AT3 - Solutions for Overscoping

Zachary Zerafa

April 2023

1 Abstract

The nature of agile methods exposes a risk of scope-creep, which can lead to delays and compromised project quality. To counter such an issue, cost-estimation is used to run analysis tests on the project's attributes and determine the scope in numeric terms, acting as a decision support system in explicating to what extent a project should be developed. This article will delve into the cost-estimation solution and the various techniques it employs as well as provide an experiment on how it would be conducted in a realistic scenario.

2 Overscoping

2.1 What is Overscoping?

Over-scoping (otherwise known as scope-creep) is defined as undertaking a specific or abundance of requirements that cannot be fulfilled within the time, financial or resource constraints of a development team. This can occur in various contexts, such as overly ambitious stakeholders or a lack of experience of developers, however most cases demonstrate the quality of having an "unrealistically large array of functions" (Bjarnason et al., 2012).

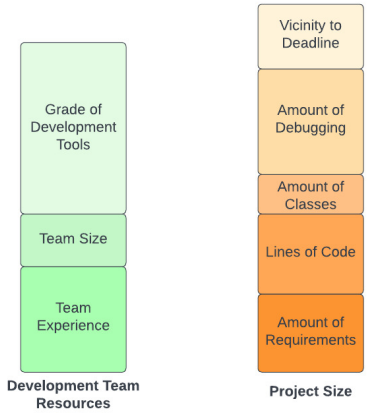

Figure 1: The image above gives a graphical example of scope-creep; where the specifications of a given task are larger than the resources of a development team.

2.2 Reasons for Scope-Creep

Scope-creep may arise from many different factors, including miscommunication of the development process with stakeholders. When stakeholders are misinformed on the issue of scope-creep, there is a possibility of the magnitude and quantity of requirements superseding the capabilities of their development team and leading to redundant work.

Over-scoping also occurs due to a difficulty in synthesising a set of requirements that suits the specifications of each individual stakeholder (Bjarnson et al. 2010). A misguided project as such may lead to prolongations of the project due to the slow development of foundations for the software project.

3 Outline of Cost-Estimation Solution

3.1 Cost-Estimation

Cost estimation encompasses a variety of techniques utilized to estimate the financial and chronological cost that requirements of a stakeholder require (Pfleeger et al., 2005). The objective of this solution is to process data on the project's qualities as well as past projects to produce information describing the travail required to successfully comply with the planned requirements, determining the presence of scope-creep.

3.2 Cost-Estimation Techniques

Cost estimation techniques are based on mathematical and machine learning frameworks that apply statistical analysis such as regression as well as system equation modelling (Jorgensen et al., 2007) paired with the data of previous software projects that assists in approaching a quantified estimate that can offer great assistance in deciding whether a requirement is beyond the means of a development team.

Though outlining the relationships between variables offers grand assistance in cost-estimation, regression proposes many assumptions on the data provided such as the data not containing outliers, the data is normally distributed, and that the data is in fact correlated by the chosen polynomial function (e.g The amount of workers must have a linear relationship with the amount of effort for the project)

3.3 Regression

Regression refers to modelling a linear function based on previous datasets which demonstrates the correlation between two independent types of data (Jorgensen et al., 2007). Regression may take one independent variable (simple) or may produce an output based on several independent inputs (multivariate) (Sharma et al., 2020). Regression can also be mapped through both linear and polynomial functions, depending on the data's correlation. Linear regression equation are used to produce a line-of-best-fit that outlines the trend of the relationship between

two variables.

$$Y_i = \beta_0 X_i + \beta_0 + \varepsilon_i$$

Figure 2: Simple linear regression formula (Matson et al., 1994)

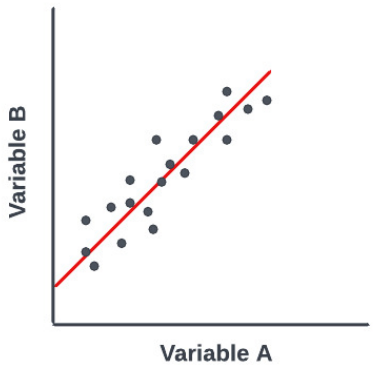

Figure 3: Line-of-best-fit; Without sufficient data, an accurate regression function may not be calculable.

3.4 Efficacy of Regression

Compared to other techniques, it is particularly reliant on a strong dataset, and has become feasible primarily due to the large collection of data of software projects in recent decades (Jadhav et al., 2022). The data of previous case studies is used in generating the gradient, finding the intercept, and determining the error. Without a reliable and populated data set, regression may have an error too large to have a meaningful result. On the contrary, error estimation can also add to regression's merit as it is particularly useful in describing the range of accuracy to expect the output to replicate, through formulae such as the Root Mean Squared function (Arslan, 2019; Kaushik et al., 2019).

Regression is the most commonly used method for cost-estimation in software development settings (Gorod et al., 2019) due to its ability to outline a proportional relationship between cost values (such as hours and lines of code) and resources (like team size and tools) of the modeller's choice rather than those specified by designed equation system frameworks.

3.5 Simultaneous Equation Modelling

In addition to regression, cost-estimation can be conducted through a range of different simultaneous equation models (SEM). These are specific frameworks that are either a static or dynamically generated system of equations taking quantitative data describing the software project in question to produce an output predicting the magnitude of the project's cost, effort, and time (Gorod et al, 2019) by parsing this data through the equation system set to specific variable settings. There exist systems of equations used by development teams to predict project costs, one of the most commonly employed being the Constructive Cost Model (COCOMO). This particular model developed by Barry Boehm chooses the primary parameter as the amount of lines of code (LOC) the project contains (Goyal et al., 2018), and sets variables based on the context of the

development team.

$$E = a(KLOC)^b$$

Figure 4: The effort equation in the COCOMO model (Goyal et al., 2018), which uses two set variables a and b to manipulate the amount of lines of code (divided by 1000) to estimate an effort value for the project.

3.6 Efficacy of SEMs

Although SEMs are useful frameworks, they are ultimately bound by the quality and abundance of data in addition to the precision of the modelling and setting of the mathematical techniques employed. These models are able to process general quantifiable data proficiently, however have a lack of qualitative measure which can discount information that can further refine the cost-estimation process such as accounting for the development team culture and individual skill set of each worker (Pospieszny et al., 2017).

4 Prerequisite settings and data

4.1 Experiment description and considerations

This article's experiment will be using regression techniques to visualise the accuracy of COCOMO effort estimation in providing an accurate result by comparing COCOMO's effort estimation with the true effort. A development team may undertake this experiment when they want to use COCOMO models for projecting cost, but also want to understand the margin for error that may be present when doing so. The foundations of this project rely on knowing what variables the functions are going to take so that a suitable output is produced.

$$P \times T = E$$

Figure 5: "Persons required times time equals effort"; the COCOMO equation that will serve as the basis of this experiment.

4.2 Data set

A strong data set is intrinsic to properly executing cost-estimation techniques as data with high accuracy will converge to a more accurate prediction of resources required. Accuracy in data is achieved by using data in previous cases that is comparable to the project in context as well a sufficient quantity of different types of data points (Alex Gorod et al.). This is acquired so that cost can be estimated based on a variety of different characteristics instead of, for instance, predicting cost using only the size of the development team, but rather make estimations based on how the development team size and time spent on project in synthesis can affect the amount of effort necessary to meet a deadline. Therefore the experiment will take a dataset from SEERA that contains authentic data from real software projects with a large pool of different data points from which analysis can be conducted.

4.3 State of Team and Requirements

Setting for equation modelling requires having a precise idea of the current state of the development team and requirements so that variables can be set to suitable values. For instance, a case study of a company with an undisclosed name was conducting cost-estimation on the effort a project requires, they chose to proportionally decrease variables due to their smaller development team size (4 developers) as well as mid-range project size (25,000 lines of code) (Butt et al., 2021). COCOMO models often have a variety of different development modes that offer values for constants based on specific project environments, for instance, organic, semidetached, and embedded (Goyal et al., 2018).

5 Experiment Process

To form a linear regression analysis, a collection of processing tools are required to perform the calculations such as determining the line-of-best-fit. For this project, Python with mathematics modules will be used as the use of such a programming language allows for a large degree of control in the process of formulating the exact values to parse. For instance, Python allows for a range of mathematical functions to be used to mix different data points together to assist in creating a stronger correlation (in the case of this experiment, Python will amalgamate the team size and assignment duration). A CSV file with real case cost-estimation data downloaded from SEERA will then be included within the Python code, allowing the regression to include the data on which a correlation will be identified and the line-of-best-fit will be calculated.

6 Results

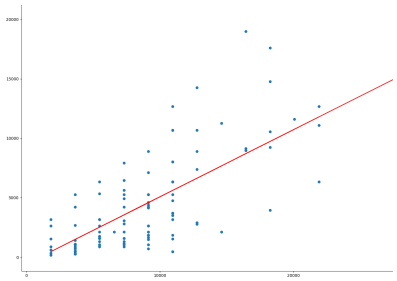

Figure 6: The experiment takes the horizontal axis to represent the team size multiplied by the project time, while the vertical axis represents how much effort the project tool in reality. All data here is sourced from SEERA (Mustafa et al., 2020).

After processing the data through the Python program, a linear trend with albeit lower correlation is identified. The gradient of the line-of-best-fit is within the range of $(0, 1)$, which implies that the COCOMO formula used to calculate effort tends to return a value lower than the actual effort required. However, the graph displays a disparity of data with many more lower estimations than larger estimations, meaning that there is a data bias towards smaller scale projects. A project with a considerably shorter deadline and smaller team may benefit greatly from these results, using this information to consolidate faith in COCOMO methods of cost-

estimation, while projects of a larger scale may need a larger data set to make an informed decision.

7 Comparison to State of the Art Solutions

A differing solution to cost-estimation includes establishing a formal communication system, which is effective in dissolving misinterpretations of requirements so that excess work is not undertaken (Ajmal et al. 2019). In comparison to cost-estimation, this solution can be more cost effective however it does not have the same accuracy and meticulosity of precision as cost-estimation, akin to the demonstrated experiment. Cost-estimation is able to systematically gather information that can verify whether scope-creep is imminent in a project and knowing when to withdraw requirements, while ameliorating communication networks can only prevent scope-creep in more peculiar situations, such as having misinformed stakeholders or vague requirements. Hence cost-estimation proves its value above other solutions by its ability to provide a structured assessment of scope-creep.

References

[1] Mian Ajmal, Mehmood Khan, and Hanan Al-Yafei. "Exploring factors behind project scope creep – stakeholders' perspective". In: *International Journal of Managing Projects in Business* 13 (Nov. 2019), pp. 483–504. DOI: 10.1108/ijmpb-10-2018-0228. URL: https://www.emerald.com/insight/content/doi/10.1108/IJMPB-10-2018-0228/full/html.

[2] Farrukh Arslan. "A Review of Machine Learning Models for Software Cost Estimation". In: *Review of Computer Engineering Research* 6 (2019), pp. 64–75. DOI: 10.18488/journal.76.2019.62.64.75. (Visited on 11/22/2021).

[3] Elizabeth Bjarnason, Krzysztof Wnuk, and Björn Regnell. "Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering". In: *Information and Software Technology* 54 (Oct. 2012), pp. 1107–1124. DOI: 10.1016/j.infsof.2012.04.006. (Visited on 06/07/2019).

[4] Shariq Aziz Butt Butt et al. "A Cost Estimating Method for Agile Software Development". In: *A Cost Estimating Method for Agile Software Development* 1 (Sept. 2021).

[5] Alex Gorod et al. *Evolving toolbox for complex project management*. Crc Press, Taylor Francis Group, 2020.

[6] Somya Goyal and Anubha Parashar. "Machine Learning Application to Improve COCOMO Model using Neural Networks". In: *International Journal of Information Technology and Computer Science* 10 (Mar. 2018), pp. 35–51. DOI: 10.5815/ijitcs.2018.03.05. (Visited on 04/30/2023).

[7] Anil Jadhav, Mandeep Kaur, and Farzana Akter. "Evolution of Software Development Effort and Cost Estimation Techniques: Five Decades Study Using Automated Text Mining Approach". In: *Mathematical Problems in Engineering* 2022 (May 2022). Ed. by Amandeep Kaur, pp. 1–17. DOI: 10.1155/2022/5782587. (Visited on 09/06/2022).

[8] Magne Jorgensen and Martin Shepperd. "A Systematic Review of Software Development Cost Estimation Studies". In: *IEEE Transactions on Software Engineering* 33 (Jan. 2007), pp. 33–53. DOI: 10.1109/tse.2007.256943.

[9] Anupama Kaushik, Devendra Kr. Tayal, and Kalpana Yadav. "A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO". In: *Arabian Journal for Science and Engineering* (Nov. 2019). DOI: 10.1007/s13369-019-04250-6. (Visited on 11/29/2019).

[10] J.E. Matson, B.E. Barrett, and J.M. Mellichamp. "Software development cost estimation using function points". In: *IEEE Transactions on Software Engineering* 20 (Apr. 1994), pp. 275–287. DOI: 10.1109/32.277575. (Visited on 05/14/2021).

[11] Emtinan I Mustafa and Rasha Osman. "The SEERA Software Cost Estimation Dataset". In: (Aug. 2020). DOI: 10.5281/zenodo.3987969. (Visited on 04/30/2023).

[12] Shari Lawrence Pfleeger, Felicia Wu, and Rosalind Lewis. *Software Cost Estimation and Sizing Methods, Issues, and Guidelines*. Sept. 2005. (Visited on 04/30/2023).

[13] Przemyslaw Pospieszny, Beata Czarnacka-Chrobot, and Andrzej Kobylinski. "An effective approach for software project effort and duration estimation with machine learning algorithms". In: *Journal of Systems and Software* 137 (Mar. 2018), pp. 184–196. DOI: 10.1016/j.jss.2017.11.066. (Visited on 07/01/2020).

[14] Amrita Sharma and Neha Chaudhary. "Linear Regression Model for Agile Software Development Effort Estimation". In: *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* (Dec. 2020). DOI: 10.1109/icraie51050.2020.9358309.