



The Software Development Cycle

There are many different software development models used by developers to plan and create a software system; in this article we will describe the main stages of the oldest and best-known model – **The Waterfall Model**.

1. Requirements Specification

This is a complete description of the behaviour the piece of software will have. This will be written by your software developer based on your [software brief](#). The developer may choose to use a formal modelling language (such as [UML](#)) to define the requirements specification. It may include various diagrams to help the developer to visualise the various components of the system and their interactions. The types of diagrams they may use include:

- Class Diagrams
- Structure Diagrams
- Behavioural Diagrams
- Interaction Diagrams

The requirements specification may also include a set of [use cases](#) that describe each of the interactions a user may have with the system.

2. Design

The software design is a plan stating how the requirements stated above will be implemented. If the software will not be completely automated then a user interface design may be included. If the software will include a database then the structure of that database should be defined here.

The success of the design stage of a project is vital to its overall success; it is the blueprint for the software. The developers should try to ensure the design is flexible to future changes and unexpected input, fault-tolerant, secure, maintainable and easy to understand.

The testing that will be performed to verify the software may also be specified at this point.

3. Implementation (AKA Coding)

The implementation stage is where the software is actually created by the programmers. Depending on your programmers preference and the type of application you require, a variety of different software languages may be used. User Interfaces and databases are also created at this stage.

4. Testing

Testing allows the developers to perform a variety of checks on the software to assess its quality. The main purpose of testing is to find any bugs in the system that may cause undesirable results. Testing should also attempt to validate and verify the software:

- **Verification** – does the software match the requirements specification?
- **Validation** – is the software what the customer wanted?

Testing can be performed at a variety of levels of detail, including:

- **Unit Testing** – test each basic module (or unit) in the system individually to ensure they perform their task correctly.
- **Integration Testing** – test the interactions between the modules.
- **System Testing** – test the system as a whole against the requirements.
- **System Integration Testing** – test that the system works alongside any third party systems that it is required to work with.

Finally, Acceptance Testing is performed by the customer/end-user to assess whether or not the software achieves their requirements.

5. Installation (AKA Deployment)

Installation is the process of setting up the software so that it is ready to run in a live environment. User training may also be provided at this stage.

6. Maintenance

Maintenance is the process of further developing the software after the initial version has gone live. Maintenance may be performed for a variety of reasons, including:

- Correcting faults
- Improving performance
- Adapting for a new operating environment
- Implementing additional functionality

Maintenance will often involve restarting the software development cycle to specify, design, implement, test and deploy the modifications to the software.

© Copyright 2008 Software-Developer.co.uk