3-1-2009

Exploring Software project characteristics on Scope Creep

Denise Williams
denisew@utm.edu

David Williams

Exploring Software Project Characteristics on Scope Creep

Denise Williams
The University of Tennessee at Martin
denisew@utm.edu

David Williams
The University of Tennessee at Martin
davidw@utm.edu

ABSTRACT

Scope creep is the growth of the scope of a project's requirements beyond those originally intended. Information technology projects, in particular, suffer scope creep. One comment from an IT project manager about this research acknowledged the prevalence of scope creep, noting the paucity of solutions. While many researchers have looked at surveys of various people involved in or with IT projects, the goal of this research is to explore scope creep based on project characteristics: using regression analysis to create a model for scope creep for software projects. The results of the regression analysis for the proposed model will help to better understand both scope creep and also the influence of some project characteristics on scope creep.

Keywords

Software Projects, Scope Creep, Requirements Creep

INTRODUCTION

Information technology projects are plagued by scope creep. It is an ongoing problem that also affects software projects (a subset of IT projects). The persistence of scope creep in IT projects is problematic because it is difficult to complete a project on-time or within budget if the software's functionality continues to expand. It is hard to provide increased functionality without using more resources. Additionally, given the difficulties of evaluating IT projects, including their costs, benefits (potential and realized), and overall success or lack thereof; the potential that scope creep may impact any or all of these dimensions only increases the challenge. If a project's costs are to be estimated, should the estimation include a provision for the expenses that may result from scope creep? If so, how much should be allocated? If one considers the potential benefits of a project, how should scope creep be treated? Does scope creep pull resources or use them in a non-optimal way for firms? These questions are beyond the scope of this research. The questions, however, illustrate the importance of scope creep.

PRIOR RESEARCH

According to the Wideman Comparative Glossary of Common Project Management Terms (v3.1), Scope Creep is defined as "[o]n-going requirements increase without corresponding adjustment of approved cost and schedule allowances. As some projects progress, especially through the definition and development phases, requirements tend to change incrementally, causing the project manager to add to the project's mission or objectives without getting a corresponding increase in the time and budget allowances" (Wideman, 2008). Wideman's definition helps to clarify the term, but the concept is familiar to project managers and IT professionals. Indeed, runaway projects, and the difficulty of their management (or mismanagement) is a central focus of project success generally. Scope Creep is a primary contributor to some of the most notorious project failures (Nelson, 2007). This is particularly true in IT projects, where "the average project experiences about a +25% change in requirements over its lifetime" (Nelson, 2007). Given its negative potential, it comes as no surprise that numerous articles, books, blogs and wiki have explored the concept of Scope Creep.

One reason that scope creep remains so elusive is that it is inextricably linked with the natural evolution of a project's implementation. The reality of a project's implementation can distort its original vision. Projects fail to meet deadlines, eclipse their original budget and frequently change in terms of direction and organizational enthusiasm. It can be difficult to recognize what is scope creep and what is distortion. Perhaps for this reason, focused concerns such as scope creep are sometimes ignored or subsumed into purely financial measures of project success (Irani, et al., 2005). The determination and limitation of prospective project features remains a great challenge (Yadav, et al., 2009; Crowston and Kammerer, 1998).

To avoid scope creep, the original project's scope and the fundamental parameters of that scope must be identified, but the critical parameters of any project can prove difficult to identify or isolate. Researchers continue to attempt to define the relevant critical success factors for project success. Nelson proposes a model of three process-related criteria: (1) Time, (2) Cost, (3) Product; and three outcome-related criteria: (4) Use, (5) Learning, and (6) Value (Nelson, 2005). Alternatively, Keil, Tiwana, and Bush (2002) describe 23 distinct risk factors which might imperil a given project, as identified by a Delphi study of users and project managers. Project managers identified "misunderstanding the requirements" as their second most significant risk factor, "unclear/misunderstood scope/objectives" as their ninth most significant risk factor, "changing scope/objectives" as their tenth most significant risk factor, and "introduction of new technology" as their twelfth most significant risk factor. Users identified "misunderstanding the requirements" as their sixth most significant risk factor, and "changing scope/objectives" as their twelfth most significant risk factor (Keil, et al, 2002). Thus it seems that while scope creep is a recognized problem, its significance in project distortion may be under-represented in the minds of some project managers and in the corporate cultures of some organizations. Given the difficulty in assessing the success of IT projects that the literature suggests (Gable, et al., 2008), this lack of emphasis can be more easily understood.

Failure to identify and properly characterize the relevant factors will inhibit or diminish project success, and may lead to unexpected and undesired investments of money, time, scope, and organizational focus. The problem is particularly profound in IT projects and software development (Keil, et al., 2003; Zmud, 1980), where runaway projects are well-known and sometimes notorious, but have received surprisingly "little attention from information systems researchers." (Keil, et al., 1994). "Although runaway projects are eventually terminated or significantly redirected, anecdotal data suggest that many of these projects are allowed to continue for too long before appropriate action is taken" (Keil, et al., 1994). De-escalation, defined as "the reversal of escalating commitment to failing courses of action, either through project termination or redirection" (Keil and Robey, 1999), has been addressed in the project success literature in only a limited fashion (Mahring, et al., 2008).

Nelson surveyed 13,522 IT projects, finding that nearly two-thirds suffered from limited or total failure (Nelson, 2005). Philbin states that "increased complexity, together with the nonlinear nature of technology-based projects and the tighter connectivity between subsystems, is leading to a greater risk attached to the delivery of projects and programs" (Philbin, 2008). Given the realities of potential project failure, or the mitigated success of bloated projects (Nelson, 2005), an approach to identify scope creep is clearly very important. Monitoring an ongoing project is both challenging and essential (Njaa, 2008). While many variables are certainly involved, some specific factors have been recognized as influencing the evolution and success of a project (Barki, et al., 2005). Among these are the size of the project, and the experience of the project team (McFarlan, 1981). Other recognized factors are technical newness, application newness, and novelty of application (Barki, et al, 2005).

While not explicit, it seems reasonable to assume that certain organizations and industries are more fundamentally likely to possess relevant experience in the development of a given project than other organizations or industries might possess. The relevance of industry has long been recognized. Specifically addressed as to their impact on project definition and success are industry structure, competitive strategy, and industry position (Batiste and Jung, 1984; Rockart, 1979).

While scope creep can sometimes be recognized and minimized, it can also contribute to runaway projects. Two possible explanations for the prevalence of runaway projects can be found in Self-Justification Theory, and Prospect Theory. The former suggests that rationalization may play a part in the continued support for a project despite adverse feedback. The latter suggests persons may be willing to continue to invest resources in a poorly-performing project, because the cost of complete and utter failure seems higher than that of continued investment (Keil, et al, 1994). Moreover, a phenomenon known as the "Mum Effect" describes the reluctance of a project member to discuss ongoing or potential problems (Park, et al., 2008). Given the reality of these dimensions, scope creep seems unlikely to go away. Useful tools to identify the elements most likely to contribute to scope creep remain very important.

PROPOSED SCOPE CREEP MODEL AND METHODOLOGY

While previous authors have looked at IT projects and software projects, this research seeks to better understand the impact of project characteristics on scope creep. The goal of this research is to use regression analysis to explore scope creep in actual projects using self-reported data that belongs to the International Software Benchmarking Standards Group. The authors wish to thank the International Software Benchmarking Standards Group for the use of their data. The data used in the regression analysis will include 370 software projects. These projects range across a variety of industries and types of software projects. The project characteristics include some factors that are discretionary, and others that cannot be changed. For example, an organization can choose to do a project internally or to hire someone to do the project externally. Does this impact scope creep? That will be examined. An organization in one industry (excluding the IT services industry) would only undertake projects for their own organizational goals. For example, a bank would only make use of banking or banking-related IT projects. Does industry influence scope creep? The authors will explore that question also.

The goal of this project is to use regression analysis to identify a model for scope creep given the modeled characteristics of software projects. The proposed model uses scope creep as the dependent variable. In our model, scope creep is operationalized as added functional points. Functional points added to the project are our measure of scope creep for the projects used. The independent variables that are used for the model are: (1) size of the project, (2) requirements analysis, (3) extent of newness or change of the project, and (4) industry. This leads to our first hypothesis: an F-test to test the model overall, testing the null hypothesis that all of the coefficients of the independent variables in our model are equal to zero. (H_0: $B_i = 0$). If this null hypothesis can be rejected at an important or interesting level of significance, then we can explore the related hypotheses about the independent variables.

We hypothesize that size will be a significant variable in our scope creep model. The project success literature supports the proposition that size impacts project success (Nelson and Ravichandran, 2004), and we hypothesize that the regression results will indicate that size is significant to scope creep with a positively signed coefficient. In our model, size is operationalized as Adjusted Function Points in the project. It seems likely that the larger a project is, the harder scope creep will be to control. After all, if we have a large enough project, the marginal change for adding a few function points should seem relatively small. Also, if larger projects are harder to manage (as they should generally be), then it should be harder to stop increasing their scope.

Requirements analysis is the state of system development where services and resources required to support selected objectives are clearly identified (Hoffer, et al., 1999; Nelson, 2007; Yadav, et al., 2009). The project success literature supports the proposition that this identification is crucial to project success, particularly in a turbulent environment (Battin, et al., 2003; Yadav, et al, 2009). Project failure has been attributed to "poor requirements determination" (Nelson, 2007). Mathiassen, et al. reinforces the significance of requirements development by elaborating three Requirement Development Risks: (1) Requirements Identity, (2) Requirements Volatility, and (3) Requirements Complexity (Mathiassen, et al., 2007). Requirements analysis is hypothesized to be a significant variable in our scope creep model. For this model, requirements analysis is measured as the percent of work effort that was used to determine requirements. Our hypothesis is expected to be both significant and negative based on the idea that care in determining requirements and thorough requirements analysis ought to help control scope creep. Alternatively, the determination of requirements might open the door to scope creep.

The newness or change of the project is the classification of a project as an enhancement, a re-development, or a new development. These classifications are hypothesized to be significant because the extent of change or newness should impact scope creep. A project that is an enhancement or re-development would include current knowledge about the project and the related functionality. An organization that undertakes a project to change something that already exists ought to involve people with knowledge about the project area/domain which should result in a better understanding of the needs for the project prior to undertaking the project. A new development project should have more potential for additional functionality over the course of the project and less organizational knowledge about the project area. We hypothesize that new developments will be positively signed and significant. We anticipate that enhancements and re-developments will be negatively signed.

The industry of the firm undertaking a software project is hypothesized to be significant, with variation by given industry. Industries with greater information needs and greater industry infrastructure needs should have more experience with IT projects. The increased experience with IT projects, along with greater IT resources already in place, should result in less potential for scope creep. Industries with less familiarity or sophistication in IT project development are more likely to outsource their project needs, creating further potential for scope creep due to the potential lack of domain knowledge of the

development team (Dibbern, et al., 2008; Gefen, et al., 2008). While Nath concludes that the quality of off-shore project development may be equivalent to local development, the author also notes that more research is needed (Nath, 2008). Relevant domain knowledge is also recognized in the literature as a benefit to project development (He and King, 2008).

In the data we are using, industry is self-reported. For example, finance is an industry that is very information intensive and requires intensive information infrastructure. The authors hypothesize that finance projects will be less likely to suffer scope creep because of the extensive use of IT resources and experience with IT. If this is not the case, an alternate explanation might be that people in industries with greater IT resources may be more inclined to want and ask for more functionalities in projects. Other industries may be more likely to suffer scope creep in their projects. Government entities are widely thought of as inefficient. Many examples exist of government IT projects that have failed – the IRS provides an example as does the Denver Airport (Montealegre and Keil, 2000; Nelson and Ravichandran, 2004). The authors hypothesize that government projects will be significantly positively-signed.

For the reasons stated above, we hypothesize that the independent variables described will be significant in our scope creep model. After applying linear regression to the data set, we will test the hypotheses provided and be able to articulate our model. The answers to this research are of interest not only to other researchers, but also to industry. While not all of the factors are within the control of project decision makers, awareness of the impacts of these factors on scope creep should lead to better management of scope creep.

REFERENCES

1. Barki, H., Rivard, S., and Talbot, J. 1993 "Toward an Assessment of Software Development Risk." *Journal of Management Information Systems* (10:2), pp. 203-225.

2. Batiste, J. and Jung, J. 1984 "Requirements, Needs, and Priorities: A Structured Approach for Determining MIS Project Definition." *MIS Quarterly* (8:4), pp. 215-227.

3. Battin, R., Crocker, R., Kreidler, J., and Subramanian, K. 2001 "Leveraging Resources in Global Software Development." *IEEE Software* (18:2), pp. 70-77.

4. Beneroch, M., Lichtenstein, Y., and Robinson, K. 2006 "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationships." *MIS Quarterly* (30:4), pp. 827-864.

5. Crowston, K., and Kammerer, E. E. 1998 "Coordination and Collective Mind in Software Requirements Development." *IBM Systems Journal* (37:2), pp. 227-241.

6. Dibbern, J., Winkler, J., and Heinzi, A. 2008 "Explaining Variations in Client Extra Costs Between Software Projects Offshored to India." *MIS Quarterly* (32:2), pp. 333-366.

7. Gable, G., Sedera, D., and Chan, T. 2008 "Re-Conceptualizing Information System Success: The IS-Impact Measurement Model." *Journal of the Association for Information Systems* (9:7), pp. 377-408.

8. Gefen, D., Wyss, S., and Lichtenstein, Y. 2008 "Business Familiarity as Risk Mitigation in Software Development Outsourcing Contracts," *MIS Quarterly* (32:3), pp. 531-551.

9. He, J., and King W. 2008 "The Role of User Participation in Information Systems Development: Implications from a Meta-Analysis." *Journal of Management Information Systems* (25:1), pp. 301-331.

10. Hoffer, J., George, J., and Valacich, J. 1999 Modern Systems Analysis and Design, USA: Addison Wesley.

11. Irani, Z., Ghoneim, A., and Love, P. E. D. 2005 "Evaluating Cost Taxonomies for Information Systems Management." *European Journal of Operational Research (*173), pp. 1103-1122.

12. Keil, M., and Robey. D. 1999 "Turning around Troubled Software Projects: An Exploratory Study of the Deescalation of Commitment to Failing Courses of Action." *Journal of Management Information Systems* (15:4), pp. 63-87.

13. Keil, M., Mixon, R., Saarinen, T., and Tuunainen, V. 1994 "Understanding Runaway Information Technology Projects: Results from an International Research Program Based on Escalation Theory." *Journal of Management Information Systems (*11:3), pp. 65-85.

14. Keil, M., Tiwana, A., and Bush, A. 2002 "Reconciling User and Project Manager Perceptions of IT Project Risk: a Delphi Study." *Information Systems Journal (*12), pp. 103-119.

15. Keil, M., Rai, A., Mann, J. C., and Zhang, G. P. 2003 "Why Software Projects Escalate: The Importance of Project Management Constructs." *IEEE Transactions on Engineering Management (*50:3), pp. 251-261.

16. Mahring, M., Mathiassen, L., Keil, M., and Pries-Heje, J. 2008 "Making IT Project De-Escalation Happen: An Exploration into Key Roles." *Journal of the Association for Information Systems* (9:8), pp. 426-496.

17. Mathiassen, L., Tuunan, T., Saarinen, T., and Rossi, M. 2007 "A Contingecy Model for Requirements Development." *Journal of the Association for Information Systems* (8:11), pp. 569-577.

18. McDougal, P. 2006 "Dexterity Required," *Information Week* (June 18), pp. 34-39.

19. McFarlan, F. W. 1981 "Portfolio Approach to Information Systems." *Harvard Business Review (*59:5), pp. 142-150.

20. Montealegre, R., and Keil, M. 2000 “De-escalating Information Technology Projects: Lessons From The Denver International Airport.” *MIS Quarterly* (24:3), pp. 417-447.

21. Nath, D., Sridhar, V., Adya, M., and Malik, A. 2008 “Project Quality of Off-Shore Virtual Teams Engaged in Software Requirements Analysis: An Exploratory Comparative Study.” *Journal of Global Information Management* (16:4), pp. 24-45.

22. Nelson, M. and Ravichandran, T. 2004 “Repeating Failure in Large-scale Government IT Projects: A Taxing Story.” *Proceedings of the Tenth Americas Conference on Information Systems*, New York, New York.

23. Nelson, R. 2005 “Project Retrospectives: Evaluating Project Success, Failure, and Everything In Between.” *MIS Quarterly Executive (*4:3), pp. 361-372.

24. Nelson, R. 2007 “IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices.” *MIS Quarterly Executive* (6:2), pp. 67-78.

25. Njaa, D. 2008 “Project Checkup: Periodic Health Checks Can Guide a Systems Implementation Project along the Narrow Path to Success.” *Internal Auditor* (65:4), pp. 31-33.

26. Park, C., Im, G., and Keil, M. 2008 “Overcoming the Mum Effect in IT Project Reporting: Impacts of Fault Responsibility and Time Urgency.” *Journal of the Association for Information Systems* (9:7), pp. 409-431.

27. Philbin, S. P. 2008 “Managing Complex Technology Projects: Systems Methodologies Help Meet the Formidable Challenge of Managing Increasingly Complex Engineering Systems so They Deliver the Specified Requirements.” *Research-Technology Management* (51:2), pp. 32-39.

28. Rockart, J. F. 1979 “Chief Executives Define their Own Data Needs.” *Harvard Business Review (*57:2), pp. 81-93.

29. Wideman, R. M. 2008 "Max's Project Management Wisdom." December 30, 2008. http://www.maxwideman.com.

30. Yadav, V., Adya, M., Sridhar, V., and Nath, D. 2009 “Flexible Global Software Development GSD: Antecedents of Success in Requirements Analysis,” *Journal of Global Information Management*, 17.1, 1-30.

31. Zmud, R. W. 1980 “Management of Large Software Development Efforts.” *MIS Quarterly (*4:2), pp. 45-55.