar facility for computing power, cloud computing isn't restricted to just that.
    - A cloud can offer many different services, from web hosting, right down to word processing. In fact, a computing cloud can combine services to present a user with a homogenous optimized result.

8.4 Enterprise Mobility

- Enterprise mobility is the trend toward a shift in work habits, with more employees working out of the office and using mobile devices and cloud services to perform business tasks.

- The term refers not only to mobile workers and mobile devices, but also to the mobility of corporate data. An employee may upload a corporate presentation from his or her desktop PC to a cloud storage service, and then access it from a personal iPad to show at a client site.

- Enterprise mobility can improve employee productivity, but it also creates security risks.

- Enterprise mobility management products, such as data loss prevention technologies, are available to help IT departments address these risks. A strong acceptable use policy for employees can also contribute to a successful enterprise mobility strategy.

The Pros

- Using mobility enterprise is flexible.
- They help in combining personal and professional work and also provide employees with the freedom to work away from their office workstations. This in turn facilitates in improving the productivity of an organization.
- There is financial benefit of saving money that would be spent on supplying corporate mobile devices.

The Cons

- A bit more complicated-especially in terms of security.
- Consumer mobile devices are simply not designed for use in enterprises, and statistics show that employee-owned devices are exposed to twice the security risks of employee-owned devices.
- In addition to the risk of cyber threats, there are also many incidents where companies have lost confidential information when mobile devices were stolen or when employees lost their devices.

8.5 Business Intelligent and Approaches

8.5.1 ERP, Supply Chain Management, Service Oriented Architecture and web services

ERP

- ERP is short for enterprise resource planning.
- Enterprise resource planning (ERP) is business process management software that allows an organization to use a system of integrated applications to manage the business and automate many back office functions related to technology, services and human resources.
- ERP software integrates all facts of an operation, including product planning, development, manufacturing, sales and marketing.
- ERP is an industry acronym for Enterprise Resource Planning. Broadly speaking, ERP refers to automation and integration of a company's core business to help them focus on effectiveness & simplified success.
- Enterprise resource planning (ERP) is a business management software—usually a suite of integrated applications—that a company can use to collect, store, manage and interpret data from many business activities, including:
    - Product planning, cost

- Manufacturing or service delivery
- Marketing and sales
- Inventory management
- Shipping and payment

- ERP provides an integrated view of core business processes, often in real-time, using common databases maintained by a database management system.
- ERP systems track business resources—cash, raw materials, production capacity—and the status of business commitments: orders, purchase orders, and payroll.
- The applications that make up the system share data across the various departments (manufacturing, purchasing, sales, accounting, etc.) that provide the data.
- ERP facilitates information flow between all business functions, and manages connections to outside stakeholders
- Enterprise resource planning (ERP) is an industry term for the broad set of activities that helps an organization manages its business.
- An important goal of ERP is to facilitate the flow of information so business decisions can be data-driven.
- ERP software suites are built to collect and organize data from various levels of an organization to provide management with insight into key performance indicators (KPIs) in real time.

Characteristics

- ERP include following characteristics
  - An integrated system that operates in (or near) real time without relying on periodic updates
  - A common database that supports all applications
  - A consistent look and feel across modules
  - Installation of the system with elaborate application/data integration by the Information Technology (IT) department, provided the implementation is not done in small steps

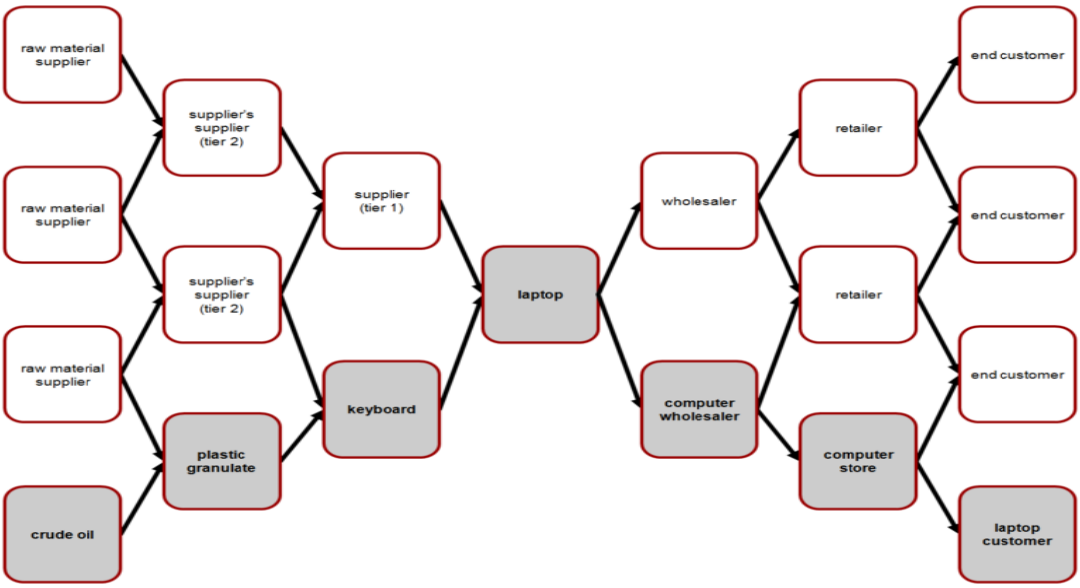
Benefits of ERP for your Business

- **Integration across all business processes** - To realize the full benefits of an ERP system it should be fully integrated into all aspects of your business from the customer facing front end, through planning and scheduling, to the production and distribution of the products you make.
- **Automation enhances productivity** - By automating aspects of business processes, ERP makes them more efficient, less prone to error, and faster. It also frees up people from mundane tasks such as balancing data.

- **Increase overall performance -** By integrating disparate business processes, ERP ensures coherence and avoids duplication, discontinuity, and people working at cross purposes, in different parts of the organization. The cumulative positive effect when business processes integrate well is overall superior performance by the organization.
- **Quality Reports and Performance analysis -** Analysis on ERP will enable you to produce financial and boardroom quality reports, as well as to conduct analysis on the performance of your organization.
- **Integrates across the entire supply chain** - A best of breed ERP system should extend beyond your organization and integrate with both your supplier and customer systems to ensure full visibility and efficiency across your supply chain

Supply Chain management

- Supply chain management (SCM) is the oversight of materials, information, and finances as they move in a process from supplier to manufacturer to wholesaler to retailer to consumer.
- Supply chain management involves coordinating and integrating these flows both within and among companies.
- It is said that the ultimate goal of any effective supply chain management system is to reduce inventory (with the assumption that products are available when needed).
- As a solution for successful supply chain management, sophisticated software systems with Web interfaces are competing with Web-based application service providers (ASP) who promise to provide part or all of the SCM service for companies who rent their service.
- Supply chain management flows can be divided into three main flows:
    - The product flow
    - The information flow
    - The finances flow
- The product flow includes the movement of goods from a supplier to a customer, as well as any customer returns or service needs. The information flow involves transmitting orders and updating the status of delivery. The financial flow consists of credit terms, payment schedules, and consignment and title ownership arrangements.



- Supply chain management (SCM) is "the systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole."
- It has also been defined as the "design, planning, execution, control, and monitoring of supply chain activities with the objective of creating net value, building a competitive infrastructure, leveraging worldwide logistics, synchronizing supply with demand and measuring performance globally."

- Supply chain management is the streamlining of a business' supply-side activities to maximize customer value and to gain a competitive advantage in the marketplace.
- Supply chain management (SCM) represents an effort by suppliers to develop and implement supply chains that are as efficient and economical as possible.
- Supply chains cover everything from production, to product development, to the information systems needed to direct these undertakings.

Service oriented architecture

- Service-oriented architecture (SOA) is an approach used to create an architecture based upon the use of services.
- Services (such as Web services) carry out some small function, such as producing data, validating a customer, or providing simple analytical services.
- In addition to building and exposing services, SOA has the ability to leverage these services over and over again within applications (known as composite applications).

- SOA binds these services to orchestration, or individually leverages these services. Thus, SOA is really about fixing existing architectures by addressing most of the major systems as services, and abstracting those services into a single domain where they are formed into solutions.
- One of the keys to SOA architecture is that interactions occur with loosely coupled services that operate independently.
- SOA architecture allows for service reuse, making it unnecessary to start from scratch when upgrades and other modifications are needed. This is a benefit to businesses that seek ways to save time and money.
- SOA is known to provide both time-to-market advantages, as well as business agility. The use of orchestration engines, or leveraging development environments that leverage services and SOA, allow those who build applications to do so quickly, since the services provide much of what the application requires. This provides the time-to-market advantage.
- A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.
- Service-oriented architecture (SOA) is a design pattern based on distinct pieces of software providing application functionality as services to other applications via a protocol. This is known as service-orientation. It is independent of any vendor, product or technology.
- A service is a self-contained unit of functionality, such as retrieving an online bank statement. Services can be combined by other software applications to provide the complete functionality of a large software application.
- SOA makes it easy for computers connected over a network to cooperate. Every computer can run an arbitrary number of services, and each service is built in a way that ensures that the service can exchange information with any other service in the network without human interaction and without the need to make changes to the underlying program itself.
- SOA is based on the concept of a service. Depending on the service design approach taken, each SOA service is designed to perform one or more activities by implementing one or more service operations. As a result, each service is built as a discrete piece of code. This makes it possible to reuse the code in different ways throughout the application by changing only the way an individual service interoperates with other services that make up the application, versus making code changes to the service itself. SOA design principles are used during software development and integration.
- Service-Oriented Architecture (SOA) is an architectural style that supports service-orientation. Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services.

- A service:
  - Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports)
  - Is self-contained
  - May be composed of other services
  - Is a "black box" to consumers of the service

SOA Benefits and Disadvantages

- SOA provides a strategic capability for integrating business processes, data, and organizational knowledge. Because interfaces are platform-independent, a client from any device using any operating system in any language can use the service.
- In a service-oriented architecture, clients consume services, rather than invoking discreet method calls directly.
- There are many benefits of SOA, including improved information flow, location transparency, internal software organization and better data translation.
- The most commonly discussed **disadvantage** of SOA is for applications with GUI functionality. These types of applications become more complex when using SOA.

SOA is Not Web Services

- Although it is built on similar principles, SOA is not the same as Web services, which indicates a collection of technologies, such as SOAP and XML. Web services can be thought of as a consumer-provider relationship on the Web while SOA is about designing your architecture to best work in a Web service environment. It is more than a set of technologies and runs independent of any specific technologies.

Web services

- Web services (sometimes called application services) are services (usually including some combination of programming and data, but possibly including human resources as well) that are made available from a business's Web server for Web users or other Web-connected programs.
- Providers of Web services are generally known as application service provider s. Web services range from such major services as storage management and customer relationship management (CRM) down to much more limited services such as the furnishing of a stock quote and the checking of bids for an auction item. The accelerating creation and availability of these services is a major Web trend.
- Users can access some Web services through a peer-to-peer arrangement rather than by going to a central server. Some services can communicate with other services and this exchange of procedures and data is generally enabled by a class of software known as middleware.

- Services previously possible only with the older standardized service known as Electronic Data Interchange (EDI) increasingly are likely to become Web services.
- Besides the standardization and wide availability to users and businesses of the Internet itself, Web services are also increasingly enabled by the use of the Extensible Markup Language (XML) as a means of standardizing data formats and exchanging data. XML is the foundation for the Web Services Description Language (WSDL).
- A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing.
- The W3Cdefines a Web service generally as:- a software system designed to support interoperable machine-to-machine interaction over a network.
- A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

To summarize, a complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

An Example

- Consider a simple account-management and order -processing system. The accounting personnel use a client application built with Visual Basic or JSP to create new accounts and enter new customer orders. The processing logic for this system is written in Java and resides on a Solaris machine, which also interacts with a database to store the information.
- The steps illustrated above are as follows:

    1. The client program bundles the account registration information into a SOAP message.
    2. This SOAP message is sent to the Web Service as the body of an HTTP POST request.

3. The Web Service unpacks the SOAP request and converts it into a command that the application can understand. The application processes the information as required and responds with a new unique account number for that customer.
4. Next, the Web Service packages up the response into another SOAP message, which it sends back to the client program in response to its HTTP request.
5. The client program unpacks the SOAP message to obtain the results of the account registration process. For further details regarding the implementation of Web Services technology, read about the Cape Clear product set and review the product components.

- Web services have special behavioral characteristics

  - XML-based

    - Web Services uses XML at data representation and data transportation layers. Using XML eliminates any networking, operating system, or platform binding. So Web Services based applications are highly interoperable application at their core level.

  - Loosely coupled

    - A consumer of a web service is not tied to that web service directly. The web service interface can change over time without compromising the client's ability to interact with the service. A tightly coupled system implies that the client and server logic are closely tied to one another, implying that if one interface changes, the other must also be updated. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

  - Coarse-grained

    - Object-oriented technologies such as Java expose their services through individual methods. An individual method is too fine an operation to provide any useful capability at a corporate level. Building a Java program from scratch requires the creation of several fine-grained methods that are then composed into a coarse-grained service that is consumed by either a client or another service. Businesses and the interfaces that they expose should be coarse-grained. Web services technology provides a natural way of defining coarse-grained services that access the right amount of business logic.

  - Ability to be synchronous or asynchronous

- Synchronicity refers to the binding of the client to the execution of the service. In synchronous invocations, the client blocks and waits for the service to complete its operation before continuing. Asynchronous operations allow a client to invoke a service and then execute other functions. Asynchronous clients retrieve their result at a later point in time, while synchronous clients receive their result when the service has completed. Asynchronous capability is a key factor in enabling loosely coupled systems.

- Supports Remote Procedure Calls (RPCs)

  - Web services allow clients to invoke procedures, functions, and methods on remote objects using an XML-based protocol. Remote procedures expose input and output parameters that a web service must support. Component development through Enterprise JavaBeans (EJBs) and .NET Components has increasingly become a part of architectures and enterprise deployments over the past couple of years. Both technologies are distributed and accessible through a variety of RPC mechanisms. A web service supports RPC by providing services of its own, equivalent to those of a traditional component, or by translating incoming invocations into an invocation of an EJB or a .NET component.

- Supports document exchange

  - One of the key advantages of XML is its generic way of representing not only data, but also complex documents. These documents can be simple, such as when representing a current address, or they can be complex, representing an entire book or RFQ. Web services support the transparent exchange of documents to facilitate business integration.

8.5.2 Enterprise Portal and Content management

- **Enterprise Content Management (ECM)** is a formalized means of organizing and storing an organization's documents, and other content, that relate to the organization's processes. The term encompasses strategies, methods, and tools used throughout the lifecycle of the content
- In late 2005: Enterprise content management is the technology used to capture, manage, store, preserve, and deliver content and documents related to organizational processes.
- In Early 2010: Enterprise Content Management (ECM) is the strategies, methods and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM covers the management of information within the entire scope of an enterprise whether that information is in the form of a paper document, an electronic file, a database print stream.

- **Content Management** is essentially the process of managing any type or 'unit' of digital information (such as text, images, graphics, video, sound, documents, records etc) in such a way that its electronic storage is deemed to be 'managed' rather than 'un-managed'.
- A CMS is a software tool that allows you to create, edit, and publish content. While early CMS software was used to manage documents and local computer files, most CMS systems are now designed exclusively to manage content on the Web.
- The goal of a CMS is to provide an intuitive user interface for building and modifying webpage content. Each CMS also provides a web publishing tool that allows one or more users to publish updates live on the Web. The editing component is called the content management application (CMA), while the publishing tool is called the content delivery application (CDA). These two components are integrated together in a CMS to streamline the web development process.
- Content management systems are available as installable applications and web-based user interfaces. While CMS software programs, such as Adobe Contribute, were popular for a few years, they have largely been replaced by web-based CMS. Most people prefer a web interface, since it simplifies the website updating process. Additionally, most web-based CMS are updated automatically, ensuring all users have the latest tools to manage their content.
- Several web-based CMS tools are available. The following are some of the most popular ones:
    - **WordPress** - free web software designed for creating template-based websites or blogs
    - **Blogger** - Google's blogging tool designed specifically for maintaining a blog
    - **Joomla** - a flexible web publishing tool that supports custom databases and extensions
    - **Drupal** - an open source platform often used for developing community-based sites
    - **Weebly** - a web-based platform for building simple personal and business websites
    - **Wix** - a collection of web publishing tools for creating a highly customizable website
- Some CMS tools are free to use, while others require a monthly fee. Many CMS provide free basic components, but charge for high-quality templates, web hosting, custom domain names, or other features. Before deciding on a CMS, it is a good idea to review multiple options so you can choose the one that best fits your website goals.
- A content management system (CMS) is a system used to manage the content of a Web site.
- Typically, a CMS consists of two elements: the content management application (CMA) and the content delivery application (CDA). The CMA element allows the content manager or author, who may not know Hypertext Markup Language (HTML), to manage

the creation, modification, and removal of content from a Web site without needing the expertise of a Webmaster. The CDA element uses and compiles that information to update the Web site. The features of a CMS system vary, but most include Web-based publishing, format management, revision control, and indexing, search, and retrieval.

8.6 Introduction to OOSE

- Object-Oriented Software Engineering (OOSE) is a software design technique that is used in software design in object-oriented programming.
- OOSE is developed by Ivar Jacobson in 1992.
- OOSE is the first object-oriented design methodology that employs use cases in software design.
- OOSE is one of the precursors of the Unified Modeling Language (UML), such as Booch and OMT.
- It includes a requirement, an analysis, a design, an implementation and a testing model.
- Interaction diagrams are similar to UML's sequence diagrams. State transition diagrams are like UML state chart diagrams.

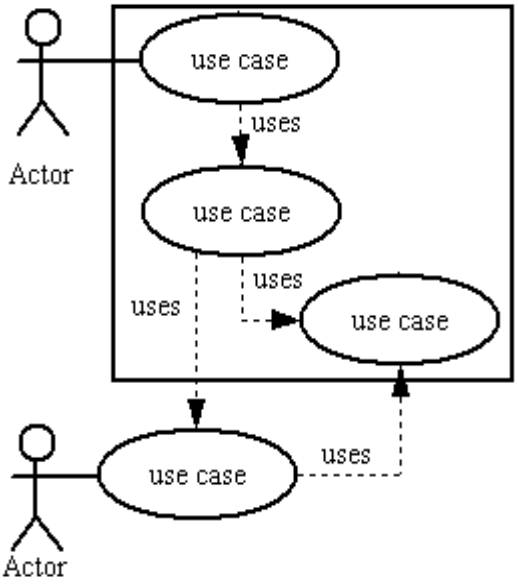


Figure. Use Case diagram.