SUNWAY

INT'L BUSINESS SCHOOL

Programme Name: BCS

Course Code: CSC1017

Course Name: SYSTEM ANALYSIS AND DESIGN

Assignment / Lab Sheet: Research Paper summary

Date of Submission: 21st January 2022

Submitted By:

Student Name: Unisha Tamang

IUKL ID: 042101900085

Semester: 2nd semester

Intake: March

Submitted To:

Faculty Name: Yogendra Bahadur Mahata

Department: BCS

Impact of incorrect and new requirements on Waterfall Software Project outcomes

Abstract

This study evaluates the effects of change requests caused by requirement flaws on the results of Waterfall-based software development initiatives. Incorrect requirements, incomplete requirements, and novel requirements are the three categories of requirement faults investigated. The entire effort consumed and software flaws introduced during software development are used to calculate the outcomes.

Introduction

While there is a lot of active study on how to improve requirement procedures, there isn't much on the effects of the many sorts of requirement flaws that are identified during development. It was found that the various sorts of requirement defects routinely monitored by software development firms have distinct effects on project outcomes, and project managers can benefit from considering these differences as they manage projects. Hypotheses about these consequences are investigated using a data set of 49 software projects obtained from a software development organization at the highest degree of software development process maturity (CMMI level 5).

Prior research on the effects of software requirements on project success may be divided into two categories. The first stream examines the overall effects of requirements engineering on software project results, as well as the amount to which improved management may enhance these outcomes. The second stream examines the requirements engineering process in greater depth, focusing on the impact of specific components of requirement quality on software project results.

Research Model and Hypotheses

Our study model is based on the assumption that requirements engineering activity necessitates collaboration among members of a software project team in order to solve a specific challenge. This supports the knowledge-based approach to requirements collection and execution that has been established. To motivate the usage of this approach, we first describe the requirements engineering environment. Software projects are often executed by teams organized particularly for the purpose of the project. Available individuals are chosen depending on the specific needs of a given project, and most software project team members have little prior experience working together in a project team. These teams are entrusted with comprehending customer requirements and delivering the best technological solutions feasible, given financial limits, present IT infrastructure, and expected future requirements.

Whenever the development team works with the requirements analysis, the requirements are not the specifications themselves, but rather a representation of them. A requirement is a user need, but a requirement specification, which drives development, is just a model of the need. The seeming unpredictability in needs is usually caused by incorrect depiction. Accurate representations can help with reuse of use cases and requirement specifications.

Incorrect Requirements

Change requests resulting from faulty requirements may be classified into two types: basic and difficult. In the most basic situation, the resolution may be innocuous. Misinterpretations may be restricted to a few stakeholders, and misread requirement design factors may interact with few requirement design parameters. Teams that detect and address a greater proportion of erroneous criteria should obtain better belief agreement, leading to higher quality. As a result, because an increase in the number of faulty requirements handled might suggest a more shared understanding of customer demands, hypothesis is:

H1: The resolution of incorrect requirements is positively related to software quality

H2: The resolution of change requests due to incorrect requirements is positively related

With the number of new requirements

Incomplete Requirements

Even if no requirement design parameters are misinterpreted, there may be missing parameters in the requirement representation. These missing attributes are most frequently the result of unarticulated user expectations and unrecorded analyst needs. Based on this, they hypothesize the following to account for the change management process and its associated coordination benefits, while acknowledging the possibility that resolving incomplete requirements presents fewer opportunities for creating a shared understanding than resolving incorrect requirements:

H3: The resolution of change requests due to incomplete requirements is positively associated with software quality.

New Requirements

A rise in new needs as a result of correcting inaccurate requirements should indicate changes in mutual perceptions and expectations, resulting in better project outcomes. New requirements that develop as a result of changing company needs or user expectations, on the other hand, are likely to have a negative influence on project outcomes.

H4: The resolution of change requests due to new requirements is negatively associated with software quality

H5: The resolution of change requests due to new requirements increases software project effort

Methodology

Archival data from one of the world's largest software businesses was utilized to evaluate these predictions. To identify the impacts hypothesized in their natural environment of software development for business applications, archival data from a commercial software development firm was deemed most appropriate for this investigation. Project-related data for a sample of 53 projects were acquired from the research site's internal systems at their request for this study. One project was duplicated, while data for two other projects was missing a large number of elements. As a result, these three initiatives were rejected. We canceled one additional project due to data quality difficulties. As a consequence, a total of 49 projects were included in the final data set.

Agile procedures that leverage iterative development processes are becoming increasingly popular for mitigating project risks. However, a variant of the Waterfall technique with separate stages and well-documented baseline requirement specifications is still extensively employed, particularly in contractual offshore software development projects. All projects in this research employed the Waterfall development process, which increased consistency in development environment characteristics.

Out of the 49 projects, 48 were categorized as development projects, with the remaining project classified as re-engineering. The sample's average project size was 736 function points (FPs), and the average project effort was 8014 person-hours.

Data Analysis

As previously stated, we have a valid data sample of 49 projects to analyze. During the model estimating phase, we discovered some missing data in four observations, reducing the data sample to 45 analyses for regression estimation. This study's sample size compares favorably to those utilized in previous published investigations, including 37 in (Agrawal and Chari 2007), 15 in (Kemerer 1987), 15 in (Baik 2002), 15 in (Mukhopadhyay et al. 1992), 43 in (Krishnan et al. 2000), 30 in (Harter et al. 2000), and 42 in (Krishnan et al. 2000). (Ramasubbu et al. 2004). SAS 9.3 was used to run all tests.

Validity Threats

As an empirical study, the research's validity must be evaluated. Internal validity, construct validity, and external validity are the key validity. Internal validity is defined as the absence of alternative explanations for observed effects on dependent variables. The appropriateness of the mapping between the measured effects and the theoretical notions described is referred to as construct validity. External validity relates to the results' generalizability outside of the organization under study.

Limitation

There are several significant limitations to this study. The size of the dataset and the fact that the data was acquired from projects conducted by a CMMI Level 5 company are the key constraints of this research. The professional effort necessary to recover data from archive records limits the amount of the dataset, which is an intrinsic constraint of the archival technique in software project management literature. The dataset size was agreed upon before to the investigation and was restricted by the amount of professional work the organization was able to put in as part of the study.

Future Research

The findings and limitations point to a variety of potential study options. The most important are those with the identification of the sources of requirement deficiencies and omissions. Our projects' requirements were created by highly experienced project teams and validated by highly competent quality assurance teams. Our research location has a big number of large clients with well-established software procurement procedures.

Conclusion

The consistent viewpoint is used in this research to construct a model for the effects of various types of requirements flaws on software project results. We discovered that resolving inaccurate needs is related to a drop in the number of faults provided, but also to an increase in the number of new requirements developed. New requirements are connected with an increase in the number of faults provided and, possibly, an increase in the amount of effort necessary to finish the project.

Reference

Chari, K., & Agrawal, M. (2017). *Impact of incorrect and new requirements on waterfall software project outcomes*. https://doi.org/10.1007/s10664-017-9506-4