Reflections of the role of UML in Object Oriented Programming

The features of Object Oriented Programming (OOP) have been considered throughout the first 4 units. Firstly, the key metrics of reusability mark a unique feature of the paradigm. Unified modelling language (UML) provided structured and consistent means of communicating throughout the software development lifecycle (SDLC). UML can provide high-level design though model management or granular detail for programmers through class and action diagrams.

While Padhy et all state that their future goal is to, 'reduce the maintenance cost and staffing save time.' (Padhy et al, 2018, p436), it may be better to be concerned with the quality of the software, the longevity and user satisfaction. The role of UML is pivotal to success in this area. At the beginning of the project, UML provides an effective means of communication between the developers and the client. The high-level design aspects can capture the key features of the software, ensuring that the goals of the programmers are aligned to the goals of the client.

However, when developing the software, the UML should not be a constraint, but an informative guide. It may be the case that highly skilled programmers can find for elegant and efficient solutions which may, from time to time necessitate alteration to the attributes or methods of associated objects. This should be encouraged, and embraced, but it is important for programmers to update the UML models accordingly.

The very nature of the OOP paradigm mean that it is easy for programmers to develop new methods or include new attributes as needed without the need for rewriting thousands of lines of code. Moreover, the nature of modules and encapsulation facilitate updates rather than compete rewrites.

The fluidity of the process aids an agile approach to the SDLC which, inevitably, results in a better product. This in turn means that the maintenance and updates to software can be progressed aided by updated UML models. The UML models should be seen a suite of live documents to extend the lifecycle rather than a simple blueprint for design.

References

Padhy, Neelamadhab, Suresh Satapathy, and R. P. Singh. "State-of-the-art object-oriented metrics and its reusability: a decade review." In Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 1, pp. 431-441. Springer Singapore, 2018.

Glinz, M. (2000) ‘Problems and deficiencies of UML as a requirements specification language’, *Tenth International Workshop on Software Specification and Design.* San Diego, 07 November 2000. San Diego: IEEE. 11-22.