UML Models within the Software Development Life Cycle

When looking at the Software Development Life Cycle (SDLC), it is possible to attribute different UML models to each stage. These are the stages as defined by Everett and McLeod (2007).

Plan - Although this stage of the SDLC has limited involvement with the actual software production, UML can still be used within this. Take for example the need to co-ordinate from a managerial point of view who exactly will be doing what in terms of organising finances and allocating resources, it would be beneficial to apply a Class Diagram to this. By creating a class for each member of the team, you can highlight what they will do and who they will interact with. Regarding financing a project, an activity diagram could reflect this well, as it allows for variables for different parts, such as if financing wasn't sourced.

Design - Naturally, all of the applicable UML models would be used within this part of the SDLC. I would pay specific attention to Use Case Diagrams for this part however, as it is important to understand how the piece of software being designed will be used. As software engineers, we can sometimes lose sight of the needs of customers, many of whom do not have a technical background, so Use Case Diagrams are key to ensure that they are fully considered.

Implementation - During this stage, in which the code is written, there are opportunities for UML models to be used to help this progress. Activity Diagrams can

be useful to ensure that the logical processes of a piece of code are in order, ensuring that all variables are considered.

Testing - During the testing phase, it would be beneficial to utilise all UML models to ensure that all capabilities that were originally designed have been added. These should ideally be the same models used within the design phase. Specifically, Class Diagrams would be useful at this point to ensure that all attributes and functions have been built into the system.

Deployment - Again, during the deployment stage it is important to maintain consistency while moving the software across to the live version that customers will use, so utilising the same UML models that have been developed already would be crucial. It would also be beneficial at this point to ensuring that the Use Case Diagram is fully consulted, as this is when customers will begin to use the product.

Maintenance - During the maintenance phase, the Activity and Sequence diagrams would be beneficial to ensure that processes are running how we expect them to do so. If a customer reports a bug with the system, these can be consulted to identify where the issue is coming from and can provide the timeline for fixing this.

References

Everett, G. & McLeod (Jr), R. (2007) *Software Testing: Testing Across the Entire Software Development Life Cycle*. Wiley and Sons. IEEE Press.