COMP 410
Software Engineering

Assignment 3
Readings in Software Engineering

Jason Bishop
3042012

July 25, 2015

Question

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

Response

Cost estimation is critical in any software development project. As highlighted in *Object-Oriented and Classical Software Engineering,* underestimation of costs leads to lost revenue for the development organization, while overestimation of costs may cause the purchasing client to reject development of the software (Schach, 2011). The estimation of cost for a particular project can be based on many many factors, but usually the determining factor is the estimation of effort taken to build a project. Effort estimation, in turn, is often based on the size of project. Project size can be measured using a number of metrics including, but not limited to, lines of code (LOC), thousand lines of code (KLOC) and function points. There has been significant research into many effort/cost estimation techniques for object-oriented software projects. Conclusions reached by these studies seem to indicate that, even in the earlier stages of analysis and design of a particular project, reliable and accurate cost estimation can be achieved.

A paper by Briand and Wüst (2001) discusses an analysis procedure for the estimation of effort based on class size and coupling during the design phase of an object-oriented project. Their methods focus on collecting data from developer logs and comparison with artifacts collected directly from the source code. Therefore the data obtained is not from the design and analysis workflows specifically, but this is deemed valid because:

> ...the design information we use here to quantify design properties can potentially be extracted from various UML diagrams, some of which are available at early analysis

> stages (e.g., class diagrams), while other diagrams such as collaboration or sequence diagrams showing detailed interactions between objects are typically artifacts of late low-level design. (Briand & Wust, 2001, p. 964)

After carefully analyzing data with various statistical techniques, they conclude “that useful effort estimation models could be built during the analysis and design of object-oriented systems” (Briand & Wust, 2001, p. 981).

A more recent comparative study by Zhou, Yang, Xu, Leung and Zhou (2014) focuses on different techniques to estimate the size of a project based on UML class diagrams. To perform the analysis, six class methods were extracted and analyzed from the source code of various open-source Java projects. A comparison of different models was done comparing predicted and actual size values, with a statistical analysis of the results. They found “that object-oriented project size metric, class diagram metrics, and objective class points metric are able to accurately predict source code size of object-oriented systems” (Zhou et al., 2014, p. 233).

Finally, the most recent paper discussed, by Rao and Archutarao (2014), expands the effort estimation process with the comparison of 3 different artificial intelligence (AI) techniques: Multi-Layer Perceptron (ANN), K Nearest Neighbor Regression (KNN), and Radial Basis Function Network (RFBN). The models were run to to compare Java based systems and compared with the class point analysis (CPA) technique, a non-AI technique that is frequently used in software development. The paper concludes “that the effort estimation using the RBFN model will provide more accurate results than other AI techniques” (Rao and Archutarao, 2014, p. 252).

From the work of these study authors, it is clear that there are many different ways to estimate the size and effort, and therefore cost, of a particular software product. The analysis shows that several methods can be a reliable predictor of cost for an object-oriented software project. This is because the information extracted from various analysis and design artifacts, such as class diagrams, communication diagrams, and other UML diagrams, accurately reflect the specific properties of classes within the software project. Object-oriented systems are almost exclusively based on these classes, and therefore, are provide accurate information to model the future size, effort and costs of these projects.

References

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

Rao, 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). Retrieved from http://www.ijmetmr.com/oloctober2014/MogiliUmamaheswaraRao-Aachutharao-D2.pdf.

Schach, S. R. (2007). *Object-Oriented and Classical Software Engineering (8th ed.)*. New York, NY: McGraw-Hill.

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.