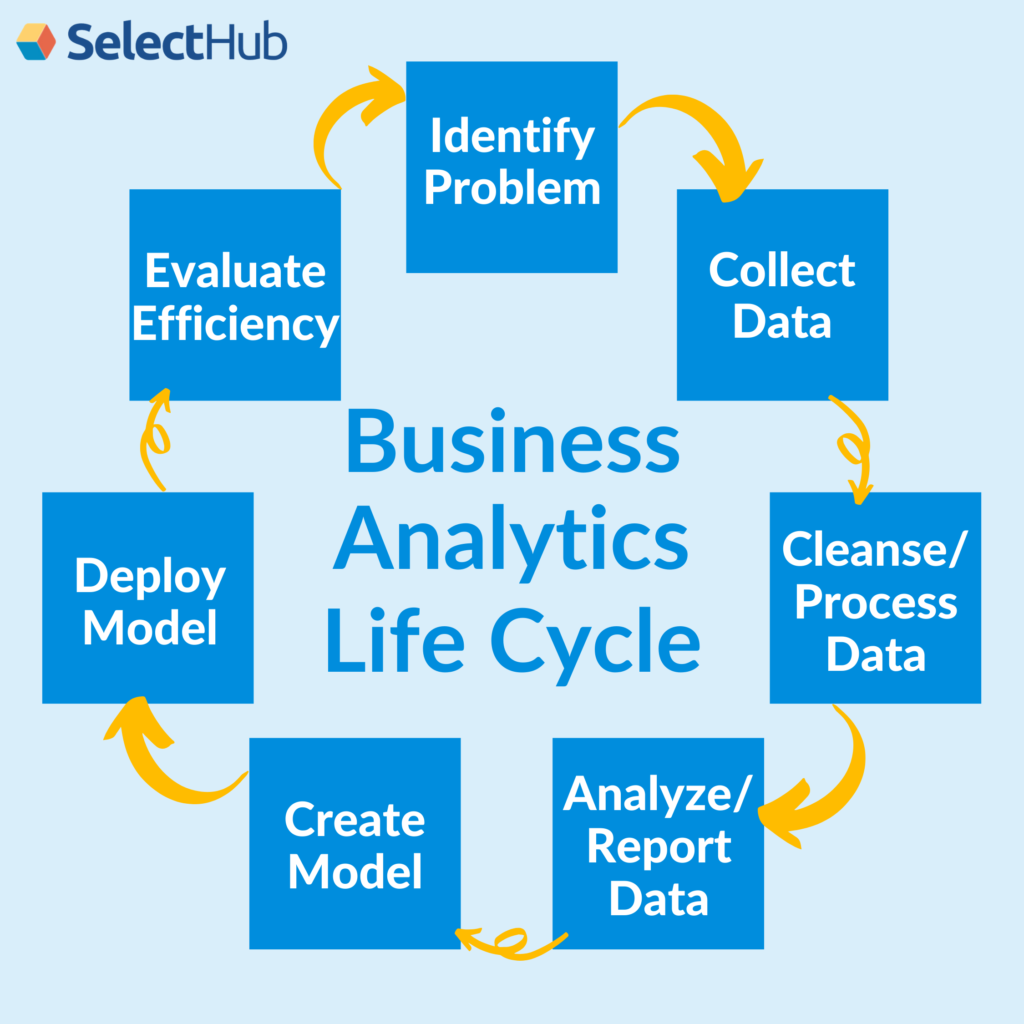
**Name : Sanket Gulhane , DA-14**

* **what is Business Analytics?**

Business Analytics (BA) involves leveraging data, statistical analysis, and modeling techniques to make more informed business decisions. By examining data, organizations can gain valuable insights into their performance and uncover opportunities for improvement. Business analysts, data scientists, and data analysts are typically responsible for carrying out these processes, which aim to optimize decision-making and drive business growth.

One of the key components of Business Analytics is data collection and cleaning. This involves gathering data from various internal and external sources, such as databases, datasets, and market research, and then preparing it for analysis. Data cleaning ensures that the information is accurate and usable, providing a solid foundation for further analysis. Once the data is ready, different types of analytics come into play, including descriptive analytics, which focuses on understanding past events by analyzing historical data to identify trends and patterns. Predictive analytics takes it a step further by using statistical models and machine learning techniques to forecast future outcomes, such as sales trends or customer behavior. Prescriptive analytics, on the other hand, provides actionable recommendations to optimize business performance, often through simulations and optimization algorithms. Additionally, diagnostic analytics helps businesses understand why something happened by exploring the root causes behind specific outcomes.

**Business Analytics life Cycle :**



The roles in Business Analytics can vary depending on the specific area of focus. A **business analyst** is responsible for identifying business problems and providing data-driven solutions, working closely with stakeholders to interpret the data and suggest improvements. A data analyst focuses on data collection, processing, and visualization, using statistical tools to analyze trends and generate reports. With more advanced technical skills, a data scientist works on large datasets and employs machine learning and predictive modeling techniques to derive insights. Business Intelligence (BI) analysts specialize in using analytics tools to create dashboards and reports that track business performance and provide actionable insights. Finally, data engineers design and build the infrastructure to collect, store, and analyze data, ensuring smooth data flow across systems.

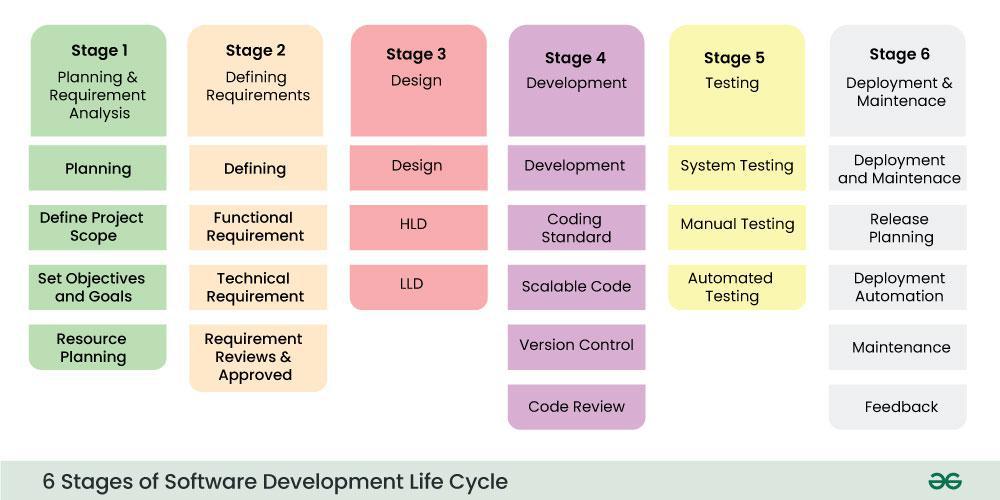
Business Analytics professionals engage in a range of activities. They conduct dataanalysis on both structured and unstructured data to uncover insights that inform decision-making. They are also responsible for reporting, which involves generating clear and concise dashboards, reports, and visualizations to present complex data in an accessible way for non-technical stakeholders. Forecasting is another key responsibility, where historical data is used to predict future trends and behaviors, guiding business strategies. Business analytics professionals also provide decisionsupport, offering actionable recommendations to business leaders based on their findings. Moreover, they work on optimization, identifying inefficiencies in processes and suggesting improvements to reduce costs and increase efficiency. Lastly, they often focus on automation, developing systems to streamline repetitive tasks and improve business operations' overall accuracy and efficiency.

**Tools and Technologies:**

Business analytics professionals use a variety of tools and technologies, including:

* Statistical Analysis Tools: R, Python, SPSS
* Data Visualization: Tableau, Power BI, QlikView
* Databases: SQL, NoSQL databases like MongoDB, Hadoop for big data
* Business Intelligence Tools: SAS, SAP BusinessObjects
* Predictive Modeling & Machine Learning: Python libraries (Scikit-learn, TensorFlow, etc.), Azure Machine Learning, IBM Watson
* **What is software Development Life Cycle SDLC :**

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step. The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user’s requirements. SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users requirements. In this article we will see Software Development Life Cycle (SDLC) in detail.



* Stages Of Software Development Life Cycle

**Stage-1: Planning and Requirement Analysis :**

The Requirement Analysis stage lays the groundwork for a successful software development project by focusing on thorough planning and understanding of project needs. During this phase, the Project Manager and Compliance Specialist collaborate to gather requirements from stakeholders, customer inputs, and market surveys, ensuring all perspectives are considered. Regulatory requirements, such as HIPAA or GDPR compliance, are carefully analyzed to guarantee the secure handling of sensitive healthcare data. Key features, including appointment booking, symptom prediction, and data visualization, are finalized to align with user needs and business goals. The output of this stage provides a clear and focused foundation for the project, serving as a blueprint for subsequent development phases.

**Stage-2: Defining Requirements :**

In this stage, the system's structure and flow are carefully planned. The UI/UX Designer creates wireframes and user journey maps to outline how the user interface will look and feel. Meanwhile, the Back-End Developer designs the database and back-end system to ensure secure and efficient data storage. By the end of this stage, design prototypes and system architecture diagrams are ready, providing a clear guide for the development phase.

**Stage-3: Development:**

This is the stage where the system is built. The Front-End Developer designs responsive web pages for user interactions, such as login screens, dashboards, and forms. The Back-End Developer creates APIs to connect the front-end with the database. Simultaneously, the Data Scientist and ML Engineer develop and train models for features like symptom prediction and health trend analysis. By the end of this stage, a working prototype of the website is ready.

**Stage-4: Integration and Testing :**

All components of the system are integrated and thoroughly tested. The Front-End, Back-End, and ML Models are combined to function as a single system. The QA Tester performs functionality, usability, and performance tests to identify and fix any issues. Compliance testing ensures the platform meets healthcare regulations. This stage ensures the system is reliable, secure, and ready for deployment.

**Stage-5: Deployment :**

The system is deployed to a live environment for users. The DevOps Engineer sets up cloud hosting and CI/CD pipelines to allow seamless updates. Once live, users can access features like symptom checkers, appointment booking, and health analytics. Monitoring tools are implemented to ensure high performance and uptime.

**Stage-6: Maintenance of Products :**

After deployment, the system is continuously monitored and maintained. Bugs reported by users are promptly fixed. The ML Engineer retrains models with new data to enhance accuracy. New features are added based on user feedback, and regular updates are made to ensure compliance with evolving regulations, keeping the system up-to-date and user-friendly.

* **Agile Methodology Sprint in Agile :**

In Agile methodology, a **Sprint** is a time-boxed iteration or cycle in which a team works to complete a set of defined tasks, typically referred to as **user stories**, from the project backlog. Sprints are a core component of Agile frameworks like Scrum and are designed to deliver incremental value through the development of a working product or feature.

A Sprint typically lasts between **1 and 4 weeks**, with **2 weeks** being the most common duration. The duration is fixed and remains consistent throughout the project to ensure regular, predictable delivery. At the beginning of each Sprint, the team conducts a **Sprint Planning** meeting. During this meeting, the **Product Owner** presents the highest-priority items from the product backlog. The **Development Team** discusses and selects which items (user stories, tasks, or features) they can realistically complete during the Sprint, considering their capacity. Additionally, the team defines a **Sprint Goal**, which is a clear, concise objective that provides focus and aligns the team’s efforts.

During the Sprint, the team works on the selected items, often in daily **stand-up meetings** (also called **Daily Scrums**) to assess progress, address any roadblocks, and adjust plans if necessary. The team is responsible for managing its own workflow, and members collaborate to complete the tasks. They may break down the work into smaller tasks, use tools like **Kanban boards** or **task boards**, and apply other Agile practices like **pair programming** or **test-driven development**.



After the Sprint Review, the team holds a **Sprint Retrospective** to reflect on the process and the team’s performance. The team discusses what went well, what didn’t go well, and what improvements can be made in the next Sprint. This reflection is essential for continuous improvement and ensuring that the team evolves in terms of both process and collaboration.

The outcome of a Sprint is typically a **potentially shippable product increment**—a piece of the product that is fully developed, tested, and ready for delivery or release (depending on the project's release strategy). Not every Sprint will result in a fully "shippable" product, but the goal is to have a functional and tested increment at the end of each Sprint.

A key principle of a Sprint is that it is **time-boxed**, meaning the duration is fixed, and no work should extend beyond the Sprint's time frame. Teams are also expected to make a **commitment to the Sprint Goal**, ensuring that the agreed-upon work is delivered in alignment with the Sprint’s objectives. **Transparency** is another key element, meaning that the progress, issues, and work done during the Sprint should be visible to all stakeholders. While teams are encouraged to be **adaptable**, any changes or adjustments to the work should occur at the beginning of the next Sprint, or during the Sprint Review if necessary. **Collaboration** is at the heart of the process, with team members working closely together to achieve the Sprint Goal.

Sprints offer numerous benefits. They enable **frequent delivery of value** by ensuring that the team delivers working software at the end of every Sprint, which provides regular value to customers or stakeholders. The **continuous improvement** inherent in the Sprint Retrospective allows teams to refine their processes over time. Sprints also provide **predictability** since each Sprint has a fixed duration and defined deliverables, making it easier to plan and track progress. Furthermore, the iterative approach allows for **flexibility**, as priorities and the product itself can be adjusted based on feedback after each Sprint.

**Benefits of Sprints:**

* **Frequent Delivery of Value**: By delivering working software at the end of every Sprint, teams can provide frequent value to customers or stakeholders.
* **Continuous Improvement**: Sprints allow for regular reflection, making it easier to improve the team’s processes over time.
* **Predictability**: Sprints allow teams to plan and predict progress, as each Sprint has a defined duration and set of deliverables.
* **Flexibility**: Teams can adjust priorities and refine the product based on feedback after each Sprint, making it easier to respond to changes.
* **Sprint Wise 8 Month Health Care Website Project Plan :**

**Month 1: Planning, Requirement Gathering, and Initial Design**

In the first month, the project kicks off with **Sprint 1** and **Sprint 2**, focusing on planning and initial system design.

* **Sprint 1 (Weeks 1–2)**: The project begins with gathering detailed requirements from stakeholders, customers, and regulatory bodies. During this sprint, the team works on defining the core features of the healthcare platform, such as patient registration, data security measures, and symptom checkers. The UI/UX Designer creates wireframes, and the team establishes project timelines, compliance requirements, and resources.
* **Sprint 2 (Weeks 3–4)**: In this sprint, the system design phase starts. The UI/UX Designer refines wireframes into high-fidelity designs, and the Back-End Developer sets up the database schema and server architecture. Compliance with healthcare regulations is prioritized to ensure the platform adheres to necessary standards (e.g., HIPAA, GDPR).

**Month 2: Front-End Development and Data Preparation**

The second month focuses on front-end development and preparing the data for machine learning models.

* **Sprint 3 (Weeks 5–6)**: The team begins front-end development. The Front-End Developer focuses on creating the landing page, user registration forms, and dashboards. Responsive design principles are applied to ensure the website is accessible on all devices. The UI/UX Designer works closely with the developer to ensure the design is implemented correctly.
* **Sprint 4 (Weeks 7–8)**: During this sprint, the Data Scientist and ML Engineer begin data collection and preprocessing. Healthcare-related datasets are gathered, cleaned, and structured for machine learning tasks such as symptom prediction and health risk scoring. Meanwhile, the Front-End Developer continues refining the user interface.

**Month 3: ML Model Development and Continued Front-End Progress**

In the third month, the focus shifts to building and integrating machine learning models, along with completing front-end features.

* **Sprint 5 (Weeks 9–10)**: The ML Engineer and Data Scientist work on training machine learning models based on the preprocessed data. Models for symptom prediction, patient health monitoring, or risk assessments are built and optimized for accuracy. Meanwhile, the Front-End Developer continues developing interactive features such as user dashboards and data visualizations.
* **Sprint 6 (Weeks 11–12)**: During this sprint, the system integration begins. The Front-End Developer integrates the website’s front-end with back-end APIs, while the ML Engineer deploys the trained models to be used within the website. The website’s functionality, including basic user interactions with health tools, is tested to ensure all components work together.

**Month 4: Testing and Compliance Review**

In the fourth month, testing is the focus. The QA team ensures that the website functions as expected, while compliance is double-checked.

* **Sprint 7 (Weeks 13–14)**: In this sprint, the Quality Assurance (QA) team thoroughly tests the website for bugs, usability issues, and performance problems. Manual and automated testing is done for user flows like symptom checking and appointment booking. Additionally, compliance with healthcare regulations is reviewed by the Compliance Specialist.
* **Sprint 8 (Weeks 15–16)**: During this sprint, the team focuses on resolving bugs from Sprint 7 and further refining the ML models. The website’s security is tested to ensure it meets industry standards for handling sensitive healthcare data. Regression testing is performed to ensure the new features don’t break existing functionality.

**Month 5: Feature Completion and Back-End Development**

In the fifth month, the team works on completing essential features and enhancing the back-end infrastructure.

* **Sprint 9 (Weeks 17–18)**: The team completes any remaining front-end features such as appointment scheduling, patient dashboards, and notifications. The Back-End Developer works on enhancing the server infrastructure, ensuring the back-end can handle the scaling demands of the platform.
* **Sprint 10 (Weeks 19–20)**: The focus shifts to the integration of all the systems with robust back-end processes. This includes setting up cloud hosting environments, improving database query performance, and finalizing the integration of the ML models with the website’s back-end. The system is tested for performance under load to ensure stability.

**Month 6: Finalizing Features and Preparing for Deployment**

In the sixth month, the team focuses on finalizing the system and preparing for deployment.

* **Sprint 11 (Weeks 21–22)**: During this sprint, the final website features are completed. Additional functions like user feedback mechanisms or additional patient tools are implemented. The front-end and back-end are polished, and the system is stress-tested for performance and reliability. The QA team continues testing for edge cases.
* **Sprint 12 (Weeks 23–24)**: The website enters the final stages of development. The deployment process is planned, and the system is prepared for cloud hosting and continuous integration/continuous deployment (CI/CD) pipelines. A staging environment is set up for pre-launch testing.

**Month 7: Pre-Launch Testing and Final Adjustments**

Month seven focuses on pre-launch tasks and final tweaks based on feedback.

* **Sprint 13 (Weeks 25–26)**: The website is tested in the staging environment for bugs, performance issues, and security vulnerabilities. User testing is conducted to gather feedback on the user interface, features, and usability. Any identified issues are addressed.
* **Sprint 14 (Weeks 27–28)**: The deployment process is finalized. Any remaining security vulnerabilities are patched, and the team prepares for the official launch. Training and documentation are provided for ongoing maintenance and updates.

**Month 8: Launch and Post-Launch Monitoring**

The final month is dedicated to launching the platform and ensuring its continued success.

* **Sprint 15 (Weeks 29–30)**: The website is officially deployed to the production environment. The DevOps Engineer sets up continuous monitoring for system performance and security. The team ensures that the website functions smoothly in real-world conditions, and the customer support system is prepared to address issues from users.
* **Sprint 16 (Weeks 31–32)**: After launch, the team focuses on post-launch monitoring, collecting user feedback, and addressing any urgent bugs. The ML models are retrained as needed based on real-world data. Additional features or improvements are identified for future iterations, and the system’s performance is continuously improved.
* **8 months Deployment Timeline(Product Backlog) :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Month 1–2: Project Planning & Foundation | Priority | Planned Days | Used Days | Work Progess |
| **Requirement Gathering :** Analyze business needs, personas, and competitors, focusing on scheduling, symptom checker, and patient dashboard. | High | 15 Days | 12 Days | Completed |
|  |
|  |
| **Tech Stack Selection:** Select front-end (React), back-end (Node.js), database (MongoDB), cloud (AWS), and ML tools (TensorFlow). | Medium | 5 Days | 5 Days | Completed |  |
|  |
|  |
| **UI/UX Design:** Create wireframes, prototypes, and visual designs for key pages, focusing on intuitive interfaces for healthcare users. | Medium | 20 Days | 19 Days | Completed |  |
|  |
|  |
| Month 3–4: Back-End and Front-End Foundation |  |  |  |  |  |
| **Back-End Development (Phase 1):** Set up database for user data, appointments, ML outputs; implement secure authentication. | High | 20 Days | 22 Days | Completed |  |
|  |
| **Front-End Development (Phase 1):** Develop responsive login, registration, and landing pages based on designs. | High | 20 Days | 18 Days | Completed |  |
|  |
| Month 5–6: Machine Learning and Feature Development |  |  |  |  |  |
| **ML Model Development:** Preprocess healthcare data to train ML models. | Medium | 20 Days | 20 Days | Completed |  |
|  |
| **Feature Integration:** Integrate ML models with front-end and back-end | High | 20 Days | 20 Days | Completed |  |
|  |
| Month 7–8: Testing, Compliance, and Deployment |  |  |  |  |  |
| **Quality Assurance & Compliance:** Test functionality , performance, and security; validate ML predictions and ensure regulatory compliance (e.g., HIPAA, GDPR). | High | 23 Days | 23 Days | Completed |  |
|  |
|  |
| **Deployment & Launch:** Deploy on cloud with CI/CD; monitor feedback and resolve critical issues. | High | 21 Days | 21 Days | Completed |  |
|  |