West Herts College

HND in Computing: Cybersecurity and Ethical Hacking

Assignment 1

Software Development Lifecycles

Fabijus Sobolevas
2018-2019

Contents

Introduction

Software development cycles are models which are used by businesses in order to carry out projects in an efficient and organised manner. There are two types of development cycles, Iterative and Sequential. Within these categories falls a number of different models which each have their own advantages and disadvantages. Some of these models, and their benefits and drawbacks, will be covered in this document.

Iterative Models

These are models which break down the process of software development, usually of a large application, into smaller chunks. In this type of development, the code is designed, developed and tested in different repeated cycles.

Agile

This is an approach to software development which develops the user requirements and the solution throughout the project by working closely with the customers and project team. It is used in projects which require attention to detail and for the software to be tailored to the needs of the customer.

Advantages

Agile allows for very malleable projects, unlike other approaches. This means that you can refer back to the client to see if the project is according to their needs, if not you can easily adapt it to better suit their needs.

Disadvantages

Agile relies heavily on good communication with the client, which in some cases can be problematic. For example, the client may not be happy with the initial product, which means that time will be wasted in order to create a new version of the product, which once again may not be what the client wants.

Cleanroom Model

This approach to software development intends to produce software which have a certifiable level of reliability. The focus of this model is to prevent defects rather than remove them. The model is based on five key strategies, which are the following:

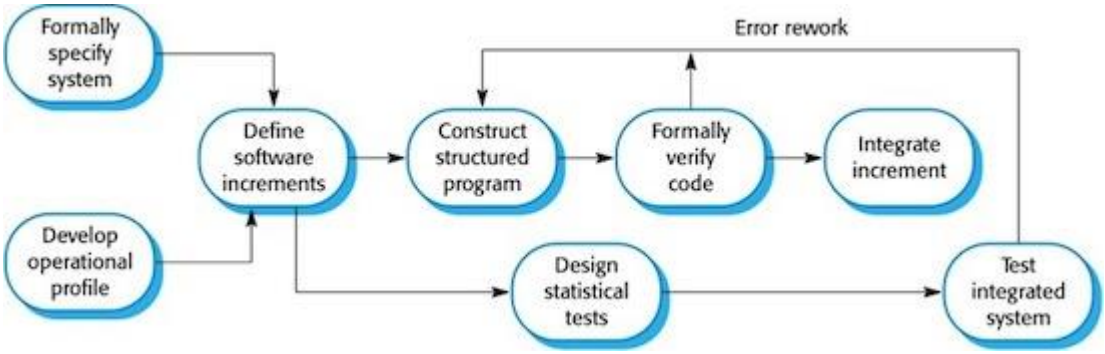
Formal specification - The software to be developed is formally specified.

Incremental development – The software is split into increments which are specified by customer input, then developed and validated using the cleanroom process.

Structured programming - The program development process is a process of stepwise refinement of the specification.

Static verification - The developed software is statically verified using rigorous software inspections.

Statistical testing of the system - The integrated software increment is tested statistically, to determine its reliability.



Advantages

The cleanroom model offers the advantage of the software being very robust against errors and defects, meaning it can be used in systems which require the software to have 100% uptime.

Disadvantages

One of the issues with having very robust software is that some features may not be implementable due to the risk of them causing errors or defects.

This type of software development takes a very long time in order to ensure that it is clear of all defects, which may mean that certain features cannot be implemented due to time or budgeting restrictions.

Sequential Models

These are models in which the project team spend a large amount of time gathering and refining the user requirements before the project design and development begin. The product is intended to be complete when it is released to the customers.

Waterfall Model

This model is a linear approach to sequential software development, it has low flexibility as the model flows mostly in one direction through all of the phases of the project. A common order for the phases of waterfall development are the following:

Requirements – The client requirements are defined.

Design – The system architecture is created.

Implementation – The code is developed and integrated.

Verification – The code is tested.

Maintenance – The support and maintenance of the complete project.

Advantages

The waterfall model is split into different phases; this means the project team are able to focus entirely on each phase to develop the best product.

If the client already knows exactly what they need and are able to simply define that, this type of model allows for an efficient way to build high quality software.

Disadvantages

There is little to no backtracking, meaning that if the end product is not ideal or does not meet all the requirements then the entire process has to be restarted from the first phase, this would cost a lot of time and money.

Spiral Model

This is a model which combines aspects of both the waterfall model and rapid prototyping methodologies in order to combine the advantages of top-down and bottom-up concepts. It provides emphasis on an area which most other methodologies do not focus on, which is iterative risk analysis.

It has four basic principles, which are the following:

Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.

"Each cycle involves a progression through the same sequence of steps, for each part of the product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program."

Each trip around the spiral traverses four basic quadrants: determine objectives, alternatives, and constraints of the iteration; evaluate alternatives; Identify and resolve risks; develop and verify deliverables from the iteration; and plan the next iteration.

Begin each cycle with an identification of stakeholders and their "win conditions", and end each cycle with review and commitment.

Advantages

Disadvantages

Spiral Risk Management

Conclusion

References

http://users.jyu.fi/~mieijala/kandimateriaali/Agile%20software%20development.pdf

https://resources.sei.cmu.edu/asset_files/TechnicalReport/1996_005_001_16502.pdf