SDLC and our project management model

The software development life cycle, or SDLC for short, is used by businesses to help define and structure software development projects. It separates development into a series of steps, which should be followed sequentially, from inception to deployment. The SDLC methodologies can vary from organisation to organisation, but the process remains similar all the same. The SDLC model is split into 6 different phases. These are as follows.

1. Gathering requirements and analysis: The first phase involves a lot of work by the project manager and stakeholders. The manager will meet with stakeholders to define what is required by the project, and from the meetings, the project manager and their team will analyse the requirements to see if they are feasible and able to be incorporated in the system. After this, a requirement specification document is created, which shall serve as a reference for further phases of the model (Britton & Doake, 2006).

We have been given out requirements in the activity brief, and using the python GUI labs, assessed what can be feasible ways of implementing these requirements

1. Design: From the requirements received in the first phase, a design is made for the system and the software. The design will help guide the coding and will help define the architecture of the system. The design will serve as input when you reach the next stage. (Britton & Doake, 2006)

In our project, we have completed activity flow diagrams, pseudocode and a storyboard for our robot arena and the code. These designs will help when we implement the code

1. Implementation / Coding: This phase will be the longest phase. In the implementation phase, the coders will take the designs created, and the code will be produced to make the program work. The work will be split into units, so multiple people can produce different parts of the program (ISTQB Exam Certification, 2014).

Within our project, we have used the designs produced, and have written up code in Python, using various modules.

1. Testing: Once the code has been produced, the code will be tested to see that it both works, and is fit for purpose - addressing the issues posed in the requirements document. In this phase, various tests are done: system testing, integration tests, unit testing and acceptance testing (ISTQB Exam Certification, 2014).

We have tested our code, and rectified our errors. This phase was completed alongside our coding phase, in an iterative fashion.

1. Deployment: Once the code is tested, and is fully functional, the product is deployed to the consumer. (Britton & Doake, 2006).

Our product was shown to other groups, who evaluated it and gave their feedback, as if they were the customer.

1. Maintenance: When the system is operational, occasionally bugs that weren’t picked up during the testing will occur. These will be solved within this phase. (Britton & Doake, 2006)

Based on the feedback received from our program demo, we altered the program slightly

The waterfall model was developed as a simple and disciplined method, which follows the SDLC very rigidly. The term “waterfall model” comes from its naturally cascading format, which looks very much like a waterfall (Marchewka, 2013). The waterfall model breaks down tasks into the same stages as the SDLC, however these stages are broken down into two phases: one where the work is carried out, and the second is verification and validation of the work – where verification asks whether the product is being built the right way, and validation asks if the product itself is right (Cadle & Yeates, 2008).

Another development approach is the spiral model. The spiral model is an iterative style of development model, which should be used when the requirements of the project aren’t well defined. Barry Boehm developed the model, which is split into four phases – the phase where objectives are determined, risks are analysed, the product is developed, and then a planning for the next iteration. This will continue until a product that satisfies the customer is created. (Cadle & Yeates, 2008)

We have decided to use the waterfall model as it fits very nicely with the SDLC, and our project doesn’t require iterative or incremental work. The waterfall model will therefore allow us to quickly complete our tasks and move onto the next.

# Works Cited

Britton, C. & Doake, J., 2006. *Software System Development*. 4th ed. Maidenhead: McGraw-Hill Education.

Cadle, J. & Yeates, D., 2008. *Project Management for Information Systems*. 5th ed. Harlow: Pearson.

ISTQB Exam Certification, 2014. *What are the Software Development Life Cycle (SDLC) phases?* [Online] Available at: <http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-phases/> [Accessed 30 October 2014].

Marchewka, J.T., 2013. *Information Technology Project Management*. 4th ed. Hoboken: John Wiley and Sons.