**Unit 4 – Software Design and Development**

**346584 – Mohammed Mahin Ibnay Mamun**

Software development life cycle - the software development life cycle is a process that software developers follow. It maps out the life of a software product. The life cycle starts with an idea (sometimes referred to as the ‘conception’ stage) and moves through each stage of development until it reaches the last stage.

**Learning Aim, A**

**Stages of software development**

**6 stages:**

* 1 conception
* 2 Analysis
* 3 Design
* 4 Implementation
* 5 Testing
* 6 Evaluation

**Conception** = the beginning of life cycle. It usually begins with an idea this is what the piece of software is going to be and why it is a clever idea to create it. This is part of the development process where the pre-project planning takes place.

**Careful consideration:**

* Is this new piece of software profitable?
* How will this software benefit your organization?
* Does it support your company’s position in the market?
* Dies it fit with the philosophy of the organization?
* What impact will it have on the staff of the organization?
* Does the organization have the human resources to cope with the project?

**Analysis** = once it has been decided that a project will go ahead and the software developed, the analysis stage is needed to identify the organization’s requirements.

This stage is the focus for project managers and stakeholders. Meetings with managers, stakeholders and potential users are held to determine the requirements of the project – specifically the requirements of the organization that wishes to create the software and the requirements of the software itself.

**Possible Question?**

* Who is going to use the software? And ‘how will the software be used?
* How much money will the software cost to develop?
* Who will carry out the work?

**Key Questions**

* What is the primary aims of the software system you are going to develop
* How does the current system work
* What other software systems does is need

**Design** = once the requirements of the software have been defined, you need to consider how the program will achieve what is required. This is the design stage. You will need to draw up a design for the software program.

**This design will define:**

* The user interface that will be provided. This might include designs for the screens that users will use to
* Input data, and the reports that will be output from the software system.
* The general structure of the program, including how it will be broken up into procedures and how those procedures relate to each other.
* The detailed design, showing how each of the procedures will carry out their required tasks.
* How data will be stored by system, including the variables that will be used and the file structures required.

**Implementation** = implementation translate the design into a working software solution and its deployment to its target environment.

* Which programming language is being targeted and why?
* Which hardware and software development tools are required?
* Is there any data to convert or training to provide?
* How will the project be developed?

**Testing** = this process will check if the chosen solution is working as required by:

* All the features and functions of the software program work correctly.
* Checking the quality of the finished software product.
* Testing that the software does not contain any bugs.
* It is also important to ensure that what has ...

**Evaluation** = when the software program has been created and is in use the processes bit yet over. Even with careful testing, it would be very unusual for a program not to experience problems when it is first released.

Therefore, an evaluation needs to be carried out. Problems experienced by users need to be corrected and there may also be other improvements or additions to using the program that may be required or desirable.

**Determination of scope and size**

Understanding the needs of the software program and obtaining all the relevant information that will be needed as part of the development of a software program is what scope and size is about.

The scope and scale of a software program are critical and should be determined at the conception stage of the software development life cycle.

Understanding the scope and scale allows a judgement to be made about whether a certain software development project Is helpful and sustainable.

To understand the scope of a software project we need to ask some fundamental questions through discussions with stakeholders and users:

* What will the software system do? Abd ‘What will it not do?

**How to apply each stage of development to get the best results**

To get the best, consider:

* What would happen if this did not take place?
* What if the customer did not like your design?
* What if you failed to identify an important requirement that was not elicited during analysis stage?
* What would be the consequence if you fail?

**TASK 2**

Can you recall what steps are involved within each stage?

* 1 conception
* 2 Analysis
* 3 Design
* 4 Implementation
* 5 Testing
* 6 Evaluation

Research about the different software development methods – particularly agile and waterfall methodologies. What are the advantages and disadvantages?

**Agile**: requirement, design, development, testing, deployment, review

The Agile methodology is an iterative approach to project management and software development that uses feedback loops and test-driven development to solve problems.

**Advantages**:

* Customers can look at the working features to see if they match expectations
* Customers can give feedback
* You can deploy the software quicker, so your customer can get value sooner
* You can detect and fix issues and defects faster
* You can get immediate feedback / Developers can improve their skills based on QA feedback
* You can experiment and test ideas because its costs are low

**Disadvantages**:

* Less documentation
* Requirements are not always as clear
* Agile demands more time and energy from everyone because developers and customers must constantly interact with each other
* Features that are too big to fit into one or even several cycles are avoided

**Waterfall**: analysis, design, implement, testing, maintenance

The Waterfall Model followed in the sequential order, and so the project development team only moves to the next phase of development or testing if the previous step completed successfully.

**Advantages**:

* Allows cost estimation
* Changes can be implemented in early stages
* Works well for smaller tasks
* Easy to manage due to the rigidity to the model

**Disadvantages**:

* Testing is delayed
* High amount of risk and uncertainty
* Changes can be costly
* Lack of customer involved after analysis

How does the software development life cycle compare with different life cycle models? Which one do you feel is best and why?