Software development process
Cheat "sheet" for non-technical entrepreneurs

Introduction

In order to build a web or mobile app product, you'll need to learn something about the software development process. This is the process used by technology professionals to build and enhance the software required to run your product.

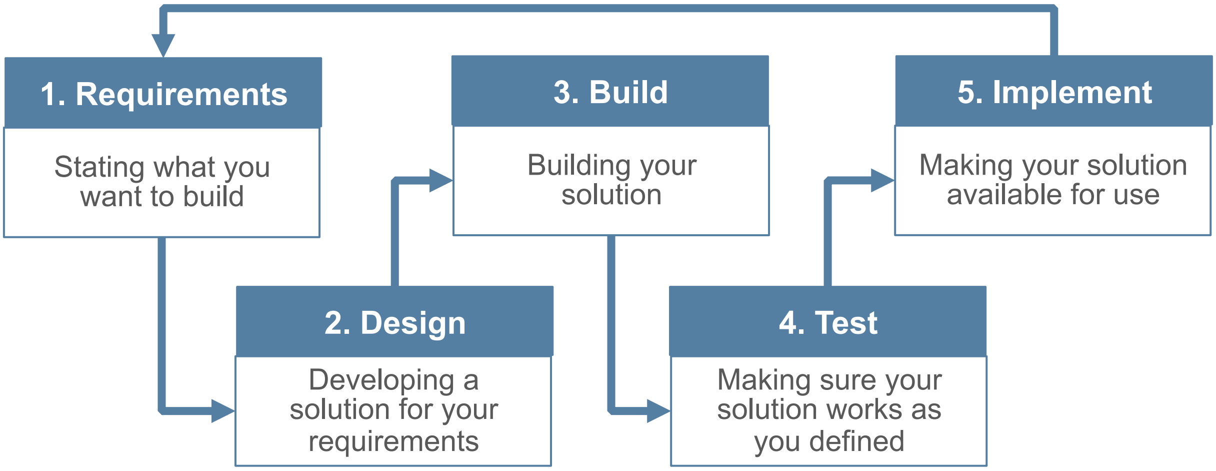
This cheat sheet will:

- Give you an initial overview of the software development process
- Provide some details about the main activities in software development
- Summarise the two main development methodologies
- Highlight the key roles and responsibilities in software development

What is software development?

The software development process is typically known as the "software development life cycle" (or SDLC). It involves a series of activities required to build and enhance software-related products and services. It's a "cycle" because as you finish one piece of development, you go back to the beginning to continue updating the software.

There are five main activities used in software development:


1. Requirements
Stating what you want to build
2. Design
Developing a solution for your requirements
3. Build
Building your solution
4. Test
Making sure your solution works as you defined
5. Implement
Making your solution available for use

A "project" involves undertaking all of these task so that you can implement what is defined in the requirements activity.

Software development process

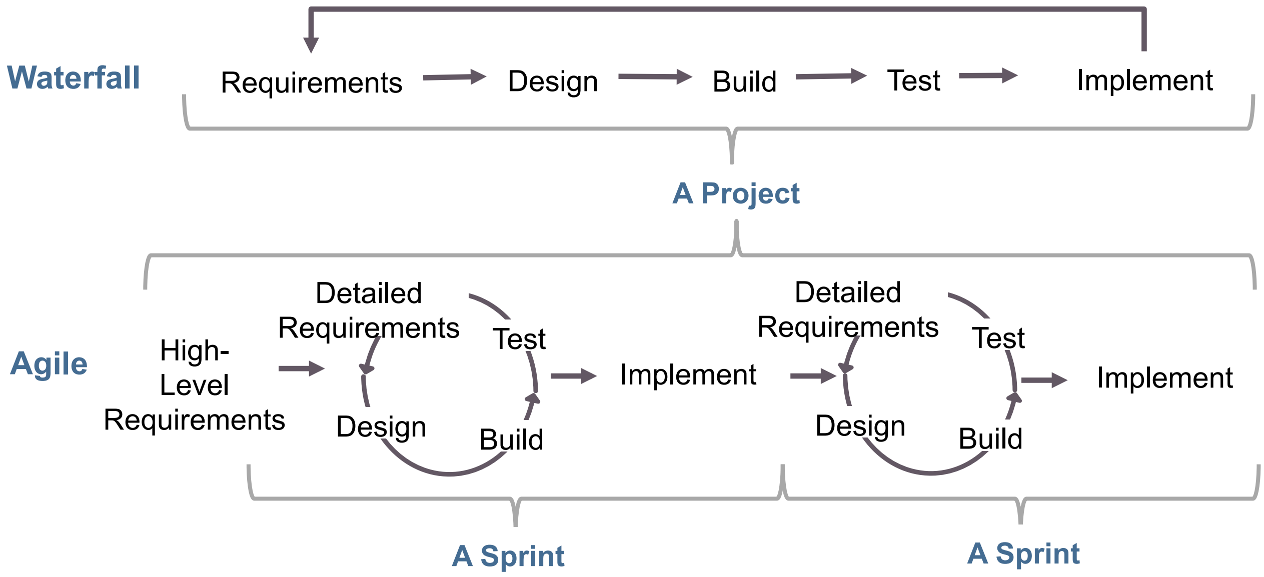
What is a development methodology?

The next concept to understand is the approach or "methodology" used to build software. A "development methodology" outlines how the software development activities are completed.

Today, software is usually built following one of two methodologies – waterfall or agile:

Waterfall	Agile
• Follows a step-by-step process where one step finishes before the next step starts • There is one final implementation at the end - when all of the coding and testing has been completed	• Uses an iterative process where detailed requirements, design, build and test are performed at the same time over multiple cycles of work (called sprints) • Implementations can be accumulated so they may not always "go live" right away

The diagram below outlines the key difference between the two methodologies:



As you can see, the activities are the same in both methodologies, they are just completed in a different way.

While agile is currently a popular approach that focuses on smaller pieces of work that deliver tangible outcomes, waterfall is still a valid approach. It is much more manageable when building software for the first time. Agile principles can be incorporated within the waterfall methodology by defining smaller projects and having a more user-focused approach. Agile requires a solid understanding of software development as the process is intended to be fast-paced; with the team relying heavily on timely responses from each of its member.

Some other definitions

Software development is filled with specialised terms and definitions and each methodology comes with its own terminology. So, before we go any further, here are some key definitions that will help you to understand what your developers are talking about:

Waterfall:

Term	Definition
Business Requirements Document (BRD)	• defines all of the product, operational and system requirements for a project
Software Requirements Specification (SRS)	• defines all of the software development requirements for a specific application
Functional Specifications Document (FSD):	• details the design solution
Technical Specifications Document (TSD):	• outlines the technical components of the solution

Agile:

Term	Definition
User story	• documents a single business requirement for a project
Epic	• an umbrella user story that groups user stories relating to a specific function or process
Backlog	• a prioritised listing of the user stories
Sprint	• cycle of development activities (e.g. detailed requirements, design, build, test, implement)
Scrum	• the team of people responsible for completing a sprint

General:

Term	Definition
System Testing	• activities to ensure that the functionality within the product works according to the design
Integration Testing	• checking the end-to-end functionality of the system; including other applications the product links to
User Acceptance Testing	• Business process and operational testing to ensure that the product works as expected
Production Environment	• “live” system with real people and data
Development and Staging Environment	• copies of the product system where testing and development occurs

Software development process

Software development activities and methodologies

This tables summarises the key activities in the development process and how they relate to waterfall and agile methodologies:

Activity	Waterfall	Agile
Requirements: • *Purpose:* to identify what is to be built • *Inputs:* user feedback, product definition, workflow diagrams, etc • *Processes:* write and prioritise requirements • *Outputs:* requirements document, estimates	• Detailed requirements provided in a BRD • Requirements converted to an system-specific SRS • Project cost estimated based on detailed requirements	• High-level user stories written based on epics • User stories are prioritised into a backlog • High-level estimate provided for the backlog
Design: • *Purpose:* to define a solution • *Inputs:* requirements document • *Processes:* determine a solution based on the requirements, write a design document • *Outputs:* design document	• A detailed FSD is created to provide a solution for all of the requirements for the project	• User stories (starting with the highest priority) are grouped into a sprint based on resource capacity and high-level estimates; additional information on requirements may be provided • Requirements are flushed out during the sprint as team works closely together to design, code and test the solution • User story document is completed with design, build and test information • Other technical documents may also be created • Several sprints may be required to deliver the whole solution
Build: • *Purpose:* to develop the solution • *Inputs:* design document • *Processes:* code the solution, perform basic testing of code components • *Outputs:* technical build document (e.g. list of code components, operational instructions, etc)	• Entire solution is developed at one time after design completed • A TSD is created to document the technical build solution	
Test: • *Purpose:* to check the solution works as designed • *Inputs:* requirements/design documents • *Processes:* define test strategy, test the solution (includes system, integration and user testing) • *Outputs:* completed test document	• A test strategy is created during the build phase to outline how the solution will be tested • Testing starts after build has finished	
Implement: • *Purpose:* to deploy the solution • *Inputs:* implementation plan • *Processes:* implement the solution • *Outputs:* completed implementation plan	• Whole solution is implemented	• Solution from sprint is implemented • Solution from multiple sprints may be combined for a single implementation

Roles in software development

Software development requires many different skillsets. The number of people and the roles that they play will vary depending on the size and complexity of your project. For a start-up, it's likely that you will only have two or three people on your team – you, a developer and maybe a project manager. Sometimes, you will have a development team with multiple developers working on your project but you might only have contact with the lead developer or the project manager.

The table below summarises some of the key technical roles in a large project so you can understand the types of skills that are required for software development:

Role	Activities
Project Manager	• Ensures that the project runs according to cost and schedule by tracking work and managing risk • If you engage a development company, they will likely provide one but they will only focus on software development • You will likely perform the project manager role for any non-technical work
Product Manager	• Defines the requirements for the product – including technical and non-technical components • Validates the solution design • Ultimately responsible for the overall launch and performance of the product
Business Analyst	• Analyses the business requirements to determine what needs to be built in the system. Some requirements may already be supported, so the project will focus on the ones that aren't • Ensures that the final solution addresses the requirements by reviewing designs and even testing the solution
Developer (or Programmer Analyst)	• Documents the technical components of the solution • Codes the solution; senior developers can also be involved in the solution design • Performs unit testing on code components but does not test end-to-end functionality
Visual / User Experience Designer(s)	• Defines the overall look and feel of the application including colour, style and formatting. • Defines the screen layout, user interaction and flow of the product to maximise a user's experience
Solutions Architect / Infrastructure or Network Specialist	• Designs the overall solution for your product • Advises on hardware, communications, platform and language selection

Roles in software development

The below roles are usually required for larger, complex projects that require specialty expertise:

Role	Activities
Database Administrator	• Responsible for the structure and integrity of data and how it is stored within the application
Security	• Responsible for the protection of a network and its applications from cyber attacks, viruses, etc • Manages access to applications; including sign-in policies and procedures
Systems / Integration Tester	• Ensures that the application works according to the design • Typically part of the development team • This role can also be performed by the business analyst or product manager (or product owner – see below)

The below roles are specific to agile:

Role	Activities
Scrum Master	• Similar to a project manager but for the team of people working on a sprint • Manages the tasks for the team; with overall responsibility for the completion of the sprint • Runs a daily standup meeting to determine progress and identify any issues
Product Owner	• Provides business input to the development team; usually the product manager • Provides clarification on business requirements and validates the proposed solution design • Slowly replacing role of business analyst

In smaller projects, people will need to take on multiple roles as all of these tasks still have to be done – even to a small degree. Keep this in mind as you evaluate people's skills and experiences when building a development team.

One final note

So, why do you need to know this?

Building an online web or mobile product can be a daunting prospect for those that have never been involved in developing technical products.

The same applies to any new experience but for some reason, software development can be particularly troublesome.

The gap between what you know and what your developers know can cause huge frustration – but more importantly, it can delay your project and cost you money.

By learning about the software development process and understanding the expectations of your developers, you will be able to give your developers the information they need to successfully build your product.

Software development is a vast and complex topic that could fill a lot of pages. We've tried to distill it into a short, digestible format specifically for non-technical entrepreneurs. However, if you still have questions about the software development process, them please feel free to email us at info@greatproductsconsulting.com.