ASSIGNMENT 3 – READINGS IN SOFTWARE ENGINEERING

Can the cost estimation performed during object-oriented analysis and design workflows be accurate? Why or why not?

When developing software, accurate cost estimation is critical (Stephen, 2011). Underestimation of the actual cost can lead to loss of money for the development organization while overestimation of the actual cost can lead to loss of client for the development organization (Stephen, 2011). Research has been carried out by some organizations to ascertain whether cost estimation performed during object-oriented analysis and design workflows can be accurate, and the conclusion has been that cost estimation can be considered reliable and accurate when carried out during the software development life cycle (SDLC).

A study carried out by Briand and Wüst (2001) discusses an analysis technique for the estimation of effort based on coupling and the size of a class during the design phase of the SDLC. Their technique focused on collecting data on a C++ system and analyzing the source code of the system to extract design information using an analyzer that they developed themselves. They collected information regarding the effort used on designing and coding the class, documenting, testing, and correcting faults. Other cost factors such as programming language, application domain, the experience of the developers, were not considered in their research because of the limited size of their project. Hence, the cost estimation was focused on the design phase. However, the cost estimation was said to be reliable because the design information that was used to quantify the design properties could be extracted from various UML diagrams, some of which are available at the early analysis stages (e.g. class diagrams), while other diagrams (e.g. sequence diagrams) depicting detailed interactions between objects are typically artifacts of late low-level design (Briand and Wüst, 2001). After carefully analyzing all collected information, they concluded that “there is a reasonable chance that useful effort estimation models can be built during the analysis and design face of object-oriented systems" (Briand and Wüst, 2001, p. 981).

Another study carried out by Rao and Achutharao (2014) focused on estimating the effort of various software projects using the Class Point Approach. During the Class Point Analysis, the parameters were optimized using various artificial intelligence (AI) techniques which include; Multi-Layer Perception, K Nearest Neighbor Regression, Radial Basis Function Network, fuzzy logic with various clustering algorithms, K-means clustering algorithm, and Subtractive Clustering algorithm. They operated data derived from forty student projects that were developed using the Java programming language. They concluded that "the effort estimation using the RBFN model will provide more accurate results than other AI techniques" (Rao and Achutharao, 2014, p. 252).

So also, a study conducted by Zhou, Yang, Xu, Leung, and Zhou (2014) made use of different modeling techniques to build prediction models for examining the accuracy of six types of metrics to estimate source code size. The modeling techniques include linear and non-linear models, tree-based models, and instance-based models while the examined metrics include class diagram metrics, predictive object points, object-oriented project size metric, fast & serious class points, objective class points, and object-oriented function points. Using 100 open-source Java systems, they discovered that "the prediction model built using object-oriented project size metric and ordinary least square regression with a logarithmic transformation achieves the highest accuracy" (Zhou et al., 2014, p. 233).

This paper shows that there are several ways to estimate the cost of a software product which includes using AI techniques and using modeling techniques with investigated metrics. Of all the AI techniques tested, the Radial Basis Function Network model provided the most accurate result. Of all the modeling techniques and investigated metrics examined, the object-oriented project size metric and the least regression with logarithmic transformation produced the highest accuracy. Therefore, cost estimation can indeed be relied upon to be accurate when performed during the object-oriented analysis and design workflows.

References

1. Briand, L. C., & Wüst, J. (2001). *Modeling development effort in object-oriented systems using design properties*. IEEE Transactions on Software Engineering, 27(11), 963-986.

2. Ra, M. U, & Achutharao, A. (2014). *Effort Estimation for Object-Oriented System Using Artificial Intelligence Techniques*. International Journal & Magazine of Engineering, Technology, Management and Research, 1(10). 248-252.

3. Schach, S., R. *Object-Oriented and Classical Software Engineering, 8th ed*. New York: McGraw-Hill, 2011.

4. Zhou, Y., Yang, Y., Xu, B., Leung, H., & Zhou, X. (2014). *Source code size estimation approaches for object-oriented systems from UML class diagrams: A comparative study*. Information and Software Technology, 56(2), 220-237.