# Product development

## What is product development?

Product development can be defined as the creation of entirely new products or the modification/updating of existing products. There are a number of ways in which we can develop a new product. Here are three ways.

* Ad-hoc, often described as build and fix. There is no development process and this way often produces products of poor quality.
* Hacking, can either be seen as getting the job done but without consideration of elegance or efficiency, or it can be seen as producing a solution using great skill?
* Development methodologies break a problem down into a series of tasks, provide focus and structure, introduces product planning and timekeeping, and leads to better programming (Knott and Dawson, 1999 - find the full reference)

## What elements might there be in a product development methodology?

If we accept that a product methodology is going to break the development down into a series of tasks we can consider the types of tasks that might need to be undertaken and if we can generate a structure to these tasks. The following structure is one approach.

* Requirements/Specification: These are usually determined through discussion with the customer. In the case of your project you have been given a very simple requirement which your team must use to create a more detailed requirement/specification.
* Design solution: There are potentially many design solutions that would meet a particular set of requirements. In some cases different companies offer design solutions in competition and a customer chooses one.
* Implementation: A design solution has to be turned into the actual product. This is the implementation stage. In some cases 'build' is used in place of implementation and implementation is reserved for the final acceptance of the product.
* Testing: A product must be able to be tested to demonstrate it meets agreed requirements/specifications. It is important to consider testing at an early stage to avoid implementing a design solution that cannot be tested to demonstrate it meets the requirements/specifications.

## The classical Waterfall methodology

In the classical waterfall, or sometimes called the traditional, methodology the project starts with determining the requirements/specification. This is followed by the design stage. At this point there may or may not be feedback to the requirements/specification stage as the design stage may reveal aspects of the specification that need changing. Once the design stage is complete the implementation stage begins. Once again there may be feedback to the design stage and even to the specification stage. Finally the implemented product is tested to demonstrate it meets the specification. As you probably have guessed there may or may not be feedback to the implementation stage. Feedback requiring changes to the specification as this stage will potentially prove costly.

### Disadvantages with the classical waterfall methodology

It has always been the case, but it is true more and more in recent times, that problems are often dynamic in that they change with time. In the waterfall methodology the solution that is provided at the end of the process may no longer solve the current problem such that your solution is obsolete. Furthermore customers understanding of their requirements are often unclear which will lead to a poorly defined specification.

### When to use the classical waterfall methodology

The methodology should be used when the goals and solutions are clear. This will typically mean that the product development timescale should be short (6 months - 1 year)

<https://youtu.be/P79kEyNTGjs>

### Project management

N week 8 project management will be discussed in detail. For now it is useful to know that in the waterfall methodology the clear path that is being followed from specification to tested implementation is usually overseen by a project manager. The project plan is usually set out at the beginning using tools such as Microsoft Project (Gantt chart, resource management, deliverables, and milestones). Project reviews are held and changes may occur but major changes are not expected. The following video provides a basic introduction to Microsoft Project which is a popular tool for project planning.

<https://youtu.be/sSS1tu1yQ-Q>

### Roles

N the waterfall methodology the roles of individuals are usually defined. Typical roles in a software project are;

* Project leader
* Project manager
* Business analyst
* Programmer/ Tester

Each person undertakes tasks specific to their role and interacts with other members of the project team when and if necessary. It may be possible that some project team members never meat as their roles do not require it.

## Agile methodology

Agile methodology is becoming more widely used in software development and even in hardware development. It is an iterative, incremental and adaptive methodology that discovers solutions along the way. It is therefore suitable for product development when the goals may be clear but the solutions very unclear.

[**Agile Manifesto**](http://agilemanifesto.org/)

The Agile manifesto states the following.

"We are uncovering better ways of developing software by doing it and helping others does it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiations

Responding to change over following a plan."

<https://youtu.be/rf8Gi2RLKWQ>

### Scrum

The focus on the scrum framework is on flexibility and adaptability. The team typically work face to face and the customer remains involved if not part of the team for the project duration. Customers can change their minds through a "requirements churn". The problem being addressed is often not fully understood. Finally the scrum method is often facilitated by the use of tools either in-house or commercial.

#### Roles

There are three main roles within the Scrum framework. These are the product owner, the scrum team, and the scrum master.

The product owner should have a vision of what is being built through a prioritised feature list (the product backlog). The product owner does not set the workload. The team members themselves decide what workload (user stories) they commit to within a sprint. No new requirements are set within a sprint, only outside sprints. The product owner must have good communication skills, be business wise, and available.

The Scrum team does not consist of team members with traditional roles such as programmer or tester. All scrum team members’ work together to complete the work agreed for the current sprint. Due to this there is usually a strong bonding within a scrum team. Teams are usually 5-9 people. For large projects there will be teams of teams and scrum of scrum meetings.

The scrum master ensures the team functions using the values and practices of scrum, and helps the team to work effectively by removing obstacles, convening meetings, and working with the product owner to maintain the product backlog. The scrum master protects the team from over commitment or complacency. The authority of the scrum master is granted to them by the team.

#### Sprints

A sprint lasts for 1 week to a month. Within a sprint the scrum team takes a small set of feature ideas, produces code, and tests the code. The sprint is made up of the following components.

A sprint planning meeting where a list of items from the product backlog are agreed to be the focused.

 A daily scrum meeting of usually lasts no more than 15 minutes. At this meeting team members share what they worked on yesterday, what they will work on today, and any problems that might have or will arise.

A sprint review at which the new functionality added during the sprint is presented. Feedback is received from the product owner or any other users/stakeholders invited to the review meeting. The result may be required changes or the addition of new items to the product backlog.

A sprint retrospective to reflect on the sprint and identify improvements for the next sprint.

## What methodology to use for the CE101 Team Project?

You should now be able to think about what methodology you will use and you might conclude that you wish to use agile methodology. However, in order for you to have a good grounding in traditional methodology we will be using waterfall methodology. As a team you will provide a specification, develop and design solution, create your code through an implementation stage, and finally test your software to demonstrate it meets the specification. The complete sets of stages are therefore;

**The requirements/specification**: A customer will have an idea of what is required of the product you are being asked to develop. Sometimes this is a very clear idea and with technical detail. Sometimes your customer may not be highly technical and the description of what is required is stated in a very non-technical manner. This step in product development is concerned with capturing the customer requirements and formulating them into a specification. There is a standard for a software requirement specification that is produced and updated by the IEEE which will be discussed later in this chapter. In your team you will develop a specification for your product using this model. Two key elements of your specification are the users and the functional requirements. It is important to consider all the possible users and then to determine the functional requirements of each user. For example if a product has a display screen and users who will interact with this screen through gesture recognition you might like to consider if the users are all of the same type and what ways they might interact with the screen. If there are several user types with different interaction needs you will need to specify different functional requirements for each user type. As part of your product development research you might search for articles on functional and non-functional requirements.

**The design solution**: Now imagine you have a product specification. The next stage in the waterfall methodology is to develop a design solution. This is not the ‘building’ phase but just the ‘idea’ stage, although you might undertake prototyping in order to develop your design solution. There are potentially many design solutions for one specification. The choice of design solution will depend on factors such as the ability of those who will actually implement the design solution, the availability of material, the cost, the time allowed etc. One important aspect of the design solution is that it must meet the requirement specification and that suitable tests must demonstrate that the design solution meets the specification. It is therefore important to produce a design solution that can be tested! In the case of your team product you will need to take into account the skills of each team member regarding implementation.

The implementation: The next step is to implement the design solution. In the case of your team product this means actually writing the code. As all team members must be involved in the product development it is recommended that each team member actually develops some part of the code, taking into account that some team members are the lead on the product development and therefore will take a larger role in this activity. How you divide the code writing up between team members is for the team to decide. The team will also need to decide how to bring all the code written by different team members together so that it works. This will require good team discipline and teamwork! A clear design solution will help to make the implementation a smooth step.  Although the next step in the waterfall methodology is testing it is more than likely that you will implement and test in small steps, more like an agile approach. If you use an agile approach at the implementation stage your overall methodology is called “hybrid”, which means a mixture of Waterfall and Agile methodologies.

**Testing**: In Waterfall methodology you test at the end. Some of the reasons for the development of the agile approach were to prevent this step being squeezed into a small time slot and therefore not done properly. If you try to only test your software after fully implementing it you are likely to have insufficient time to fully test your software. You have been warned! As part of your research into product development you should search for articles on testing and in particular search for static and dynamic testing, unit testing, integration testing, system testing, acceptance testing, regression testing, stress testing, white and black box testing. For your product development you might want to establish a test plan at the design solution stage.

**Deployment and maintenance**: All products will eventually be deployed to the customer unless some major disaster occurs. Software deployment can be by many means and these days are often through Internet download. Once the customer receives the product successfully it has been deployed. There may of course be deployment problems. What might these be? Assuming a successful deployment the next matter to consider is the maintenance. How will you support your customer with maintenance? As part of maintenance you might have to consider sustainability. Have you designed a sustainable product? Sustainability is one aspect of professional issues that are covered in the context sub-chapter of chapter 3. Other professional issues are ethical, legal, and health & safety. How are these relevant to your product?

## First steps for your team product

Your customers (Klaus McDonald-Maier and Vishuu Mohan) asked you to write a python code module for the Nao robot to demonstrate an aspect or aspects of data gathering, analysis, and reporting. The module should be included in a performance the robot will give to prospective new students. Each team will therefore supply a performance as Choregraphe behaviour for the Nao robot, with the python module embedded in the behaviour. The customer requires the software to be written in Python 2.7. There is no specific requirement regarding where the data is gathered but the customer suggests that this might be from excel workbooks, csv files on the internet, or web pages. Regarding data analysis, again the customer has no specific requirement but suggest that data analysis might be undertaken using python code, formulae in excel, or other mathematical analysis tools. The customer has a preference for statistical data analysis but you can demonstrate other forms of data analysis if you prefer. Data reporting must be by the robot but it can be information that was made available elsewhere.

Once you have begun to develop ideas about your product you can seek further information from me. Your first task is therefore to arrive at a comprehensive specification statement for your product. In the next section I will introduce the IEEE specifications standard. You will use this standard model to create your specification. You might like to consider using some of the following Python modules. You can start by following the links provided and reading about each of these modules. All the modules are installed in the CSEE labs.

[Wincom32 use with excel](http://pythonexcels.com/python-excel-mini-cookbook/) (import win32com.client)

[Urllib](http://docs.python.org/3.0/library/urllib.request.html) (import urllib.request)

[Csv file manipulation](http://docs.python.org/2/library/csv.html) (import csv)

[Matplotlib](http://matplotlib.org/api/pyplot_api.html) (import matplotlib.pyplot and import numpy)

[BeautifulSoup](http://www.crummy.com/software/BeautifulSoup/) (from bs4 import BeautifulSoup)

[Requests](http://www.python-requests.org/en/latest/) (import requests)

[ystockquote](https://github.com/cgoldberg/ystockquote) (import ystoockquote)

Example programmes will be provided for you for the Week 7 Lab to give your ideas regarding your team product.

[Example code now available](https://moodle.essex.ac.uk/mod/folder/view.php?id=333417) - Run in IDLE (Python GUI) 2.7.

[The IEEE Specification standard](https://moodle.essex.ac.uk/pluginfile.php/601021/mod_book/chapter/6822/IEEE%20Specification%20Standard.pdf)

The IEEE specification standard can be accessed via the link above. It is also provided in the [chapter 3 initial reference material folder](https://moodle.essex.ac.uk/mod/folder/view.php?id=333420).

Abstract: The content and qualities of a good software requirements specification (SRS) are described and several sample SRS outlines are presented. This recommended practice is aimed at specifying requirements of software to be developed but also can be applied to assist in the selection of in-house and commercial software products. Guidelines for compliance with IEEE/EIA 12207.1-1997 are also provided.

Keywords: contract, customer, prototyping, software requirements specification, supplier, system requirements specifications

### **An SRS outline**

#### **Introduction**

Purpose: Provide a short statement of the purpose of the SRS and state the intended audience.

Scope: Identify the software by name. Explain what the software will do. Describe the application of the software.

Definitions, acronyms, and abbreviations: Give any definitions required to read the SRS and provide any acronyms and abbreviations used in the SRS.

References: Provide a list of references used in the SRS.

Overview: Provide an overview of the SRS and explain how it is organised.

#### **Overall description**

Product perspective: You should describe the overall system clearly stating if it is an independent system or part of a larger system. A block diagram of the product may be useful.

Product function: Provide a summary of the functions the software will perform. Organise the functions such that they are clear and understandable by the identified user(s).

User characteristics: Describe the characteristics of the users of the product indicating for example the required education level, experience, or technical knowledge. Be sure to include all the user types.

Constraints: List any constraints such as hardware limitations, safety and security.

Assumptions: List assumptions such as operating system requirements on the hardware to be used for the software product.

Apportioning of requirements: Provide detail regarding any delay of specific requirements to later versions of the software.

#### **Specific requirements**

External interface requirements: User interfaces, hardware interfaces, software interfaces, communications interfaces.

Functional: State all the fundamental actions of the software product being developed. Provide information on the user type to whom the functional requirement applies. For example "the customer" will be able to download the software from the app store, and "the administrator" will be able to create an account. List all the functional requirements providing sufficient detail to describe the function. Review the SRS document from the University of Gothenburg, provided in the chapter 3 initial reference material folder. Research SRS functional requirements.

Non-functional: These can be performance indicators such as the capacity, response time, or fault recovery time. They could be design constraints such as the hardware to be used or the memory requirements. They could be software system attributes such as the adaptability, availability, reliability, usability, or security.