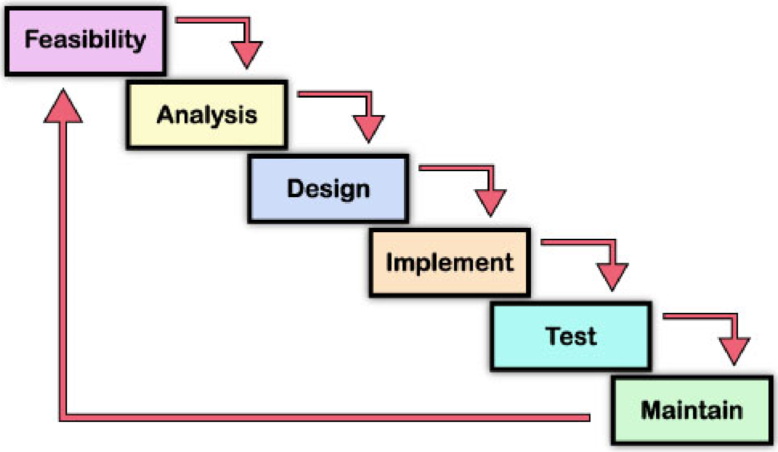
P2 – Illustrate the stages of a development life cycle

Introduction

In this assignment, I will be explaining each step of the waterfall method. Each step is different and I will be explaining what each part does for it. Each of them is important. One of them has to be done before the other can be done. Each step is shown in this picture and this is called the waterfall method. Waterfall method can implement it into an existing system or start it again from scratch. Obviously, starting from scratch will take longer than the existing system.



Feasibility

Feasibility study has different aspects of the development life cycle. This means a group/individual studies the technical, operational, economical and scheduled feasibility. They answer the project answers. Feasibility also covers the initiation part of the cycle. This means is worth looking at. Are there enough people in the system for it to be successful? Would the budget cover the whole cost? These questions are answered during this process. They cover the technical, economical, operational and scheduled feasibility. **Technical** side covers the expertise within the group. Can they cover the project or do we need alternative/added staff to finish the project? **Economical** covers the money part of the whole development. Do they need more time to get more cash to cover the project? **Operational** covers if it suits the user needs. They would need to ask the people to use it if they are able to use the future system. **Scheduled feasibility** covers how long they would take doing this. Would they need to delay it or start immediately?

Analysis

Analysing is done before because if you use an existing system, the group/individual have to analyse the system. The analysis is a cause and effect problem. For example, if they are any current causes that could affect it as a problem later on. Learning all about the background of where you are going to place the system. For example, in LSC, you need to know that they are different classes for it to be done. Another analysis is that is it worth doing it? Are there any benefits of doing this system later on in their life? For example, ELIP was created for a number of reasons for the teachers e.g. teachers to check the student's progress, fill in student's grades.

Design

Obviously, this is the part where, once all the agreement is complete, you start to put all the ideas into action. This part is important because all of the people have to be alert and digest all the information that they have. If they do not cooperate, they will not complete the task they have been waiting for, which is a long time. Moreover, this takes time to build. Any person would not take a month and finish it off. People would need a lot of time to create a new system. All of the people need to participate in the whole cycle for it to be accurately finished. If there are any issues that they have missed, they will need to go back to it and fix it which can take time.

Implementation

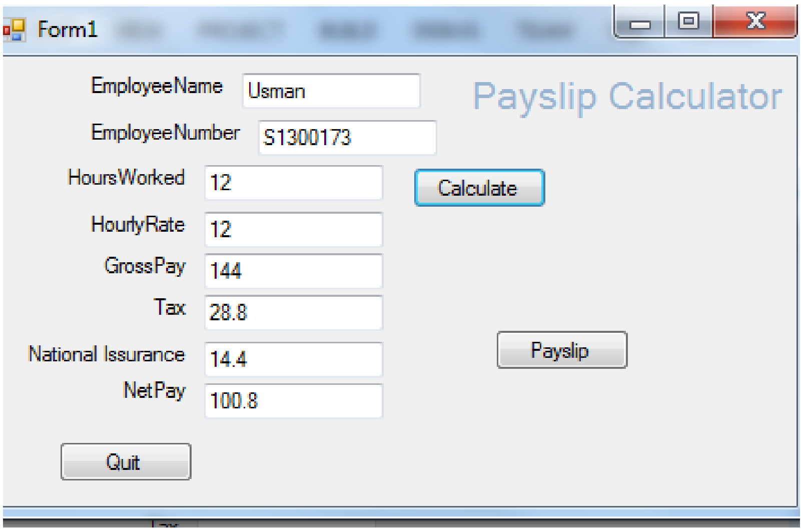
This is the part of the process where it ends. This checks includes the preparation of the new system. For example, if the design is done. You start implementing the process. You start putting the program into the system for it to work for the users. For example, ILP, once it is complete, it will be implemented into each student's account for it to work.   Implementing it the new features is important because you need to put it in each system. If someone needs the new system and it is not implemented in properly, this could cause a problem. And, if the user implements the new system without teaching the new user, this could be a problem for the user.

Testing

Testing is a big part of the whole process. I think it is the most important part after building the whole system. Testing is checking if the system works. If this part is not done, many problems could occur. For example, the user will need to process each section of it if it works. If it does not work, or another problem occurs, they have to solve the problem before releasing it. If the problem is not dealt with, other problems would occur and it would cost them.

Example

To test the program, I will always press the "start" button on visual studio. This will bring up the entire program together as shown on the picture to the right. If you insert the amount of hours you worked and the hourly rate, the calculation will be done for you once you press the 'calculate' button. If there were any errors once pressing the start button, it would be detected automatically. For example, the GrossPay adds the HourlyRate and HoursWorked. You will need to fix that as it shows by changing the code from 'plus' to 'times'.



EmployeeName	Usman
EmployeeNumber	S1300173
HoursWorked	12
HourlyRate	12
GrossPay	144
Tax	28.8
National Issurance	14.4
NetPay	100.8

Maintaining

Maintaining is the important and the part where it needs to be maintained. This means that each day, depending on how big the system is, they come and check how the system is coping. If they were any problems, they would fix it. This is important because any system needs to be maintained. Problems occur, if a virus is inserted, they will need to detect it and fix the problem. If they are big issues, the whole system may need to be shut down and be dealt with.

The example below shows that last year we made a program for it to calculates the net pay of how much an employee works and earns. We had to code it and this is an example below of how we are going to maintain it.

Example

It is always important that you monitor the program. This is to ensure that the program is not misbehaving. In addition, if you update the program, they might come across new features that can interrupt the current features that have attached to the program. If they keep up monitoring the program, the program would not have to deal with any errors and issues. Sometimes, without notice, the program or codes could delete. Even if there is an update on the program, the program could be upgraded with new features. Therefore, the programmer needs to check that no codes are missing and that everything is working perfectly. The advantage is that if you monitor the program, it detects any mistakes and errors.

The whole system can collapse if once part of this cycle it is not dealt with. All of the money that they have spent, it will go to waste if they do not do it properly.

M1 - Discuss the most appropriate uses of different development life-cycle models

Introduction

For this report, I will be discussing the two models; spiral model and waterfall model. I will be answering the questions below.

- What kind of systems do we create for each of these?
- How critical are these systems?
- Which system would you use?
- Advantages and disadvantages for each system
- When is one appropriate for each situation?

Spiral model

Spiral development models are a combination of both waterfall model and prototyping. This system is used for large projects that initially take a long time. This system can be changed repeatedly. For example, if the prototype is not up to standard with the users, they can request a change. The benefit is they can change it as many times. This can be used to build a program. The downfall for this project is the cost. As it takes time to complete, it will be expensive to build. Expertise are needed for this long project. To evaluate every situation because each part is important. I would prefer this model because they are many factors that contribute to my chose, which the prototyping, taking time on it enough changes to make it better. This system is critical because each step is important. This requires expertise in its field. The better you know the area, the more and better the result. As I have stated before, the appropriate system for this situation would be people who develop long projects e.g. making a new software (iPhone 6). IPhone 6 may be the replacement of the system (iPhone 5), but it takes even longer to add the new features that the customers want. The prototyping is a benefit as it can quickly show the users how it looks like. This would not be appropriate for a critical system because it would require tests and people would need to know what they are going to do.

Prototyping is a model that is takes days to create and it is only half of the system. One advantage of using prototyping is it is fast. They can show the people that use it a model of the actual system. However, this is only some of the system. It is only the example. This cannot be used on critical systems, because critical systems are very dangerous and it needs to be taken care off. CASE tools are used to convert it very quickly. In addition, Rapid Application Development (RAD) uses prototypes to develop it quickly. Using protocol, the user can tell the designer to add features that they would like to add to the actual system. This plays an important part of it as it is very quickly, consumes less time, and the people can use it to change it if they do not like the prototype.

Referring to Figure 1.1, this shows how the spiral model looks like. They will be many rechecks of each section. Hence, the spiral. It goes round and round until it is perfect.

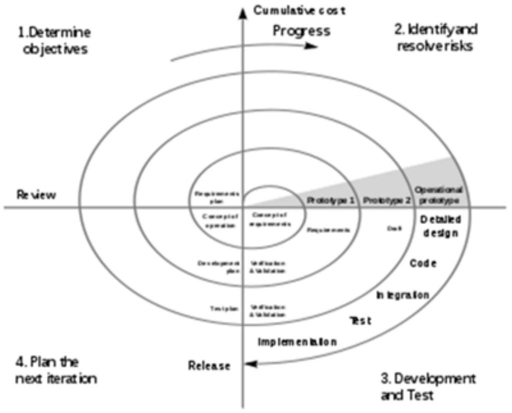


Figure 1.1

Waterfall model

Waterfall model is a long procedure, as it goes through each phrase of the cycle. This model is different to the spiral model, because they have different aspects to look at; but the concept and outcome is the same. We use these systems for any technology device/software. In LSC, we would use waterfall model for creating a register. This is important and straightforward. This is important because for any short-term project the analyst know what they are going to do. This system takes long to do with a lot of analysis. Once a decision is made, you cannot go back with it. This model is used for any business that would want to create a database, new system for their specific organisation.

The benefit for this model is that the stages are simple to understand. For example, once they look at the model, it would not be confused. It is straightforward. Each stage is reviewed properly. However, the disadvantage for this is that it cannot be changed. This is why small projects are a benefit. For example, if we are going to make a database for the NHS and we were at the stage of testing, we tested it and showed it to the employees. We made a mistake by not doing it into each department. This would cost a lot and we need to study our environment. Taking risks could be a caution to failure.

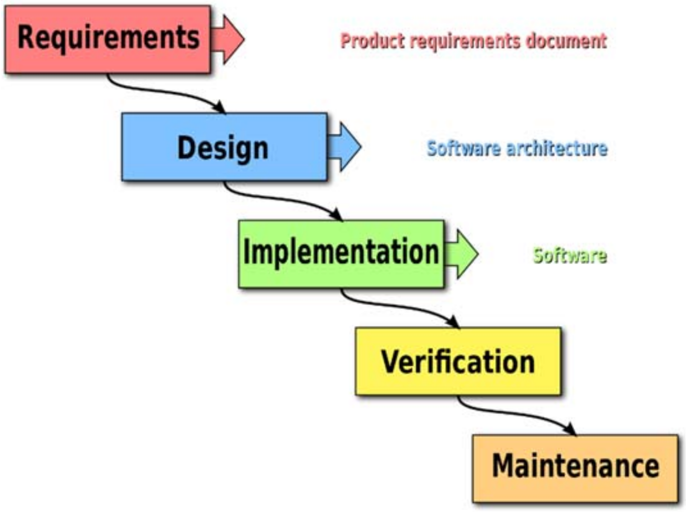


Figure 1.2

P3 - Explain the benefits of structured analysis.

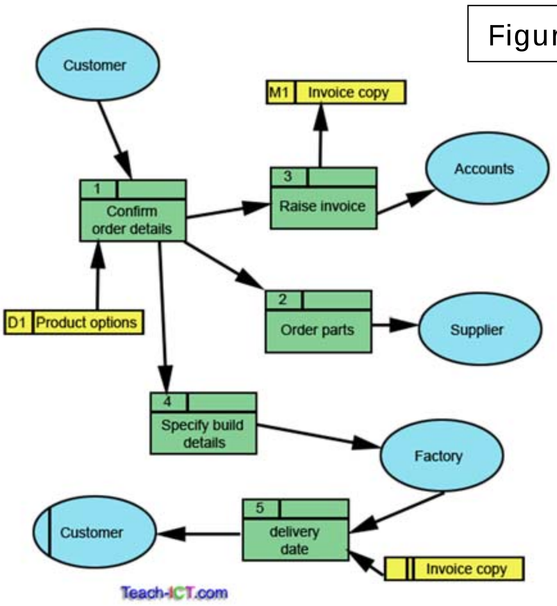
Introduction

In this assignment, they are many types of tools such as SSAMD, CASE, RAD and DFD. However, I will be naming two of them and the benefits of these tools; Rapid application development (RAD) and computer aided software engineering tools (CASE).

DFD

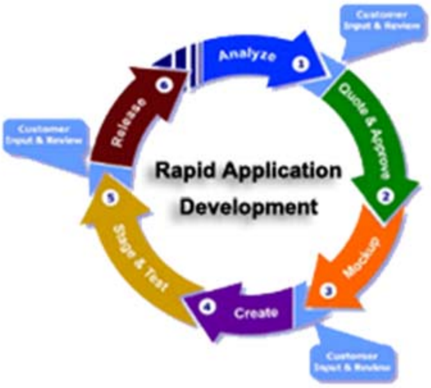
This is a flow that shows the kind of information the user has inserted and taken out. This is a detailed analysis that shows what the user used, where he used it and what he used. It looks a bit complex when another new user has seen this diagram. The user that created it, he needs to explain it so others understand. The benefit of using this structured analysis is that it is helpful to use for the analysis. One of the **benefits** is that it is easy to understand by any audiences depending on what system is created on. Another benefit is that it provides an overview of the whole system. It describes each component of how it looks like. Another advantage is that it is easy to learn. Therefore, it is easy to draw out. The **drawback** of using DFD is that they will be many alterations made during this process. Considering the changes, it will take time to do. As it gives an overview of the whole system, especially for a large system, it will become confusing. Using different symbols may confuse some people that have not dealt with any of DFD. Referring to figure 1.3, it shows how a DFD looks like. It shows the basics of how the system will work. This is simple and easy to understand. However, if it is large, it will be complex. This shows how a customer orders work. It shows clearly how it works and what is used. This is similarly used for system analysis.

Figure 1.3



RAD

Rapid development application is a fast approach technique that is used in software development. It is used in spiral model. Prototyping is used within RAD and has a major effect on the speed of the whole process. RAD is used for business-type projects that needs to be done very fast. The major **benefit** of using RAD is its speed. Unlike any of the others, it takes days to produce. With the speed, the analysts can show the users how it is used. They can make changes of what they want change to make it simpler. For example, in LSC, our ILP is a system that should off taken a very long time to create. This is because the users need to make accounts for each student; they need to put the information in (personal details) and link it to other software (register). In addition, the teachers need access to all of it for the BTEC/AS/A2 predicted grades on there. The **drawbacks** of this system is the quality of it. Although it is fast, the disadvange could be that the quality would be poor. It produces faster. Therfore, it would not be taken time on it. People only use it to show people how the actual system it is going to be.



Prototyping is a model that is takes days to create and it is only half of the system. One advantage of using prototyping is it is fast. They can show the people that use it a model of the actual system. However, this is only some of the system. It is only the example. This cannot be used on critical systems, because critical systems are very dangerous and it needs to be taken care off. CASE tools are used to convert it very quickly. In addition, Rapid Application Development (RAD) uses prototypes to develop it quickly. Using protocol, the user can tell the designer to add features that they would like to add to the actual system. This plays an important part of it as it is very quickly, consumes less time, and the people can use it to change it if they do not like the prototype.

REFERENCE

- **BTEC LEVEL 3 BOOK YR 2**
- **http://en.wikipedia.org/wiki/Rapid_application_development**
- **http://searchsoftwarequality.techtarget.com/definition/waterfall-model**
- **http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/**
- **http://upload.wikimedia.org/wikipedia/commons/thumb/e/ec/Spiral_model_(Boehm,_1988).svg/333px-Spiral_model_(Boehm,_1988).svg.png Figure 1.1**
- **http://upload.wikimedia.org/wikipedia/commons/thumb/e/e2/Waterfall_model.svg/1280px-Waterfall_model.svg.png Figure 1.2**
- **http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/analysis_tools/miniweb/images/data_flow_diagrams.gif Figure 1.3**

P1 – OUTLINE THE PRINCIPLES OF SYSTEM ANALYSIS

System Analysis and Design

System Analysis is the stages of the system where a problem is identified in the system and they are procedures to fix the problem. One of the problems could be in a school, the register is taken manually. This is a long and complex procedure where names are taken down on a piece of paper. This can be computerised. This is better as it automatically gets taken that the pupil is absent or present.

Development life cycle models

The **principle** of using this system is that systems are decomposable. To break big problems, convert it into small chunks . Models are used to represent these chunks. No person can do the whole system by its self. The user has to have a group to do this as well. It is important he/she does this. Every person must contribute to it. This is the stage of the development where planning and the modelling of the project comes into play. This is the part where the user manually draws out a rough sketch or model of the system/program. This would show how the program works. For example, if I were to get an idea that I was going to make a program that makes websites for you, (yes, I know there is already one, just an example) I would sketch it out and it would tell another person how the program is developed and how it works. Once they have made the model, they can look at it and enhance it so see if anything is wrong. Another person can review it and can tweak it a little to make it better. For example, if they were a simpler way to show the website on the web an easier way, I would tell the person to do it this way. They are many models and very popular models that are used such as waterfall model, rapid application development (RAD) and spiral model. Referring to Figure 1.1, it demonstrates how the model works. These are the steps of what the person would do after planning the software.

Another **principle** is that people are crucial. They need to understand the system because there is no point making the system if they do not understand. For example, in LSC, Columbus is a system which LSC uses for students details. This part covers the maintaining, even though the system needs to be accurate, the teachers/users need to know how to use it.

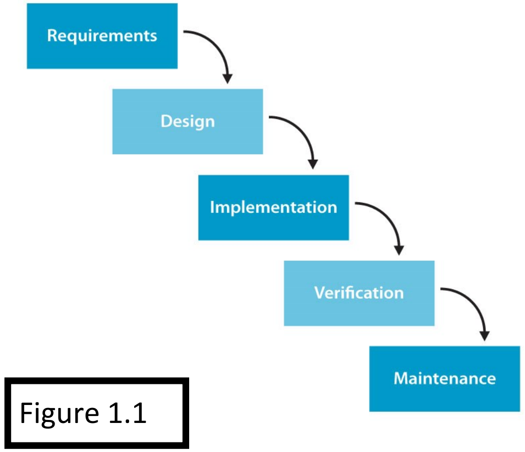


Figure 1.1

Key drivers

One of the **principles** that is necessary is to solve the problems earlier rather than later. This is the stage where the producer/creator looks at the financial part of the cycle. You would rather solve the problems earlier because once the problem has not been solved, they are other problems that could be created. This is either some requirements that are needed or components of the system. It is important to look at this principle because the budget includes in all of the planning. For example, if you want another person's logo, you would need to seek permission from him or her before using it. This is a legal requirement, otherwise the user that has unauthorised access, they would receive a hefty fine. This detailed phase includes other many business aspects of the program. Even during the planning stage, they have to answer all sorts of questions whilst doing it. Would they make a profit? Would it be a long-term benefit? Referring to Figure 1.2, it only shows a person signing a paper. However, they will be many papers to sign whilst doing the project.

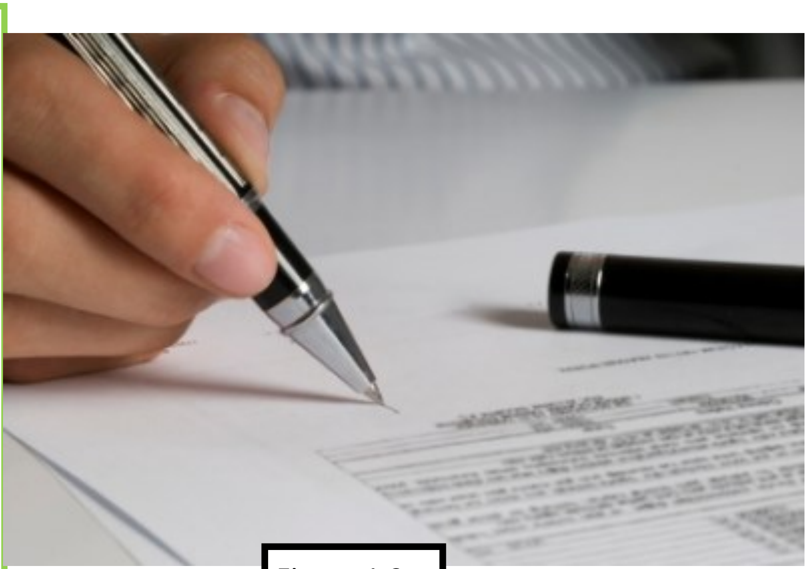


Figure 1.2

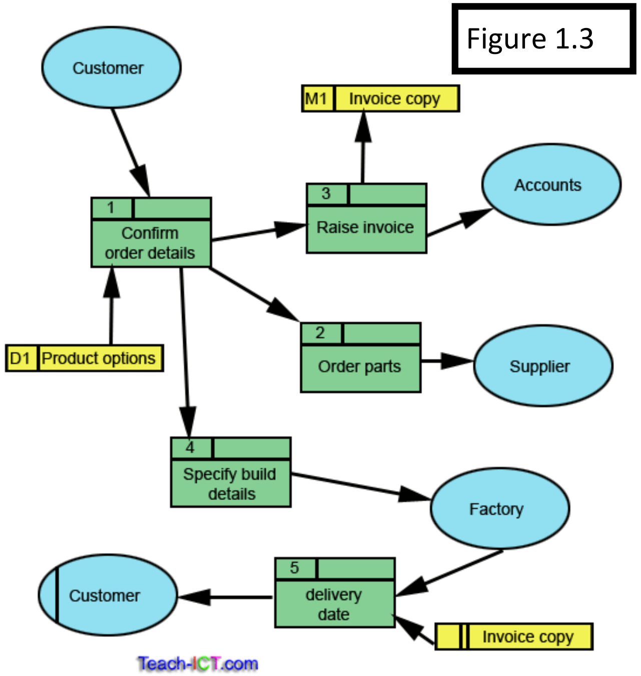


Figure 1.3

Developmental tools and techniques

Development **tools** and techniques is a phase of the process it is used in a system. One of the **principles** is that the requirements need to be fitted into it. They have tools and techniques of how it can be used. These have to be the exact ones for the program. Otherwise, the user will need to put the system at halt. Some examples that can be used in this system is Data Flow Diagram (DFD) and Structured Systems Analysis and Design Methodology (SSAM). These examples shows the overview of the system. The tools can be internet. If the user does not have internet, he cannot research about the project. Figure 1.3 demonstrates the diagram of how the tools and techniques how an development tools and techniques would be used. It would be laid out on a diagram of how and what tools they used for each department or part of the system. It will make it easier once the diagram has been complete as shown on Figure 1.3.

REFERENCE

http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/analysis_tools/miniweb/images/data_flow_diagrams.gif