Introduction

Modern technology and enterprises depend heavily on software development, and effective project management depends on a grasp of the different software development life cycle (SDLC) models. Three SDLC models will be compared in this course: the V-Model, Agile techniques, and the classic Waterfall model. We will obtain a thorough grasp of the various methods for software development by examining the benefits and drawbacks of each model.

Waterfall Model

In the sequential and linear SDLC paradigm known as "Waterfall," each stage of the development process must be finished before moving on to the next stage (Shatat, 2015). This paradigm offers a planned and controlled approach to software development, making it best suited for projects with clear needs and little changes (Shatat, 2015). The Waterfall methodology does have drawbacks, too, since it is frequently criticised for being rigid and unable to adapt to changes in the project's requirements, making it challenging to address client requests and feedback (Shatat, 2015).

Agile Techniques

While client happiness and cooperation between development teams and stakeholders are given priority by agile approaches (Kumar & Singhal, 2017). Based on the requirements and comments of the client, adjustments can be made as the development process progresses thanks to this flexibility and adaptability (Kumar & Singhal, 2017). Agile approaches enable for the project to change and adapt to changing circumstances, making them perfect for projects with quickly changing requirements or uncertain outcomes (Kumar & Singhal, 2017). However, because Agile techniques are decentralised, they can occasionally cause confusion and a lack of focus in the development process (Kumar & Singhal, 2017).

V-Model

The V-Model is a hybrid model that incorporates aspects of both Agile and the Waterfall models (Adams, 2019). The V-Model has two parallel tracks, one for development and the other for testing, with the development phases on one track (Adams, 2019). This paradigm maintains a sequential and linear evolution while allowing testing to occur early in the development process (Adams, 2019). The V-Model offers a systematic approach to software development while also enabling early testing and quality assurance, making it best suited for projects with stringent regulatory requirements and a need for rigorous testing (Adams, 2019). The testing phase must be finished before moving on to the next development phase, hence the V-Model might occasionally result in a lengthier development period can begin (Adams, 2019).

Conclusion

The selection of model will rely on the particular demands and requirements of each project. The three SDLC models covered in this course each have strengths and disadvantages. Software development can be structured and organised using the standard Waterfall paradigm, however it may not be able to adapt when project requirements change. Agile techniques promote adaptability and client happiness, yet they occasionally cause ambiguity and lack of focus. The V-Model contains aspects of both the Waterfall model and Agile techniques, but because it emphasises comprehensive testing, it may take longer to build.

In conclusion, understanding the advantages and disadvantages of each SDLC model is crucial for successful project management and the selection of the appropriate model will depend on the specific needs and requirements of each project.

References

Adams, B. (2019). Software development life cycle models and methodologies. Proceedings of the 2019 IEEE International Conference on Software Quality, Reliability and Security, 517-524.

Kumar, R., & Singhal, R. (2017). Agile software development life cycle (SDLC) model. International Journal of Advanced Research in Computer Science, 8(3), 454-460.

Shatat, M. (2015). A comparison between the traditional waterfall model and agile methodologies in software development. Journal of Theoretical and Applied Information Technology, 73(1), 131-139.