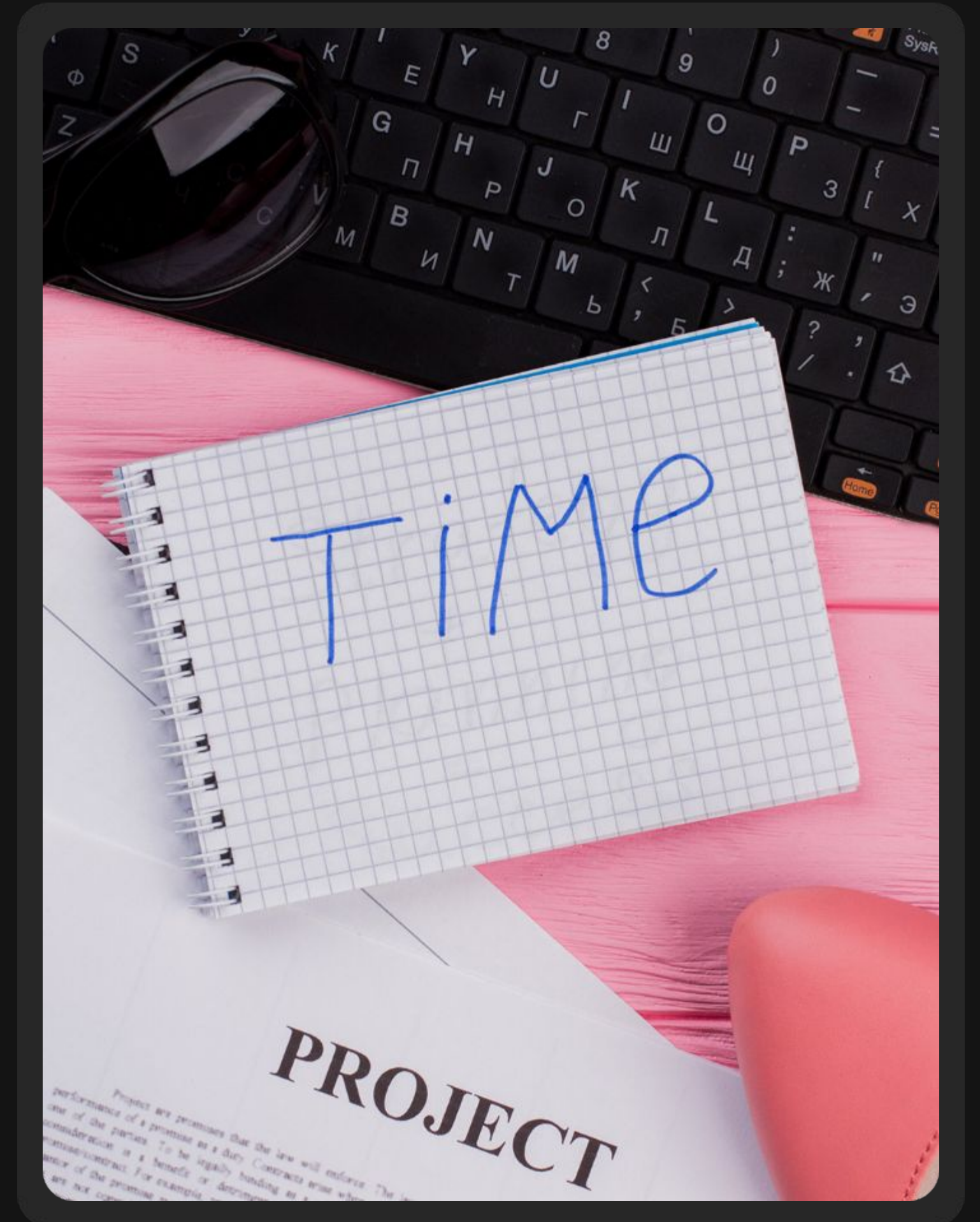


Understanding Software Project Management and the SDLC Process

This presentation provides a comprehensive overview of Software Project Management along with the Software Development Life Cycle (SDLC). We will delve into the core phases of the SDLC, best practices for managing software projects effectively, and explore tools that facilitate each phase to ensure successful software development from initial conception through to deployment and ongoing maintenance.

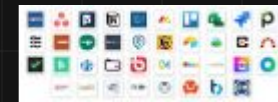


Eddy Lugaye
Instructor



An Overview of Software Project Management and SDLC

Understanding the principles and practices of effective software project management for successful outcomes.



Definition of Software Project Management

Software Project Management involves planning, organizing, assigning responsibilities, and managing resources to achieve software goals effectively.



Achieving Successful Project Completion

The ultimate goal of Software Project Management is to ensure the successful completion of software projects within scope, time, and budget constraints.



Resource Management in Projects

Effective management of resources is crucial for the successful execution of software project management practices.



Importance of SDLC in Projects

The Software Development Life Cycle (SDLC) is essential for delivering software systematically, ensuring quality and efficiency.



Reducing Errors Through Structure

Implementing SDLC reduces errors by providing a structured approach to software development and project execution.



Improving Quality and Control

Adopting SDLC helps in enhancing the quality of software products while maintaining control over time and cost.

Understanding the Software Development Life Cycle (SDLC)

A comprehensive overview of the stages in SDLC and their significance.

Definition of SDLC

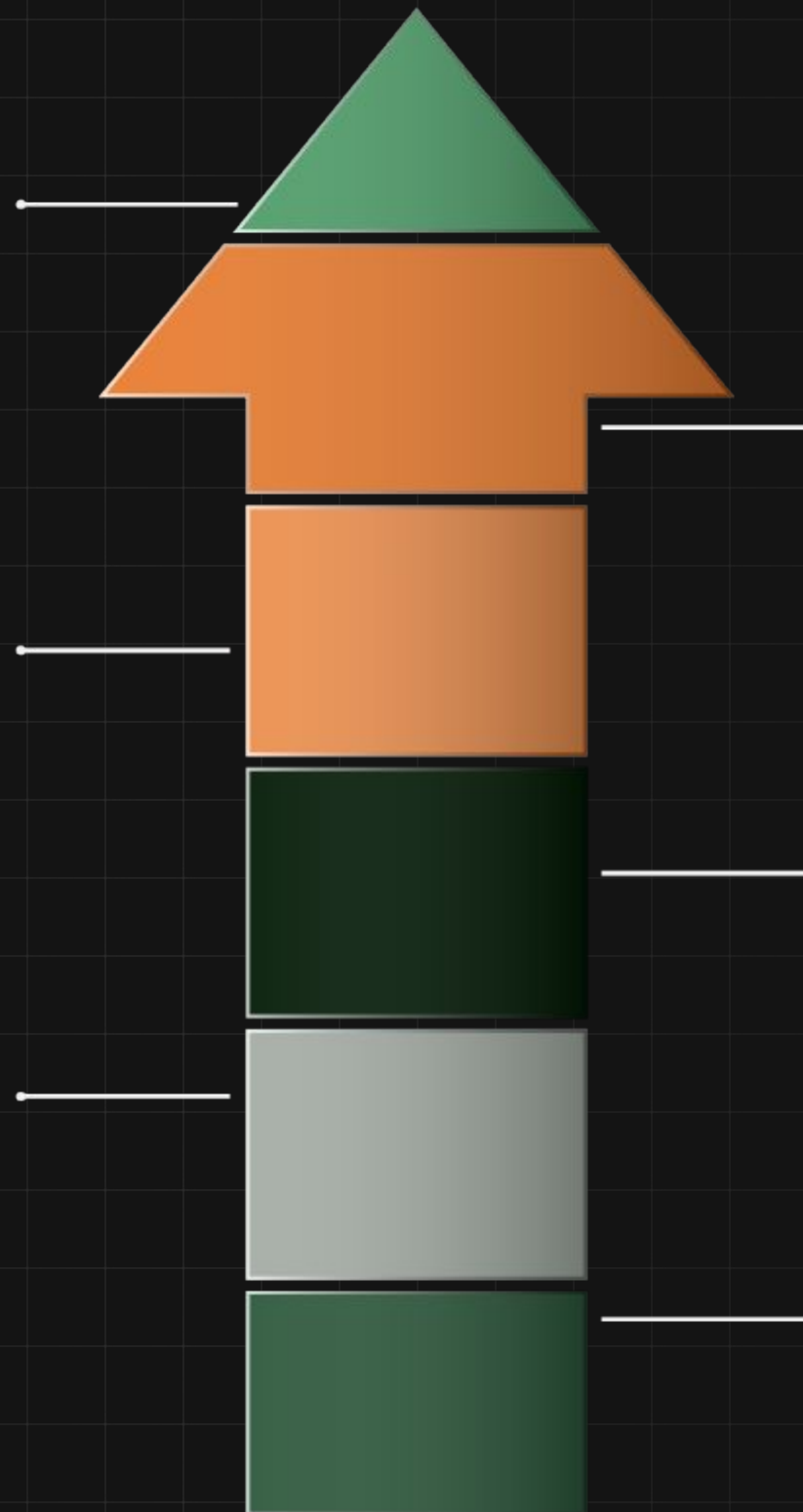
The **Software Development Life Cycle (SDLC)** is a structured approach for developing software applications efficiently and effectively.

Importance of Efficiency

Following the SDLC process ensures that software systems are built efficiently, minimizing waste and maximizing productivity.

Long-term Functionality

Implementing SDLC practices allows for ongoing maintenance and upgrades, ensuring the software remains functional over time.



Stages of Development

SDLC encompasses several stages, from initial requirements gathering to design, development, testing, and deployment.

Meeting User Needs

SDLC helps in identifying user requirements early, ensuring that the final product meets user expectations and needs.

Feedback Loops

The SDLC includes feedback mechanisms to allow for ongoing improvements based on user feedback and performance metrics.

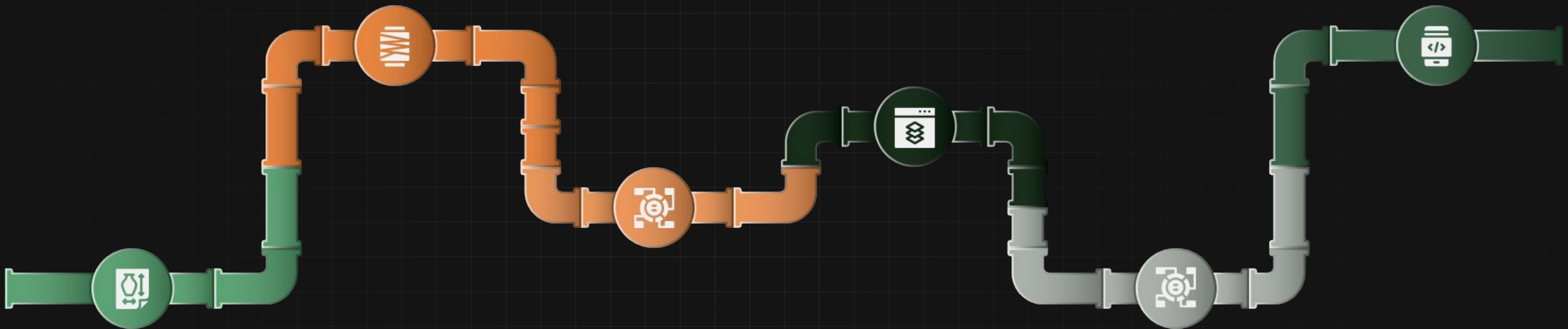
Understanding the Planning Phase in SDLC

Explore the key activities and goals during the planning phase of software development



Understanding the Design Phase of SDLC

A comprehensive overview of the design phase in software development life cycle with key activities and examples.



Purpose of the Design Phase in SDLC.

The main goal is to translate requirements into a detailed blueprint for development, ensuring clarity and direction.

Designing UI/UX Wireframes.

Creating visual layouts for user interfaces to enhance user experience and streamline functionality.

Defining System Architecture.

Establishing the overall structure of the application, including key components and their interactions.

Choosing the Right Tech Stack.

Selecting suitable technologies for frontend, backend, and database to meet project requirements and goals.

Creating Data Flow Diagrams and ERDs.

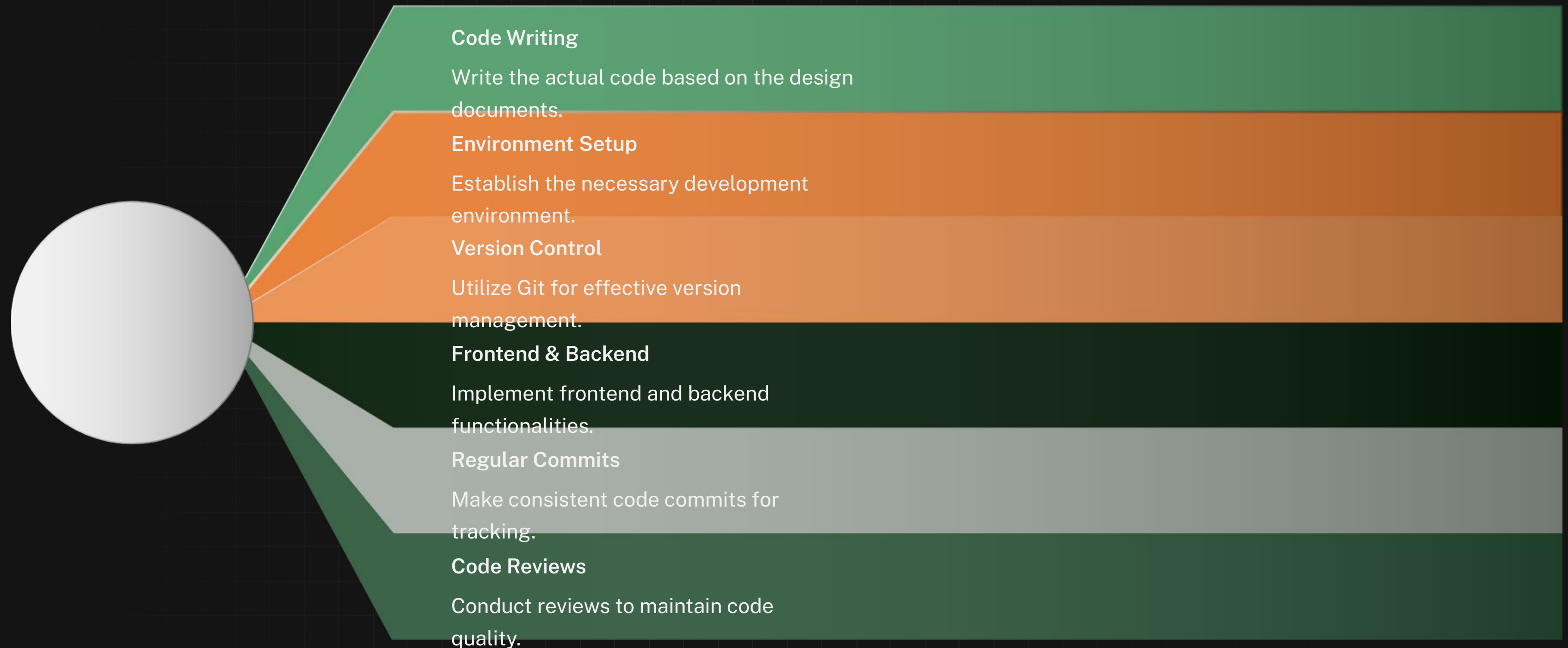
Visualizing how data moves through the system and defining entities and relationships for database design.

Example of Application Interface Design.

Illustrating practical examples like login screens and booking forms to clarify design objectives.

Understanding the Development Phase of SDLC

Key activities involved in coding during the Software Development Life Cycle



Understanding the Software Testing Phase in SDLC

Learn the critical activities involved in software testing to ensure quality and functionality.

Purpose of Testing

Verify software works and is bug-free.

Integration Testing

Testing combined parts of the software.

Bug Tracking

Identifying and fixing software bugs.

Unit Testing

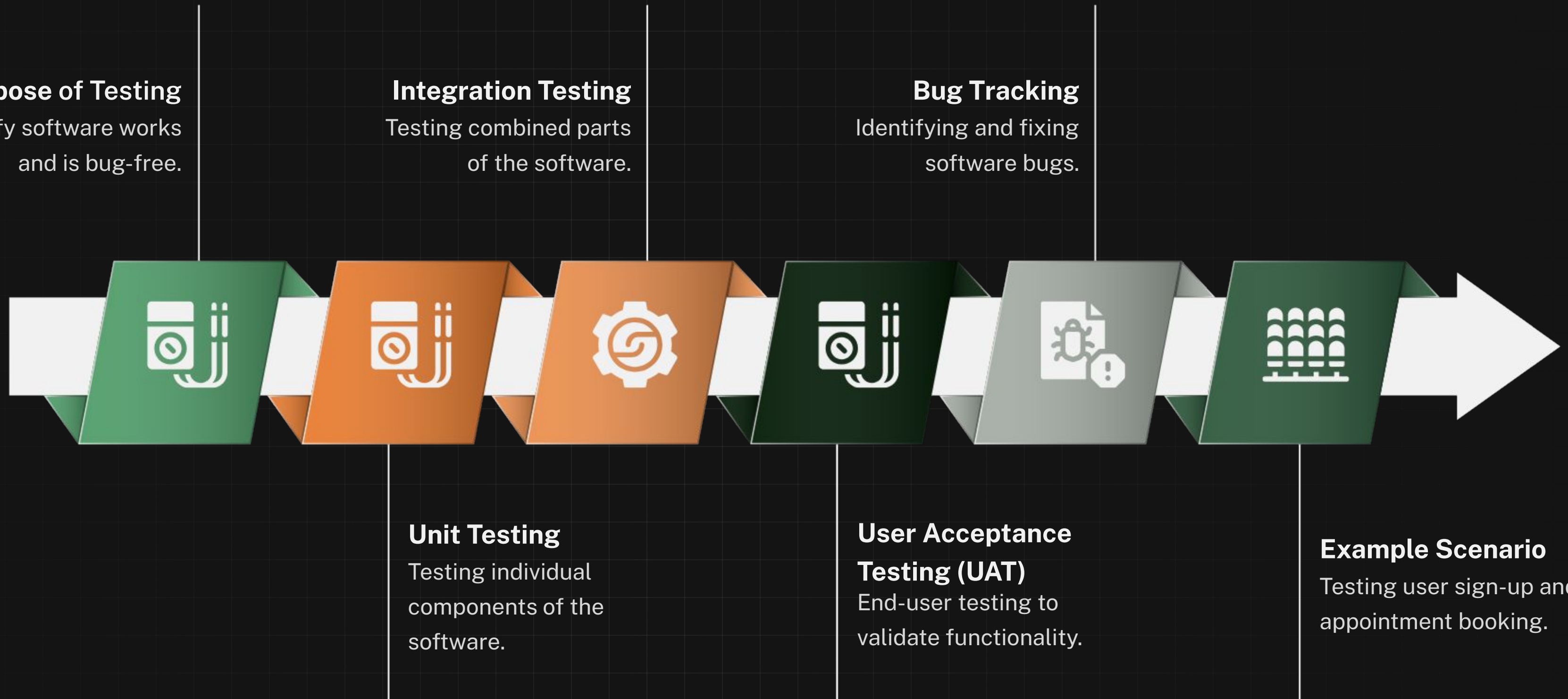
Testing individual components of the software.

User Acceptance Testing (UAT)

End-user testing to validate functionality.

Example Scenario

Testing user sign-up and appointment booking.



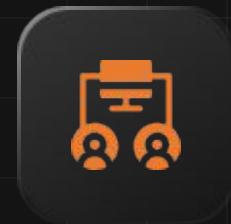
Understanding the Deployment Phase in SDLC

Key activities involved in deploying software to users and stakeholders.



Release to Users

The main purpose of deployment is to release the product to users.



Final Code Push

Execute the final code push to the production server for live use.



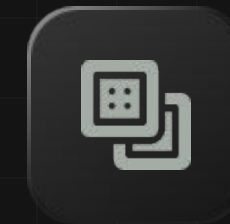
Set Up Hosting

Establish hosting on platforms like AWS, Firebase, or shared hosting.



CI/CD Pipelines

Implement CI/CD pipelines for smooth automated deployment processes.



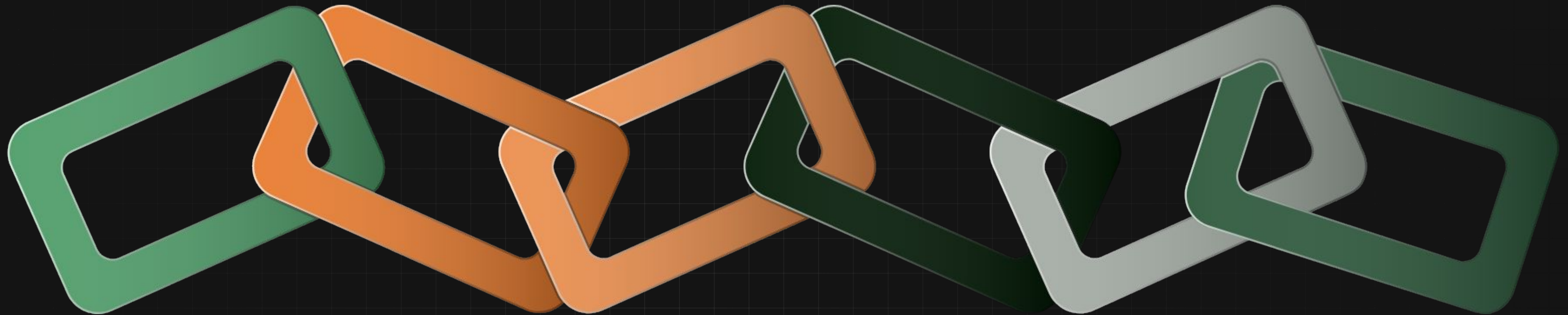
Version Tagging

Create version tags and establish rollback strategies for safety.



Connect to Database

Link the deployed app to a live database for real-time data access.



Understanding the Maintenance Phase of SDLC

Exploring key activities and purposes of software maintenance post-launch



Purpose of Maintenance Phase



Monitoring System Performance



Fixing Bugs and Vulnerabilities



Handling Feature Requests



Updating Libraries and Dependencies

Effective Strategies and Tools for Software Project Management

Explore best practices and tools for successful project execution and collaboration.

Best Practices



Agile or
Scrum



Version
Control



Stand-ups &
Demos



Clear
Documentation



Stakeholder
Involvement

Tool Suggestions



Project
Planning



UI/UX Design



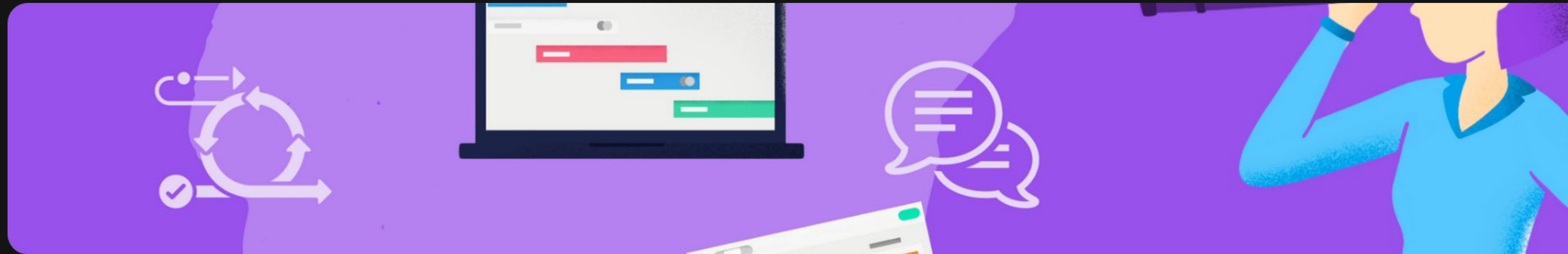
Code
Collaboration



CI/CD &
Deployment



Bug
Tracking



Take the next step towards mastering software project management

Mastering **Software Project Management** and understanding the **SDLC** equips you with the skills to confidently tackle real-world development challenges. Whether you are a developer, project lead, or an aspiring software entrepreneur, embracing these phases ensures the delivery of software solutions that are functional, scalable, and user-friendly.