

Lab 01: Software Prototyping

Software prototyping is the activity of creating *prototypes* of software applications. A prototype is an **incomplete** version of the software system being developed.

Prototyping is a commonly used activity in the software development process and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing.

A prototype typically simulates only *some* (a few) features of the final product. The prototype is often completely different from the final product.

Prototyping has several benefits:

1. The software designer and developer can get valuable feedback from the users early in the project. It aids the requirements elicitation and specification activity.
2. The client and the contractor can determine if the software being delivered matches the software requirements specification
3. It allows the software engineer some insight into the accuracy of initial project estimates and whether the proposed deadlines and milestones can be successfully met.
4. The degree of completeness and the techniques used in the prototyping have been in development and debate since its proposal in the early 1970's.

The original purpose of a prototype is to allow users of the software to evaluate the developers' proposals for the design of the eventual product by actually trying them out. This is a much more accurate way to interpret and evaluate the design rather than using written descriptions.

Prototyping can also be used by end users to describe and identify requirements that have been overlooked or misinterpreted. UED (user centered design) makes extensive use of prototyping and promotes good commercial relationships between the developer and client.

As you analyse the requirements for your chosen software system, you should build a prototype to help identify and demonstrate the functionality of each software application:

- Assigning of identifiers (key attributes)
- Inputs required (data entry or read from file?)
- Validation & error handling (messages)
- Confirmation of process completion
- Resetting of UI (clearing the UI)

There is no functionality (code) implemented with the exception of basic code to:

- Navigate through the software system
- Generate key values
- Demonstrate simple error handling and messaging
- Terminate the application

You will use Microsoft Visual Studio – C# to build a working prototype for your chosen system.