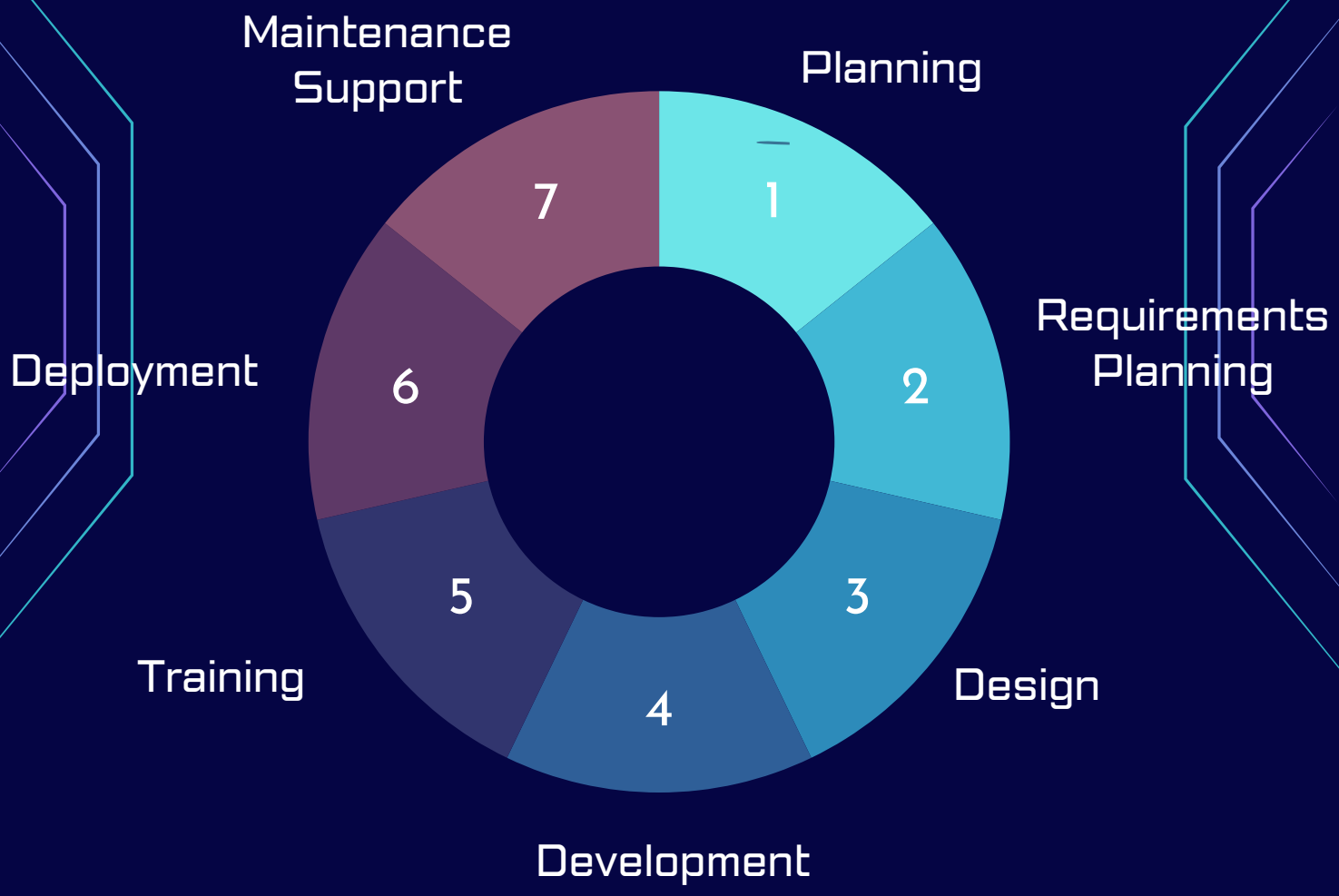
SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

The Software Development Life Cycle (SDLC) is a process used by software developers and project managers to design, develop, test, and deploy high-quality software. Below is a typical SDLC with key phases and their corresponding activities:



SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

The Software Development Life Cycle (SDLC) is a process used by software developers and project managers to design, develop, test, and deploy high-quality software. Below is a typical SDLC with key phases and their corresponding activities:

1. Planning
   - Activities:
     - Define project scope and objectives.
     - Conduct feasibility studies and risk analysis.
     - Identify resources, budget, and schedule.
     - Develop project plans and timelines.

2. Requirements Analysis
   - Activities:
     - Gather and document detailed business and technical requirements.
     - Conduct stakeholder interviews and workshops.
     - Create use cases, user stories, and functional specifications.
     - Prioritise requirements based on business needs.

3. Design
   - Activities:
     - Create high-level and detailed design documents.
     - Design system architecture, data models, and interfaces.
     - Develop user interface designs and prototypes.
     - Review and approve design specifications with stakeholders.

4. Development
   - Activities:
     - Write and implement code based on design specifications.
     - Follow coding standards and best practices.
     - Perform unit testing and peer code reviews.
     - Integrate modules and components.

5. Testing
    - Activities:
        - Develop test plans, test cases, and test scripts.
        - Conduct various types of testing (e.g., functional, integration, performance, security).
        - Report and track defects, and ensure they are fixed.
        - Perform regression testing to validate changes.

6. Deployment
    - Activities:
        - Prepare deployment plans and schedules.
        - Set up production environment and infrastructure.
        - Deploy software to the production environment.
        - Perform post-deployment verification and monitoring.

7. Maintenance and Support
    - Activities:
        - Monitor software performance and address issues.
        - Provide technical support and troubleshooting.
        - Implement updates, patches, and enhancements.
        - Conduct periodic reviews and optimisations.

Additional Considerations

- Documentation: Throughout each phase, maintain comprehensive documentation to ensure clarity and continuity.
- Quality Assurance: Implement quality assurance practices across all phases to ensure the software meets the required standards and specifications.
- Security: Incorporate security best practices and compliance requirements from the planning phase through to deployment and maintenance.
- Agile/Iterative Approach: Consider using Agile or iterative methodologies to allow for flexibility and continuous improvement throughout the development process.

This SDLC framework ensures a structured approach to software development, promoting effective planning, execution, and maintenance of high-quality software solutions.