Stages of Building a Software Project

This document outlines the main stages involved in building a software project from idea to deployment. It reflects both academic and industry practices.

1. Requirement Gathering

- Identify the problem the software will solve.
- Talk to users or stakeholders to collect their needs.
- Document functional and non-functional requirements.

Example: "The system must allow users to register and book appointments."

2. Planning & Analysis

- Analyze the requirements to determine the best approach.
- Define the system's scope and features.
- Choose technologies (e.g., C#, SQL, ASP.NET).
- Estimate time and resources.

Tools used: flowcharts, UML diagrams, user stories.

3. System Design

- Break down the system into components (modular design).
- Design the database structure (tables, relationships).
- Create user interface sketches or wireframes.
- Define classes, objects, and how they interact (OOP).

Example: Create a Patient class and a Doctor class with clear relationships.

4. Implementation (Coding)

- Start writing the actual code based on the design.

- Use clean, modular, and well-documented code.

- Organize code into files and folders.

Languages used: C#, with features like classes, inheritance, and methods.

5. Testing

- Test individual components (unit testing).

- Perform integration testing (does everything work together?).

- Fix bugs and refine user experience.

Tools: Console-based testing, test cases, input validation.

6. Deployment

- Make the system available for use (locally or on a server).

- Ensure it runs smoothly in its environment.

- Train users if needed.

7. Maintenance & Updates

- Collect user feedback.

- Fix bugs that appear after deployment.

- Add new features over time.

Conclusion

Following these stages ensures that the software is well-planned, robust, and maintainable.

Even in small student projects, applying this structure improves code quality and professionalism.