Assessment task 1

Ciarán Maher 10/11/2015

# Systems Life Cycles

## Agile Software Development Life cycle (SDLC) Model

The Agile development model is a type of incremental model whereby software is developed in incremental, yet short and rapid cycles. This results in small incremental releases each with additional functionality and bug-fixes built upon the work of all previous iterations.

Care must be taken to ensure that each release is thoroughly designed, reviewed and tested to ensure software quality is maintained. Agile is fundamentally ideal for short phases of each stage of development, which in result ensures that updates are very frequently completed and made available to the user. This system is ideal for software development when the product is already released, as this allows the development team to quickly release new features and bug-fixes as quickly as possible.

### Advantages

* Increases customer satisfaction by allowing for rapid and continuous delivery of software
* User interactions are emphasised rather than processes and tools.
* Usable software is delivered frequently (weeks rather than months or years)
* Face-to-face conversation allows for instantaneous feedback
* Allows for close, daily co-operation between business people and developers
* Regular adaptation to changing circumstances
* Allows for changes in requirements on a continuous basis

### Disadvantages

* It can be very difficult to assess the effort required at the outset of the project
* Less emphasis on necessary designing and documentation
* The project can easily get slowed down by on-going changes to requirements

### Examples

Extreme Programming (XP) is currently one of the most well-known agile development life cycle model.

SCRUM is another example of an agile development life cycle that I am familiar with.

## Waterfall Software Development Life cycle (SDLC) Model

The Waterfall model was one of the first process models to be developed, and as such is one of the most simple, and easy to understand. In a waterfall model each phase must be fully completed before the next phase can begin. This system tends to only be used for small projects with very specific requirements.

At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model testing only begins after the initial development is completed. This is because in a Waterfall model, the phases do not overlap.

### Advantages

* This model is very simple and easy to understand and use
* Due to the simplicity and rigidity of the model, it is very easy to manage
* Each phase has specific deliverables and review processes
* Phases are processed and completed one at a time; one ofter another allowing for clear and concise deliverables

### Disadvantages

Once a product has entered the testing stage of the Waterfall model, it can be very difficult to make new changes to the requirements/final product

No working product is produced until near the end of the life cycle

Initial investment must include the majority of the cost of the product before sales can begin, this can also mean high risk for investors

This life cycle is not very effective for managing and maintaining complex and/or object-orientated projects

This model is particularly poor for long and/or on-going projects as the requirements are very inflexible once the design stage begins

### Examples

# Importance of following a structured methodology

It is very important to follow a structured methodology when designing and developing new software products. There are many reasons for this, however the most important are:

1. **Efficiency**
   1. Following a structured methodology for developing any new product is important to keep man-hours and costs, as low as possible. When using any structured methodology, one of the first steps is to identify requirements for the product; and thus realistic goals and expectations can be set. This allows the team to ensure they remain on-target at all times.
2. **Documentation**
   1. A good structured methodology requires the team to produce reports during research, design and development stages. These reports can then be used to develop additional documentation such as technical documentation, user manuals and future updates.
3. **Good design principles**
   1. A structured methodology allows
4. **Taking advantage of Object-orientated Programming (OOP) tools**
   1. Following a structured methodology allows for the use and re-use of Object-orientated code in multiple, on-going products. Without clear and concise design methodologies this is significantly more difficult.