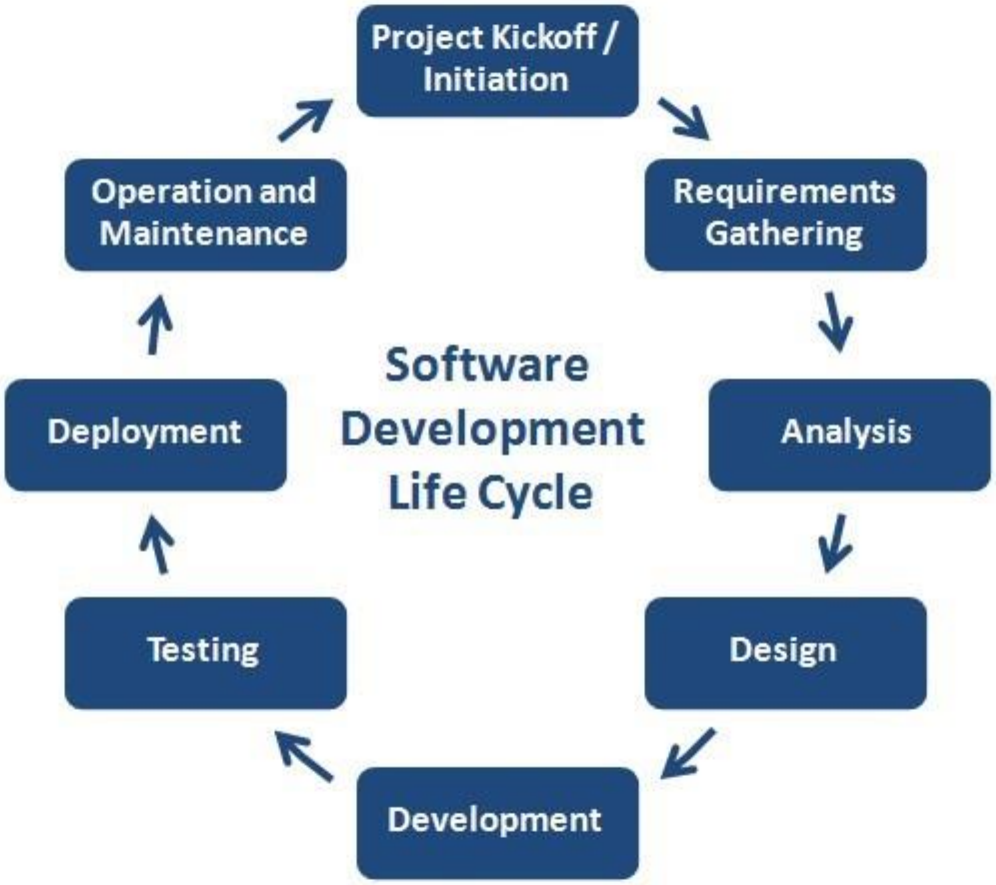
- Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

Software Development Life Cycle (SDLC) Phases


Project Kickoff / Initiation
Requirements Gathering
Analysis
Design
Development
Testing
Deployment
Operation and Maintenance
Software Development Life Cycle

1. Requirements

- **Importance:** Identifying and documenting the needs and expectations of stakeholders.
- **Interconnection:** Foundation of the entire development process, informing subsequent phases.

2. Design

- **Importance:** Translating requirements into a blueprint for development.
- **Interconnection:** Bridges the gap between requirements and implementation, ensuring alignment with stakeholders' expectations.

3. Implementation

- **Importance:** Transforming design into executable code.
- **Interconnection:** Directly influenced by design, ensuring that the developed software aligns with the planned structure and functionalities.

4. Testing

- **Importance:** Evaluating software for defects, bugs, and adherence to requirements.
- **Interconnection:** Validates whether the implemented software meets the specified requirements and design standards, facilitating quality assurance.

5. Deployment

- **Importance:** Releasing the software for its intended users.
- **Interconnection:** Culmination of the SDLC, integrating all previous phases' efforts to deliver a functional product to stakeholders.

Conclusion

The SDLC phases are interconnected stages crucial for delivering high-quality software that meets stakeholder expectations. Each phase builds upon the preceding one, emphasizing collaboration and iteration to ensure successful software development and deployment.