MODULE I
INTRODUCTION TO ENTERPRISE APPLICATION

Enterprise Architecture - Life Cycle, Development Framework, architectural model- Conceptual Layers, Enterprise IT architecture domain, Enterprise Resource Planning (ERP)-Customer Relationship management (CRM)-Supply Chain Management (SCM) and HRM; Enterprise Java- Introduction to web application and its life cycle.

Enterprise Architecture

Enterprise architecture is a precise method for doing the business assessment, designing, management, and execution utilizing a complete approach at every phase in order to establish and execute strategy successfully.

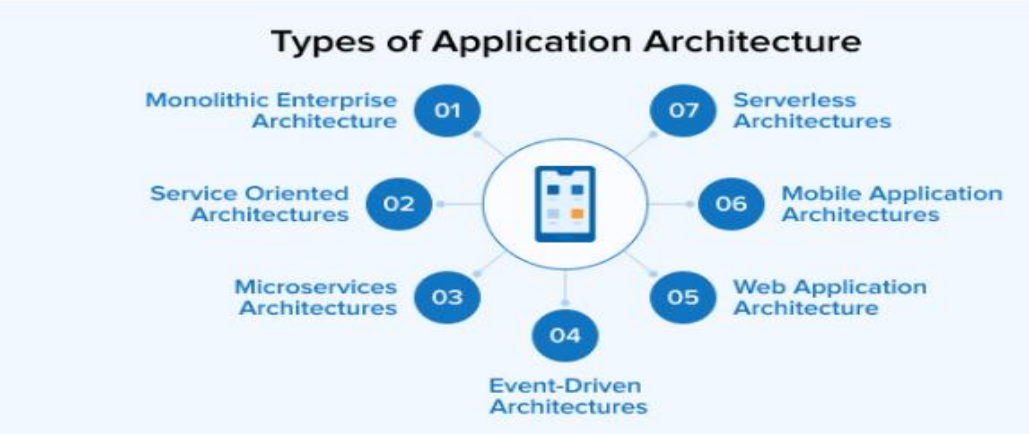
Why Enterprise Architecture is Important?

New technology is evolving rapidly and developments like AI, machine learning, and IoT are making IT systems more complex at a higher speed than ever. Digital Enterprise Management helps a company stay on top of all this new tech and where it fits into the business strategy.

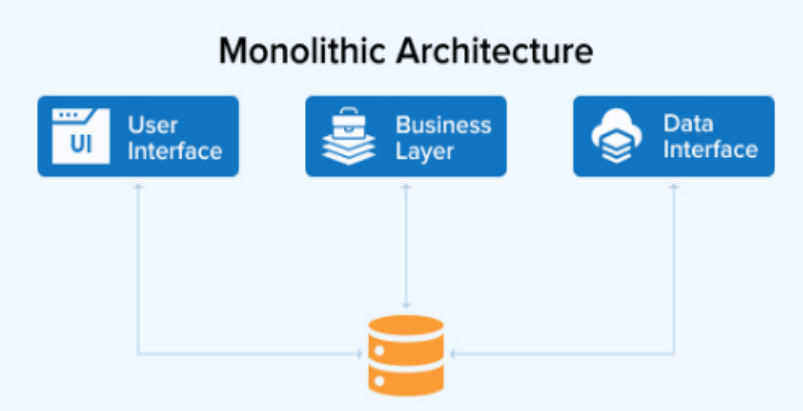
EA has the potential to bridge the business and IT teams by providing a unifying language centered on capabilities and business outcomes. It clarifies how all the components of the organization work together and makes critical interdependencies visible.

Enterprise Architecture shows a business where it is and how it is doing, which helps a company understand where it wants to be and how it might get there. Plus, when an organization understands how systems work together, it can ensure IT investments are tied to those goals.

Types of enterprise application architecture



Monolithic Enterprise Architecture



Monolithic enterprise software architecture is linked with old systems which were designed before the advent of newer services-oriented frameworks. In this design, every function is self-contained. Therefore, any changes must be reviewed and revised throughout the whole program. There are several explanations why this strategy is hardly adopted anymore. For example, monolithic software is complicated, unable to grow, and tough to update. Nevertheless, it is ideal for tiny enterprise applications with little operability and low-traffic utilities like online calculators and blogs.

Service Oriented Architectures

Service oriented architecture is an evolution of Message Bus Architecture. This architecture's advancement is the awareness that many corporate tasks with significant precision are automated in numerous locations.

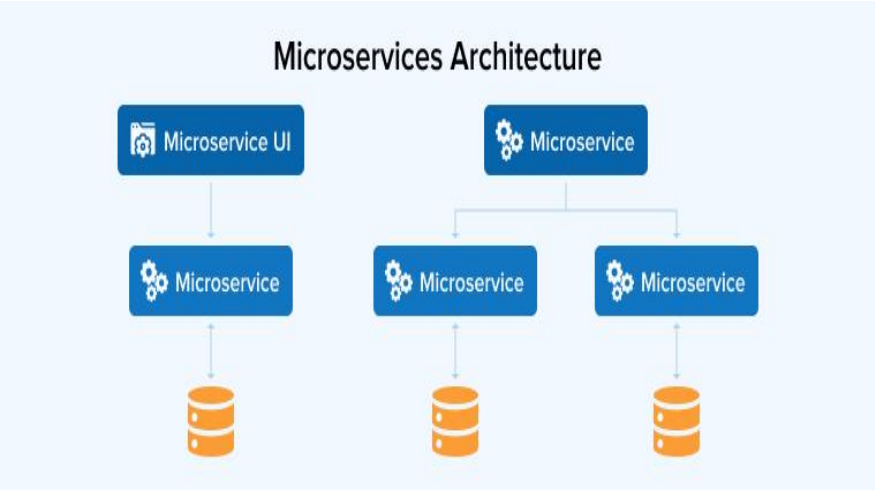
The majority of the programs will incorporate a user interface, the majority will be concerned with protection, the majority will execute some type of business logic, etc. The service oriented architectures suggest that the software should be refactored, and these parts of operation should be eliminated and represented as a single 'service' that can be accessed in real-time. So, for instance, assessment becomes the role of the information delivery service, which could be incorporated via a database system, the user interface may be assigned to a portal service, security features will be applied by a verification and authorization service, and application logic may be performed by a business rules service.

The most probable choices for service orientation are often business-neutral, due in part to the prevalence of these services throughout the application inventory, as a result of building a service.

Exemplified to its logical extreme, service orientation enables programs to abdicate duty for safety, business logic, execution, delivery, leaving just data storage and configuration. Using a middleman to establish service orientation in a communication context is optional and not required. The majority of the existing

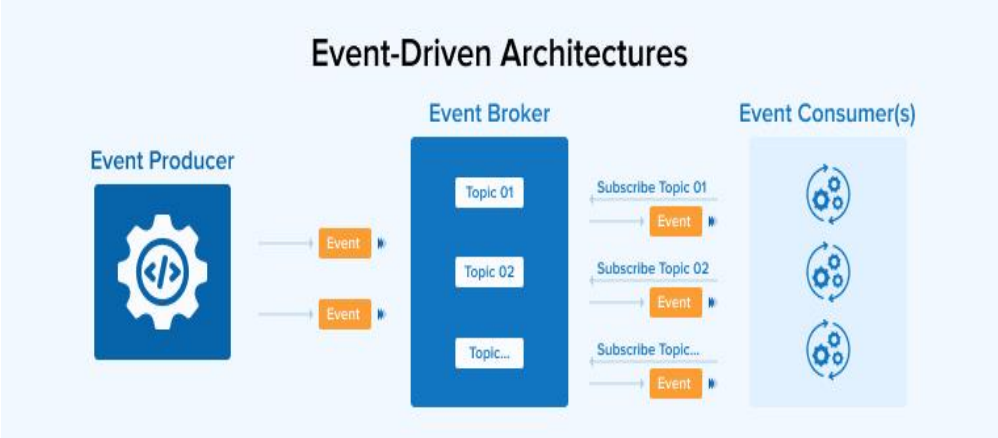
literature conflates the technology of successful execution with the notion, particularly when the tech of execution is Web services.

Microservices Architectures



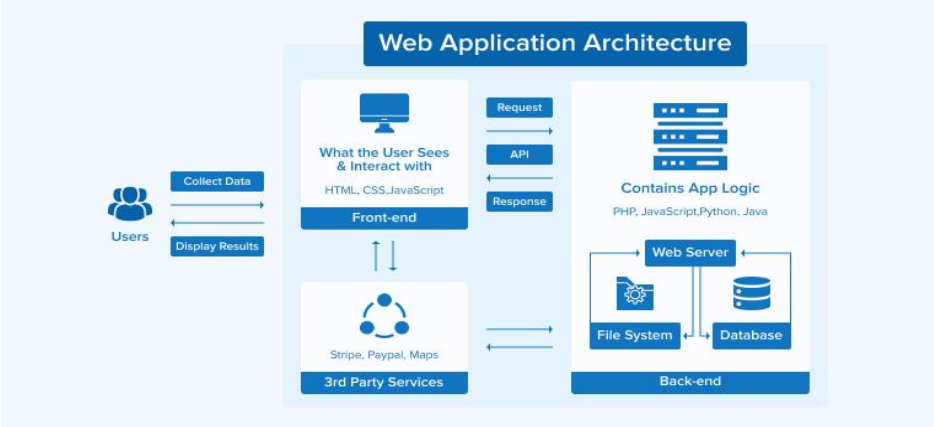
Microservices designs are extensively employed in cloud-native, agile methodologies settings, such as DevOps, which favor the utilization of microservices. With this methodology, programs are broken down into the tiniest, most loosely linked, operationally independent, and most reusable modules feasible. Enterprise software developers construct apps from microservices, hence accelerating software development. The resultant applications are scalable and robust, as the loss of a single service does not take down the program as a whole. They also allow for the introduction of upgrades without disturbance. Numerous programmers can simultaneously work on a single project.

Event-Driven Architectures



In real-time computing and self-service settings, event-driven architectures are commonly employed. Rather than handling data in chunks on a predetermined timetable, event-driven systems react to occurrences such as the touch of a button, the scan of a bank card, or the measurement of a device's temperature. Event-driven systems are frequently implemented on top of microservices enterprise architecture due to the fact that one event initiates a series of distinct activities related to the action.

Web Application Architecture



The most recent technological development is Web services. Web services are poised to become the embodiment of the Message Bus Architecture depending on open standards. Where programs now use a vendor-specific adapter to connect with the Message Bus, a common Web service interface will be implemented.

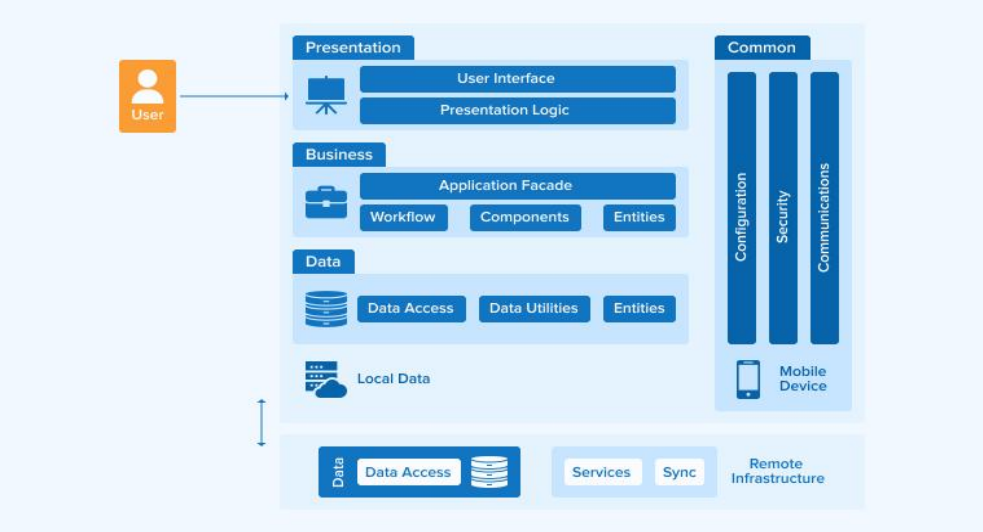
Whereas the Message Bus Architecture handles message routes utilizing a bespoke extensional routing or orchestration tool or with explicit publication logic, the web design will employ the equivalent web service standard, currently BPEL4WS. In contrast to the Message Bus Architecture, which provides promised delivery using proprietary queuing methods like IBM-MQSeries and many others, the Web service architecture will employ emerging protocols like HTTP-R or Web services reliable messaging.

Web services standards are now insufficient and do not completely overlay the offers of private companies, but the promise is apparent that Web services will soon provide an open standards substitute. Web services are juncture links by nature. This will construct a state-of-the-art version of the default architecture, with enterprise software firmly coupled to one another via a large number of unregulated interfaces if used incorrectly. In consequence, the orchestrated Web service architecture provides a broker to whom all Web service queries are directed, and that is accountable for passing those responses to the programs delivering the service.

This centralized orchestration enables the Web service architecture to remain disconnected. Likewise, by implementing asynchronous request/reply logic that is, the client does not stall while awaiting the response and by improving the normal Web service request via HTTP with assured delivery, the broker is capable of

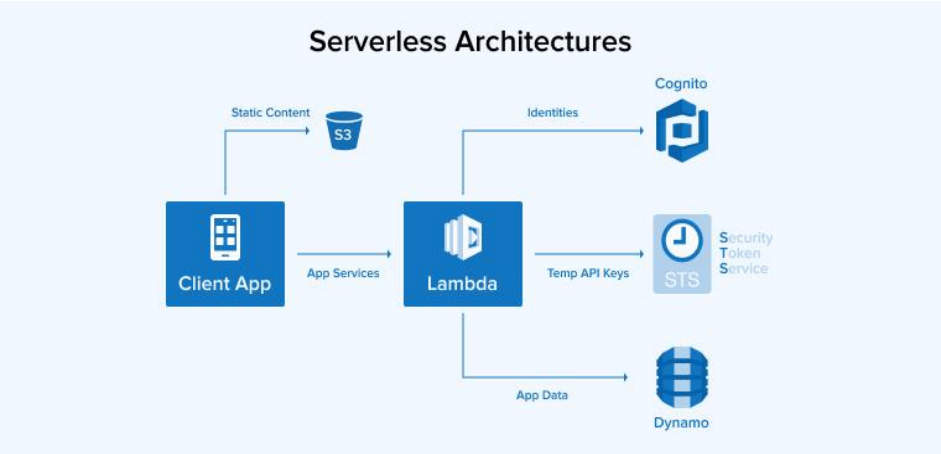
establishing an ecosystem comparable to the Message Bus Architecture. The Web service architecture is now functioning and supported by a number of unique solutions. It is superior to the Message Bus Architecture because it is built on open standards, hence avoiding vendor lock-in.

Mobile Application Architectures



Similar to their web-based counterparts, mobile application designs take into account the increased processing, memory, and storage capabilities of mobile devices. They also define structures for platform portability.

Serverless Architectures



The most recent iteration of the microservices concept serverless architecture, is still uncommon. Using this methodology, applications are built using cloud-based third-party services that operate within enterprise software containers. Serverless services are scalable and can be rapidly set up and taken off. As cloud servers are not required, serverless deployment is also one of the most cost-effective methods for deploying

enterprise software. Event processing, image recognition, automated software testing, and machine translation are popular applications.

Enterprise Architecture Life Cycle (EALC)

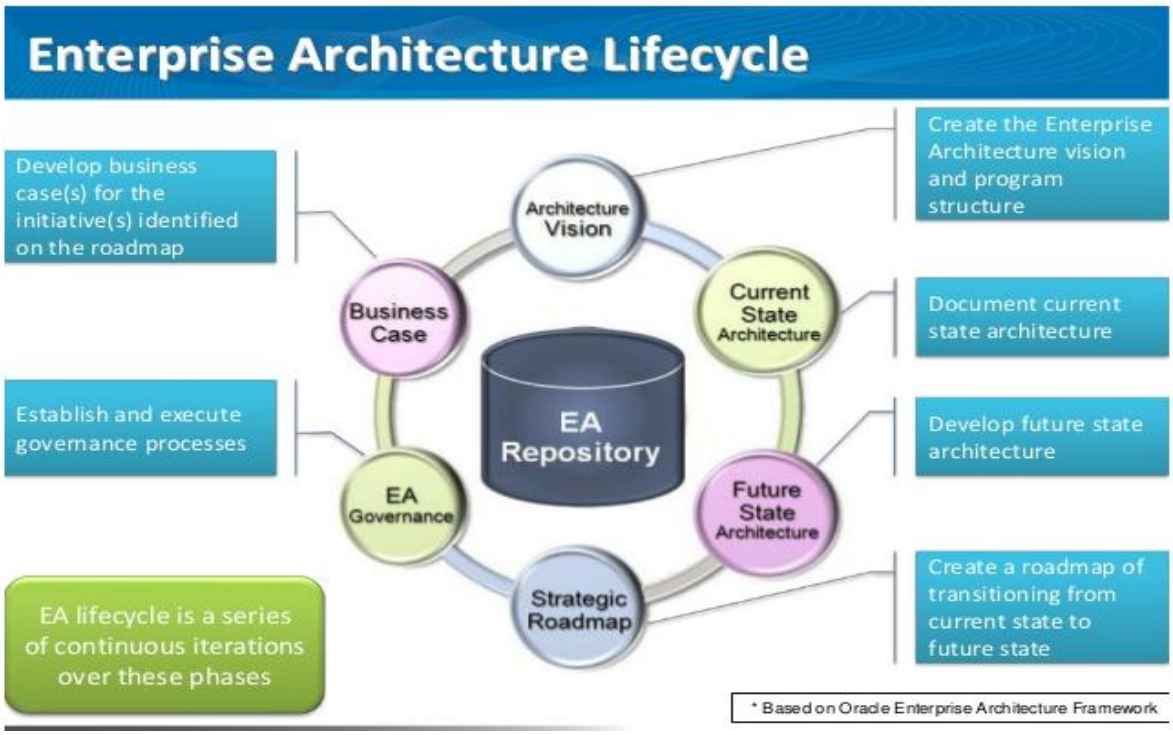
An Enterprise Architecture Life Cycle (EALC) is the basic planning format applied to Enterprise Architecture and strategic planning. Within an EALC, different activities in the project are implemented in a timely and effective manner.

There are several activities involved in Enterprise planning.

- First is to envision and define the scope of the architectural environment.
- Second is to identify key stakeholders.
- Third is to create a business case for systems and finally to create the project and evaluate and maintain it.

The various steps in an EALC are:

- EA Development
- Portfolio Management
- Project Management
- Solution Delivery and
- Organization Change Management.



Enterprise Architect Framework

An EAF is a systematic strategic blueprint that firms can employ to evaluate, construct, and govern the overall structure of an organization's business processes and information technologies. Thus, it serves as the creation of the system for organizational structure, business processes, supporting information systems, and the technology environment for the purpose of their alignment with organizational goals and objectives.

EA Framework Components

An EA framework typically consists of the following components:

- Business architecture identifies and proclaims organizational strategy, management, structure, and principal business activities.
- Information architecture defines the external and internal environment consisting of an organization's data, resources, and mechanisms involved in data management.
- **Application Architecture:** Outlines the plans or architecture of the individual applications to be implemented and how they would relate to one another and the fundamental business processes in an organization.
- Technology architecture summarizes the physical and computer components that are required for corporations to house their most important applications.

Implementing an EA framework involves several steps:

1. **Assessment:** Assess the current status of the organization's architecture.
2. **Planning:** Describe the vision and strategy of the future state.
3. **Design:** Design each component in terms of architecture to the last detail.
4. **Implementation:** Put into use the architecture components as per the designs.
5. **Management:** Sustain updates on the architecture to reflect on the business strategies and objectives in implementation.

Types of EA Frameworks

Several types of EA frameworks are widely used, including: Several types of EA frameworks are widely used, including:

- **The Open Group Architecture Framework (TOGAF)** is one of the most widely known frameworks that offers a holistic viewpoint of how to design, plan, implement, and govern an enterprise information architecture.
- **Zachman Framework:** A structural framework that can be used to categorize data, processes, and technologies in an organization.
- **Federal Enterprise Architecture (FEA):** Created by the US federal government, it aims to provide a common structure for strategic, business, and technology management.
- **Gartner EA Framework:** Emphasizes the strategic alignment of business with IT investments.

Conceptual Layers

The conceptual layers of an enterprise application are the presentation layer, the business logic layer, and the data layer. These layers work together to process user input, perform business logic, and store and retrieve data.

Presentation layer

- The most visible layer, which users interact with directly
- Includes user interfaces, web pages, and mobile app screens
- Focuses on providing a visually appealing user experience

Business logic layer

- Also known as the application layer
- Contains the core business rules and workflows of the system
- Performs data validation, authentication, authorization, and database queries
- Coordinates data between the presentation layer and the data access layer

Data layer

- Stores and manages the application's data
- Provides mechanisms for storing, retrieving, updating, and deleting data
- Ensures data readiness, consistency, integrity, and security.

Four Domains of Enterprise Architecture

Within Enterprise Architecture, there are four primary architecture layers or domains:

- **Business Architecture:** Focuses on the business strategy, governance, organization, and key business processes. It outlines how the business operates, the business processes used, and the interaction between these processes and the various business functions.

- **Application Architecture:** Focuses on specific applications used in the enterprise, how they are designed, and how they interact with each other, users, and other applications.

- **Information Architecture:** Deals with the structure of an organization's logical and physical data assets and data management resources. It concerns how data is collected, stored, transformed, distributed, and consumed. It also deals with data governance and policy.

- **Technology Architecture:** Documents the hardware, software, and network infrastructures used in the business and how they are deployed.

Enterprise Resource Planning (ERP)

- Enterprise resource planning (ERP) is a platform companies use to manage and integrate the essential parts of their businesses. Many ERP software applications are critical to companies because they help them implement resource planning by integrating all the processes needed to run their companies with a single system.

Enterprise resource planning (ERP) is a software system that helps businesses manage their core processes. It can be used to manage accounting, procurement, supply chain, sales, manufacturing, and human resources.

Types of ERP Systems

There's a number of different ERP solutions that can meet a variety of business needs. This list is not meant to list every single type of ERP, though the list is pretty comprehensive. Any business considering implementing an ERP system should be able to find value in some of these types of systems, and multiple systems may be relevant in any given situation.

On-Premise ERP
On-premises ERP systems involve purchasing the software licenses and installing the ERP system directly onto a company's own servers. Companies have full control over the system and data, as it resides within their premises. Customization and integration with existing systems can be more extensive, and this type of ERP usually requires dedicated IT resources for maintenance, updates, and security.

Cloud ERP
Cloud ERP systems are hosted on remote servers and accessed through the internet. Cloud ERPs like SaaS products offer better scalability, allowing businesses to easily adjust resources and features as needed without significant upfront investment in hardware. Cloud ERP systems typically have a subscription-based pricing model, and updates/maintenance are managed by the ERP provider (not the company itself).

Industry-Specific ERP
Industry-specific ERP systems are tailored to meet the unique needs and requirements of particular industries. These systems often include industry-specific modules, functionalities, and best practices to address the complexities of the industry. For example, consider a manufacturing firm that is heavily reliant on inventory, supply

chain management, and distribution of goods. That type of ERP will be vastly different than a client-based ERP such as a financial institution's ERP.

Open-Source ERP

An open-source ERP system (or any open-source software, for that matter) provides users with access to the source code. This means a company can customize, modify, or redistribute the ERP to better meet the company's needs. Implementing and maintaining open-source ERP systems may require more technical expertise and resources compared to commercial ERP solutions.

Small Business ERP

On the other hand, small business ERP systems are designed specifically for the needs of small and medium-sized businesses (SMBs). These types of ERP systems try to offer balance between being slightly niche while offering essential functionalities at a more affordable price point. Because they are less robust, small business ERP solutions are often easier to implement and require less customization compared to enterprise-level ERP systems.

Tiered ERP

Tiered ERP systems offer different levels of functionality and scalability to cater to businesses of varying sizes and complexity. Companies can choose the tier that best matches their current needs and budget, with the option to upgrade or customize as their requirements evolve. This would entail adding on modules as they become relevant (i.e. a company that is scaling to international operations may wait to implement foreign current modules).

Benefits of ERP

- ERP systems can help businesses improve performance and project management.
- ERP systems can help businesses streamline their core business processes.
- ERP systems can help businesses have a unified view of activity and provide a single source of truth.

Improves Accuracy and Productivity

Integrating and automating business processes eliminates redundancies and improves accuracy and productivity. In addition, departments with interconnected processes can synchronize work to achieve faster and better outcomes.

Improves Reporting

Some businesses benefit from enhanced real-time data reporting from a single source system. Accurate and complete reporting help companies adequately plan, budget, forecast, and communicate the state of operations to the organization and interested parties, such as shareholders.

Increases Efficiency

ERPs allow businesses to quickly access needed information for clients, vendors, and business partners. This contributes to improved customer and employee satisfaction, quicker response rates, and increased accuracy rates. In addition, associated costs often decrease as the company operates more efficiently.

Increases Collaboration

Departments are better able to collaborate and share knowledge; a newly synergized workforce can improve productivity and employee satisfaction as employees are better able to see how each functional group contributes to the mission and vision of the company. Also, menial and manual tasks are eliminated, allowing employees to allocate their time to more meaningful work.

CRM (customer relationship management)

Customer relationship management (CRM) is a set of strategies, technologies, and practices that help businesses manage customer interactions and data. The goal of CRM is to improve customer service, sales, and customer retention.

CRM systems compile customer data across different channels and points of contact between the customer and the company. These can include the company's website, telephone, live chat, direct mail, marketing materials and social networks. CRM systems can also give customer-facing staff detailed data on customers' personal information, purchase history, buying preferences and concerns.

CRM benefits

The benefits of CRM systems apply to all types of organizations, ranging from small businesses to large corporations. They include the following:

- **Enhanced customer service.** Having customer information, such as past purchases and interaction history, easily accessible helps customer support representatives provide better and faster customer service.
- **Trend spotting.** Collection of and access to customer data let businesses identify trends and insights about their customers through reporting and visualization features.
- **Automation.** CRM systems can automate menial, but necessary, sales pipeline and customer support tasks.

Components of CRM

At the most basic level, CRM software consolidates customer information and documents it into a single CRM database. This lets business users more easily access and manage that information.

Common components and capabilities of CRM systems include the following:

- **Marketing automation.** CRM tools with marketing automation capabilities automate repetitive tasks to enhance marketing efforts at different touchpoints in the lifecycle for lead generation. For example, as sales prospects come into the system, it might automatically send email marketing content with the goal of turning a sales lead into a full-fledged customer.
- **Sales force automation.** These tools track customer interactions and automate certain business functions of the sales cycle. Sales force automation tools target sales functions where it's necessary to follow leads, obtain new customers and build customer loyalty.
- **Contact center automation.** Designed to reduce tedious aspects of a contact center agent's job, contact center automation includes prerecorded audio that assists in customer problem-solving and information dissemination. Various software tools that integrate with the agent's desktop tools can handle customer requests to cut down the length of calls and streamline customer service processes. Automated contact center tools, such as chatbots, can improve customer user experiences.
- **Geolocation technology, or location-based services.** Some CRM systems include technology that creates geographic marketing campaigns based on customers' physical locations, sometimes integrating with popular location-based Global Positioning System (GPS) apps. Geolocation technology is also used as a networking or contact management tool to find sales prospects based on a location.
- **Workflow automation.** CRM systems help companies optimize business processes by streamlining mundane workloads, enabling employees to focus on high-level and creative tasks that help them close deals.
- **Lead management.** Sales leads can be tracked through a CRM platform, enabling sales teams to input, track and analyze data for leads in one place.
- **Human resources (HR) management.** CRM systems help track employee information, such as contact information, performance reviews and benefits within a company. This enables the HR department to more effectively manage the internal workforce.
- **Analytics.** CRM analytics examines user data to create targeted marketing campaigns that can increase customer satisfaction rates.
- **Artificial intelligence (AI).** AI technologies, such as Salesforce Einstein, have been built into CRM platforms to automate repetitive tasks, identify customer-buying patterns and predict future customer behaviors.
- **Project management.** Some CRM systems include features to help users track client project details, such as objectives, strategic alignment, processes, risk management and progress.

- **Integration with other software.** Many systems integrate with other software, such as call center and enterprise resource planning systems.

Supply Chain Management (SCM)

Supply chain management (SCM) is the monitoring and optimization of the production and distribution of a company's products and services. It seeks to improve and make more efficient all processes involved in turning raw materials and components into final products and getting them to the ultimate customer. Effective SCM can help streamline a company's activities to eliminate waste, maximize customer value, and gain a competitive advantage in the marketplace.

5 Phases of Supply Chain Management (SCM)

A supply chain manager's job is not only about traditional logistics and purchasing. They have to find ways to increase efficiency and keep costs down while also avoiding shortages and preparing for unexpected contingencies. Typically, the SCM process consists of these five phases:

1. Planning

To get the best results from SCM, the process usually begins with planning to match supply with customer and manufacturing demands. Companies must try to predict what their future needs will be and act accordingly. That means taking into account the raw materials or components needed during each stage of manufacturing, equipment capacity and limitations, and staffing needs.

Large businesses often rely on enterprise resource planning (ERP) software to help coordinate the process.

2. Sourcing

Effective SCM processes rely very heavily on strong relationships with suppliers. Sourcing entails working with vendors to supply the materials needed throughout the manufacturing process. Different industries will have different sourcing requirements. In general, SCM sourcing involves ensuring that:

- The raw materials or components meet the manufacturing specifications needed for the production of the goods.
- The prices paid to the vendor are in line with market expectations.
- The vendor has the flexibility to deliver emergency materials due to unforeseen events.
- The vendor has a proven record of delivering goods on time and of good quality.

SCM is especially critical when manufacturers are working with perishable goods.

3. Manufacturing

Using machinery and labor to transform the raw materials or components the company has received from its suppliers into something new is the heart of the supply chain management process. This final product is the ultimate goal of the manufacturing process, though it is not the final stage of SCM.

The manufacturing process may be further divided into sub-tasks such as assembly, testing, inspection, and packaging. During the manufacturing process, companies must be mindful of waste or other factors that may cause deviations from their original plans. For example, if a company is using more raw materials than planned and sourced for due to inadequate employee training, it must rectify the issue or revisit the earlier stages in SCM.

4. Delivery

Once products are made and sales are finalized, a company must get those products into the hands of its customers. A company with effective SCM will have robust logistic capabilities and delivery channels to ensure timely, safe, and inexpensive delivery of its products.

This includes having a backup or diversified distribution methods should one method of transportation temporarily be unusable. For example, how might a company's delivery process be impacted by record snowfall in distribution center areas?

5. Returns

The SCM process concludes with support for the product and customer returns.

The return process is often called reverse logistics, and the company must ensure it has the capabilities to receive returned products and correctly assign refunds for them. Whether a company is conducting a product recall or a customer is simply not satisfied with the product, the transaction with the customer must be remedied.

Returns can also be a valuable form of feedback, helping the company to identify defective or poorly designed products and to make whatever changes are necessary. Without addressing the underlying cause of a customer return, the SCM process will have failed, and returns will likely persist into the future.

Types of Supply Chain Models

Supply chain management does not look the same for all companies. Each business has its own goals, constraints, and strengths that will shape its SCM process. These are some of the models a company can adopt to guide its SCM efforts:

Continuous Flow Model
The continuous flow model relies on a manufacturer producing the same good over and over and expecting customer demand will show little variation. One of the more traditional supply chain methods, this model is often best for mature industries.

Agile Model
The agile model prioritizes flexibility, as a company may have a specific need at any given moment and must be prepared to pivot accordingly. This method works best for companies with unpredictable demand or custom-order products.

Fast Model
This model emphasizes the quick turnover of a product with a short life cycle. Using a fast chain model, a company strives to capitalize on a trend, quickly produce goods, and ensure the product is fully sold before the trend ends.

Flexible Model
The flexible model works best for companies affected by seasonality. Some companies may have much higher demand requirements during peak season and low volume requirements in others. A flexible model of supply chain management ensures that production can easily be ramped up or wound down.

Efficient Model
Companies competing in industries with very tight profit margins may strive to get an advantage by making their supply chain management process the most efficient. That could involve coming up with ways to do a better job of utilizing equipment and machinery, managing inventory, and processing orders.

Custom Model
If any model above doesn't suit a company's needs, it can always apply a custom model. This is often necessary for highly specialized industries with high technical requirements, such as an automobile manufacturer.

Example of Supply Chain Management (SCM)

Understanding the importance of SCM to its business, Walgreens Boots Alliance Inc. decided to transform its supply chain by investing in technology to streamline the entire process. That included using big data, collected from its 9,000 stores and 20,000 suppliers, to help improve its forecasting capabilities and better manage sales and inventory.1 In 2019, it appointed its first-ever chief supply chain officer.2

Walgreens Boots Alliance also incorporated SCM into its environmental, social, and governance (ESG) initiatives. For example, the company began asking suppliers to fill in an online survey that asks questions about their ESG practices, such as whether they have an emissions reduction target in place and the types of materials they use

Human Resource Management

It is a department that helps an organization ensure that its employees work well and feel valued. HRM functions include recruitment, performance management, learning and development, and more.

Human resource management is organising, coordinating, and managing employees within an organisation to accomplish its mission, vision, and goals. This includes recruiting, hiring, training, compensating, retaining, and motivating employees.

HRM staff also develops and enforces policies and procedures to help ensure employee safety. The HRM team manages adherence to federal and state laws that may work to protect employees' private information and ensure their physical safety and mental and emotional well-being. Organisations of varying sizes and industries rely on HRM to keep business running smoothly and efficiently.

5 basic elements of human resource management (HRM)

HRM includes:

- Recruiting new hires
- Evaluating employee performance
- Ensuring fair compensation and benefits
- Training employees and supporting education and development
- Protecting the health and safety of all employees

These are critical cornerstones of the work of HRM professionals. From crafting a job posting to providing continuing education options, HRM functions at all stages of an employee's journey with an organisation.

To be an effective HRM professional, you will need a mix of personal and technical skills like recruitment strategies, creating compensation plans, and communication and team building.

1. Recruitment

An effective recruitment process is at the foundation of HRM. If you can recruit good talent, you can build on their skills and invest in employees for years to come as they add value to the organisation. Equally important is company culture. You want employees who add to the culture of the organisation. Some common recruiting tools HRM may use include job aggregators like Naukri, Indeed or Foundit, video interviewing, or even social media sites like LinkedIn.

2. Evaluation and performance management

HRM uses data to track employee performance to ensure a highly trained and capable workforce. The data compiled can also be used to change staff training methods, implement a merit-based system for raises, and more. HRM professionals use formal measures like performance reviews and informal techniques like interviews or surveys.

3. Compensation

Compensation means salary, commission, benefits, time off, and other non-monetary benefits. HRM uses the industry standard to set salary, commission rates, and benefits. This ensures fairness and allows for a consistent company standard. Some organisations may use performance reviews to adjust an employee's salary.

4. Employee development and learning

Engaged employees are effective employees. HRM understands the importance of a workforce that is challenged but also supported. Most employees want opportunities for advancement and to feel competent and valued in what they bring to an organisation.

Part of HRM is providing employees with these learning opportunities. This might include tuition reimbursement programs, on-the-job training options, conferences, conventions, or certification programs. Aside from individual learning, HRM can also use employee development and knowledge to help employees adapt to organisational changes, such as system upgrades, technology shifts, and new policies.

5. Employee health and safety

The safety and well-being of an organisation's employees are critically important aspects of HRM. Employee health and safety covers a lot, such as safety against harassment, discrimination, or bullying in the workplace. It can mean physical security that would involve building fire code compliance. It can also mean adherence to labour laws that protect an employee's rights in the workplace and cybersecurity or safeguarding an employee's personal information.

A lot goes into protecting all aspects of employees' health and safety, and it is an HRM professional's job to ensure that protection. HRM professionals may do this by installing security cameras, enforcing internet usage rules, implementing a zero-tolerance policy, or creating restricted access areas.

Introduction to web application and its life cycle

A web application is a software program accessible through a web browser, allowing users to interact with services and data online, typically through a

user interface on a website; its life cycle encompasses the entire process from initial planning and design to development, testing, deployment, and ongoing maintenance, ensuring the application functions smoothly and meets user needs throughout its lifespan.

The 7 Stages of The Web Development Life Cycle –

1. Gathering Relevant Information
2. Planning - Sitemap and Wireframe
3. Design & Layout
4. Content Creation
5. Development
6. Testing, Review, and Launch
7. Maintenance and Updation

1) Gathering Relevant Information

The first stage is the most important in understanding the life cycle approach of website development as it involves learning about the client's requirements. Identify their needs and help them by providing the perfect solution. Have a clear idea about the target audience. What the business goals are, and how the website will be utilized to accomplish those objectives?

Here are the considerations:

Purpose:

1. What is the purpose of the website that the developer will be building?
2. What is the actual plan: to provide information solely and sell a product or service?

Objectives:

1. What is the main goal of building the website?
2. Whether it is informational or promotional?

Target Audience:

Is there a specific group of people that will help in accomplishing your objectives?

During the website design and development process, it is important to visualize what type of people you would like to visit the website. Consider their age, gender, likes, and dislikes. This will certainly help in designing the best website according to their interest.

2) Planning - Sitemap and Wireframe

With all the information that has been gathered from stage one, the design and implementation strategies are planned according to the type of website and target audience.

A site map is created in this phase.

For building a full-fledged website having impeccable features and functionalities, it is important to plan wisely. There has to be a detailed list of all the areas of the website and the sub-topics. This is a guide that describes what content will be there on the site, and there is an easy-to-understand navigational system.

Think of the best user interface that is created for easy navigation. The wireframes give an outline of the pages of the website.

Take the help of tools for creating the wireframe. All the planning processes should have the involvement of the client. So, the client has an idea about the blueprint of the project.

3) Design & Layout

Next is to determine the look and feel of the website. In the design phase, it is essential to embed elements like the logo of the company and the colours that help to enhance to the identification of the company on the website.

The designer gives life to the graphics, typography, colours, animations, buttons, drop-down and pop-up menus, and more as per the project requirement.

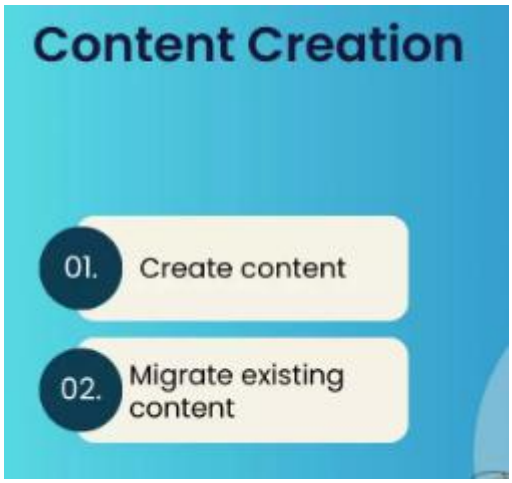
It is the responsibility of the web designer to create one or more than one prototype for the website. This is basically a jpeg image of the final design. Designers give companies access to the workflow so that they can view the progress of the development.

Meeting the demand of the audience with the help of web design is also crucial. If the design is mundane, the website will not be able to capture the user's attention, and this

will increase the bounce rate. The elements should reflect the brand image as well as the business vision.

4) Content Creation

No matter how visually appealing your website may be, effective communication with your customers is indispensable.



The primary objective of content creation is to establish a communication channel through the user interface. It involves presenting relevant information about your company in an engaging and easily comprehensible manner.

This encompasses the inclusion of:

- Compelling calls-to-action
- Crafting creative headlines
- Formatting the content for readability
- Performing line editing
- Updating the texts throughout the entire web development process.

The content creation stage plays a pivotal role in shaping the branding and marketing of your website or web application. It serves as a platform to define the purpose and objectives of your online presence through impactful and persuasive writing.

5) Development

The development is a stage where the website is built while maintaining the essence of the website's purpose. All the graphic elements are taken into consideration and are used to generate a functional website.

The process begins with first developing the home page, followed by the interior pages. The main focus is given to the navigational structure of the site.

Content Management System, interactive contact us forms, and shopping carts are made functional in this development step. The web design & development company suggests changes that are done after consideration.

Do you know that a successful website needs an understanding of front-end website development and back-end development? This means valid HTML/CSS codes are compiled as per the web standards to enhance the functionality for a larger audience.

6) Testing, Review, and Launch

After the completion of web development, it is tested. The functionality is tested along with the device compatibility.

The web designer should be well-versed with the current trends and standards so that the design and development are done accordingly. The technologies used are HTML and CSS. The tester validates the codes written for the website.

When the designer is given the final approval, the website is ready for delivery. The quality assurance team tests for functionality, compatibility, and performance to ensure that the website is ready for launch.

Other testing includes integration testing, stress testing, scalability, load testing, resolution testing, and cross-browser compatibility. Once the green flag is shown, it is deployed to the server using FTP.

7) Maintenance and Updation

The web development stages don't end after deployment. There are many post-deployment tasks that take place. Some elements are changed as per the user's feedback, support, and maintenance.

Technologies Used in The Web Development Process

There are a wide array of technologies, including programming languages, libraries, frameworks, and development tools, that are used to create highly functional websites and applications.

Here we will explore the prominent technologies used in both the front-end and back-end tech stacks:

Front-End Tech Stack:

- **HTML**: Defines web page structure and content with features like semantic elements, media embedding, form validation, and offline storage.
- **CSS**: Controls visual presentation and layout, allowing customization of colours, fonts, spacing, animations, and responsive design.
- **JavaScript**: Enables interactivity, manipulation of HTML and CSS, event handling, animations, and building complex web applications. Popular frameworks like React, Angular, and Vue.js enhance front-end development.

- **Front-End Frameworks and Libraries:** Bootstrap, Foundation, and Material-UI provide pre-built components and stylesheets for rapid development, responsive grids, and consistent UI.

Back-End Tech Stack:

- **Server-Side Languages:** PHP, Python, Ruby, Java, and C# handle data processing, business logic, and database operations. They power CMSs, enterprise-level apps, and APIs.
- **Database Management Systems (DBMS):** MySQL, PostgreSQL, MongoDB, and Microsoft SQL Server efficiently store and retrieve data, ensuring robust management, scalability, and integration.
- **Web Servers:** Apache, Nginx, and Microsoft IIS handle client-server communication, processing requests, and serving web pages for seamless content delivery.
- **API Development:** REST and **GraphQL** facilitate building APIs that enable seamless integration and communication between different systems.