

7 PHASES OF THE SOFTWARE DEVELOPMENT LIFE CYCLE

REQUIREMENTS

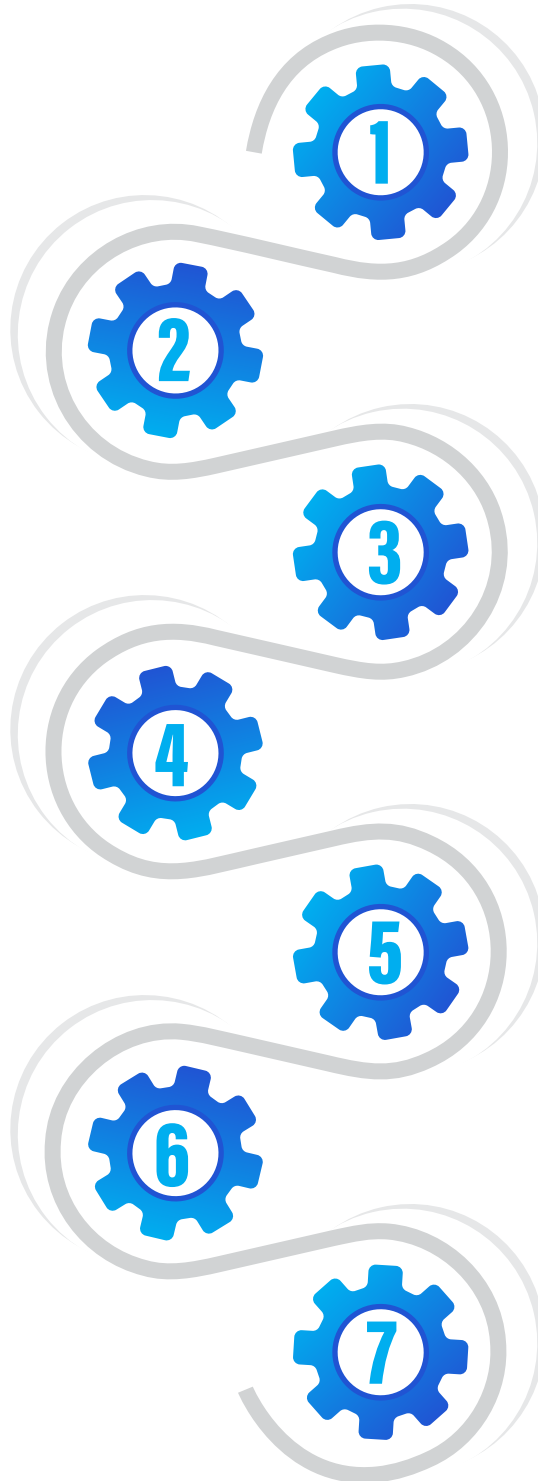
The IT team gathers requirements from business stakeholders and Subject Matter Experts (SME's.) The output of this phase is usually a document that lists these requirements. Agile methods, by contrast, may produce a backlog of tasks to be performed.

SOFTWARE DEVELOPMENT

This phase produces the software under development. This could be in Sprints or Agile, or a single block effort. The output of this phase is testable, functional software.

DEPLOYMENT

The deployment phase is a highly automated phase. In high-maturity enterprises, this phase is almost invisible; software is deployed the instant it is ready. Enterprises with lower maturity, or in some highly regulated industries, the process involves some manual approvals. The output of this phase is the release to production of working software and a final product.



PLANNING

Planning focuses on the scope of the project. The outputs of the planning phase include; project plans, schedules, cost estimation, procurement requirements, and time expectancy.

DESIGN & PROTOTYPING

Once requirements are understood, the design process takes place. It makes use of established patterns for application architecture and software development. Outputs include; design style guides, cascading style sheets (CSS) to begin the process.

TESTING

The testing phase of the SDLC is arguably one of the most important. It is impossible to deliver quality software without testing it first. Methods include; code quality, unit testing, performance testing, and security testing.

OPERATIONS AND MAINTENANCE

The operations and maintenance phase is the "end of the beginning." Though the SDLC doesn't end here. Software must be monitored constantly to ensure proper operation. Bugs and defects discovered in production must be reported and addressed, which often feeds work back into the process.