Stages of Software Development Process

Software Development Process

Figure 1: An iterative design cycle. Source: DES/NCCA (2018) *Leaving Certificate Computer Science Curriculum Specification.*

1. Investigate - Define the Problem

Identify and fully analyse the problem

- Who the users are.
- How many groups of users exist.
- The needs of each user group.
- The type of environment that the users work in.
- Who is commissioning the proposed software (the client) and why.
- What the client expects of the new system.
- The scope of the proposed system (what it can reasonably be expected to do).
- Constraints, such as time, budget, staff etc.

Investigate - Define the Problem

Gather information - interviews, questionnaires, observations, samples of existing documentation

Requirements Specification

- **User requirements**
- **Client requirements**
- **Hardware/Software requirements**

Investigate - Define the Problem

Carry out a **Feasibility study** - report written for client detailing whether a suitable solution is possible

The study sets out:

- Options for the solution including the pros and cons of each.
- Estimates of resources required.
- Timescale for completion.
- Likely implications for changes to business processes and staffing.
- The need for training.
- Benefits of the proposed solution.
- Drawbacks of the proposed solution.
- How the system will need to be maintained.
- A recommendation as to whether the project is feasible or not and if it is, how to proceed.

2. Plan - Understand the Problem

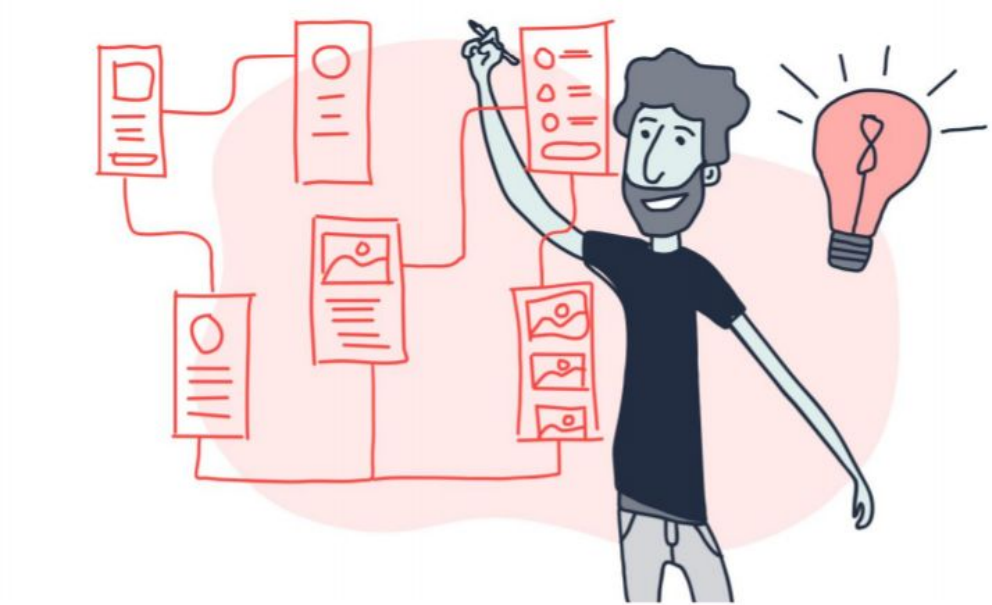
A plan is necessary to ensure that everyone knows **what** must happen, **when** and by **whom.**

The plan should cover the various stages of the project broken into tasks.

Plan - Understand the Problem

- Timescale
- Dependencies
- Hardware and Software requirements
- Staffing

3. Design - Create a representation, decide on tools

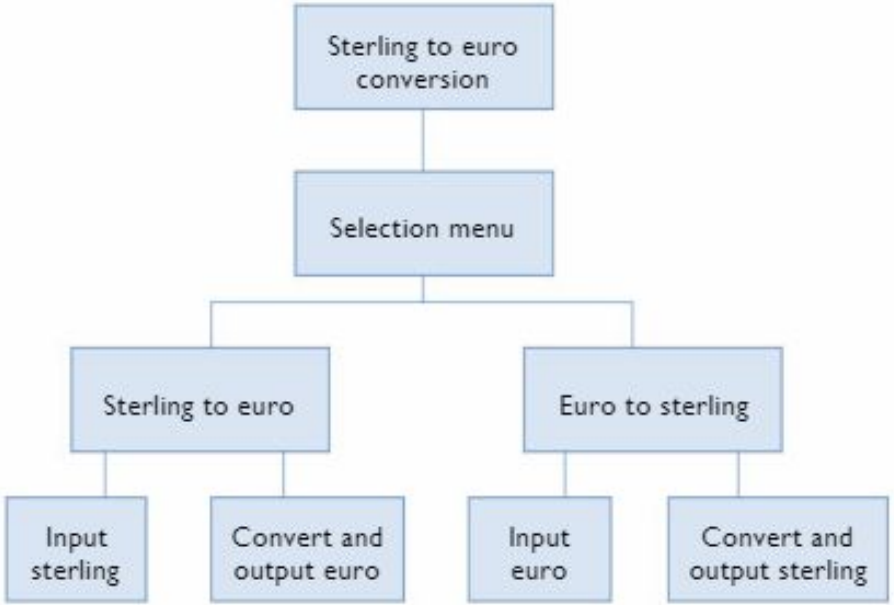


Approaches to design

Top Down Design

This approach starts with the main system at the top and breaks it down into smaller units in a hierarchical fashion.

Each unit is further broken down into smaller units.

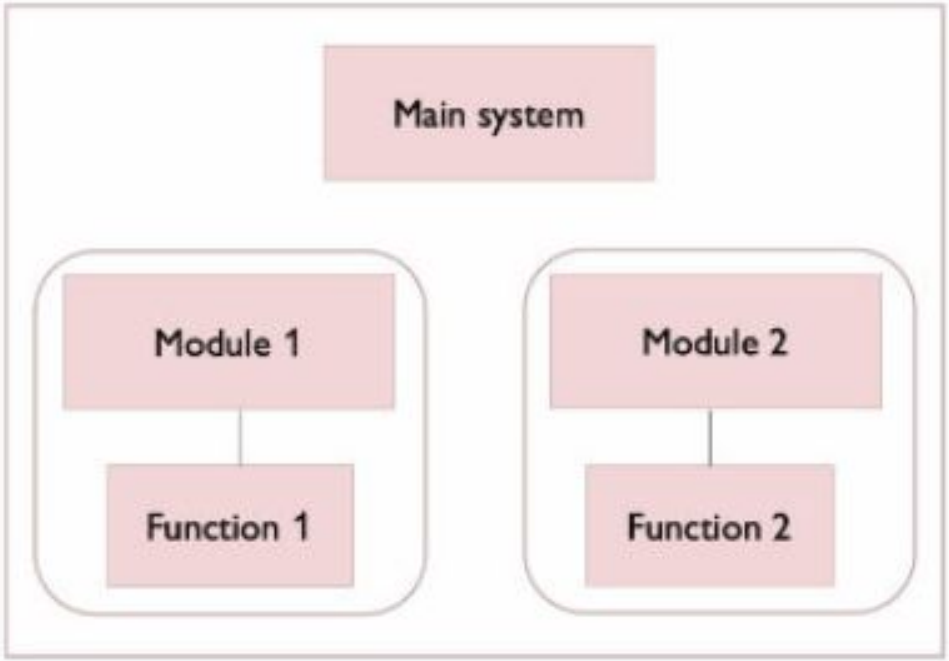


Approaches to design

Modular Design

System is divided into smaller units called modules.

Each module must represent a single self contained task or function that can be created independently of the other.



Approaches to design

Prototyping

A prototype is an early model of a product built to test a concept in advance and to get feedback from users.

Typically features only a few key aspects of the whole system.

Enables users to give feedback at an early stage - additional user requirements can be identified and problems recognised early on.

Often used in Engineering, Software Design.

Most valuable when designing user interfaces.

Approaches to design

Flowcharts

Represents the flow of operation of the system.

Clear graphical way of communicating

Difficult when complex logic is involved.

Design

Algorithms

All functionality should be described.

These can be shown using

Pseudocode
Normal Language
Flowcharts

Approaches to design

Data and Variables

All data should be listed along with their datatype.

Particularly important when dealing with databases

How will data be stored and backed up

Privacy concerns regarding data

4. Create - Implement the Plan

This stage involves implementing the plan/design.

Programmers create the system based on the work carried out in the previous stages.

The design might be modified due to issues that arise during implementation.

Unit Testing is carried out during this stage.

5. Evaluate - determine if the solution is appropriate

Testing to see if the solution works consistently.

Testing to make sure that the software does not do anything it shouldn’t.

More on Testing separately...

6. Document - report, present and reflect on the process

Records of all meetings, correspondence, communications, actions agreed, changes to requirements and design should be stored.

Evaluation of the project