Study of Factors that Induce Software Project Overrun Time

Santhosh Kumar Kamuni

St. Cloud State University

Follow this and additional works at: https://repository.stcloudstate.edu/mme_etds

Recommended Citation
https://repository.stcloudstate.edu/mme_etds/10
Study of Factors that Induce Software Project Overrun Time

by

Santhosh Kumar Kamuni

A Starred Paper
Submitted to the Graduate Faculty of
St. Cloud State University
in Partial Fulfillment of the Requirements
for the Degree
Master of Engineering Management

November, 2015

Starred Paper Committee:
Hiral Shah, Chairperson
Ben Baliga
Balasubamanian Kasi
Abstract

Software is created with programming languages and related utilities, which may come in several forms. For a software project to be successful it is essential that the project is completed on time and within the estimated budget, with all features and functions as originally specified. However due to various reasons these software projects fail some even get cancelled before they even get completed. Because of this, the projects can be categorized into Successful projects, challenged projects and Failed projects. The failure is caused mainly because of cost overrun or time overrun or because the product was not delivered with all the requirements as per initial specification. The cost of these failures and overruns are just the tip of the proverbial iceberg. The lost opportunity costs are not measurable, but could easily be in the trillions of dollars. Hence it is important that detailed work be done on these to overcome the huge costs they incur.

This research work is focused on analyzing the factors that cause software project time overruns and to provide recommendation to overcome or mitigate the effect of some of the identified factors on software project.

The project employs systematic, formal and descriptive research techniques. This study is based on the data collected through structured questionnaire and in-depth, unstructured and informal interview with key personnel. The sampling technique used is snowball sample technique. Since the research topic is highly qualitative in nature, we are prompted to use simple percentages so as to make the data more succinct.
Acknowledgement

I express my immense gratitude to Professor Dr. Hiral Shah, Associate Professor, St. Cloud State University. She has been my mentor and guide and her continuous encouragement and valuable suggestions helped me at every stage of this project.

I am grateful to Dr. Ben Baliga, Director of St. Cloud State University for his comments and encouragement at various phases of this project.

I also extend my gratitude to all the respondents from various software companies across India and USA for their prompt and timely help and guidance in completing the project.
# Table of Contents

List of Figures ......................................................................................................................... 6

Chapter

1. Introduction .......................................................................................................................... 8
   Problem Statement ................................................................................................................... 9
   Significance of the Problem .................................................................................................... 9
   Objective of the Project .......................................................................................................... 10
   Project Questions .................................................................................................................. 10
   Limitations of the Study ........................................................................................................ 11
   Definition of Terms .............................................................................................................. 11
   Summary ................................................................................................................................ 13

2. Review of Literature ............................................................................................................. 14
   Introduction ............................................................................................................................ 14
   Background .............................................................................................................................. 14
   Purpose of Literature Review ................................................................................................. 16
   Literature Related to the Methodology .................................................................................. 16
   Summary ................................................................................................................................ 20

3. Methodology ......................................................................................................................... 21
   Introduction ............................................................................................................................ 21
   Design of the Study ............................................................................................................... 21
   Budget ...................................................................................................................................... 23
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Timeline</td>
<td>23</td>
</tr>
<tr>
<td>Summary</td>
<td>24</td>
</tr>
<tr>
<td>4. Data Presentation and Analysis</td>
<td>25</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>50</td>
</tr>
<tr>
<td>Summary</td>
<td>52</td>
</tr>
<tr>
<td>5. Results, Conclusion, and Recommendations</td>
<td>54</td>
</tr>
<tr>
<td>Results</td>
<td>54</td>
</tr>
<tr>
<td>Conclusion</td>
<td>58</td>
</tr>
<tr>
<td>Recommendations</td>
<td>58</td>
</tr>
<tr>
<td>References</td>
<td>62</td>
</tr>
<tr>
<td>Appendix</td>
<td>63</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Chaos Report</td>
<td>19</td>
</tr>
<tr>
<td>2. Number of Projects Undertaken</td>
<td>25</td>
</tr>
<tr>
<td>3. Time Spent on Typical Project</td>
<td>26</td>
</tr>
<tr>
<td>4. Team Size</td>
<td>27</td>
</tr>
<tr>
<td>5. Project Completion</td>
<td>28</td>
</tr>
<tr>
<td>6. Product Delivered with All the Initial Requirements</td>
<td>29</td>
</tr>
<tr>
<td>7. Proportion of Features Added Later</td>
<td>30</td>
</tr>
<tr>
<td>8. Managing Large Teams</td>
<td>31</td>
</tr>
<tr>
<td>9. Problems Faced in Managing Teams</td>
<td>32</td>
</tr>
<tr>
<td>10. Ways of Overcoming Team Management Problem</td>
<td>33</td>
</tr>
<tr>
<td>11. Increase in Delivery Time</td>
<td>35</td>
</tr>
<tr>
<td>12. Usage of Time Tracking Tools</td>
<td>36</td>
</tr>
<tr>
<td>13. Other Measures</td>
<td>37</td>
</tr>
<tr>
<td>14. Lack of User Input</td>
<td>38</td>
</tr>
<tr>
<td>15. Incomplete Requirements and Specifications</td>
<td>39</td>
</tr>
<tr>
<td>16. Changing Requirements and Specifications</td>
<td>40</td>
</tr>
<tr>
<td>17. Lack of Executive Support</td>
<td>41</td>
</tr>
<tr>
<td>18. Technology Incompetence</td>
<td>42</td>
</tr>
<tr>
<td>19. Lack of Resources</td>
<td>43</td>
</tr>
<tr>
<td>20. Unrealistic Expectations</td>
<td>44</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>21. Unrealistic Objectives</td>
<td>45</td>
</tr>
<tr>
<td>22. Unrealistic Time Frames</td>
<td>46</td>
</tr>
<tr>
<td>23. New Technology</td>
<td>47</td>
</tr>
<tr>
<td>24. Lack of Planning</td>
<td>48</td>
</tr>
<tr>
<td>25. Complicated Standards</td>
<td>29</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Software is created with programming languages and related utilities, which are available in many forms. For a software project to be successful it is important that the project is accomplished on time and within budget, with all requirements and functions originally specified. However, due to numerous reasons these software projects fail some even get cancelled before they even get completed. As a result of this, the projects can be categorized into Successful projects, Failed projects and challenged projects. The failure is mainly caused because of cost or time overrun, or because the product was not delivered with all the requirements as per initial specification.

This analysis of work is concentrated on analyzing the factors that induce software project time overruns and to provide suitable recommendations to overcome or mitigate the effect of some of the identified factors on software project. This study is predicted on the information and data collected through structured questionnaire and informal interview with personnel who are involved in the project management. The sampling technique which I used is snowball sample technique.

For a software project to be accomplished it is very important that the project is completed on time and with the allocated budget, with all features, functions and requirements specified originally. The project success or failure is influenced by some factors. Some of them are binary, some are quantitative, and some are qualitative. Also outside factors influenced the project success or failure. The study established that the explanation for many of the project failures was not lack of funds or
technology, but most failed due to lack of skillful project management and higher authority support.

**Problem Statement**

Viewing this fact, this analysis work is aimed to investigate the factors that cause software project time overruns and to suggest recommendations to overcome or reduce the effect of some of the identified factors on software project.

However, we should understand that the product success is always not equal to project success. One project might be considered as a failed project by the management and one project may be regarded as an unsuccessful project by management. It would be many months late, and over-budget, and with some initial quality issues. But the product could be successful with customers, and will be key moneymaker for the company. The inverse could be seen a number of times—the project may be thought to be successful, delivered in acceptable time and cost, etc., however, the product of the project might not notice the advantages. This analysis work is aimed to analyze the project success or failure and not product success or failure.

**Significance of the Problem**

As per the Standish Group’s study of 2009, only 34% of the software projects succeeded, 44% was challenged, and 22% failed. However that was the situation 5 years back. It is essential to search out the situation prevailing nowadays.

The importance of this study relies on the amount of money that software companies allocate on failed projects. The cost of these failures and overruns than
estimated are huge. These costs are not measurable, but could easily be estimated in trillions of dollars. Moreover this kind of failures and overruns hinder the image of the company in today’s competitive market and decreases the level of customer satisfaction. Hence it is very essential to seek out the success rate of software projects in today’s date and establish how much cost and time overruns are occurring even today and what may be done to forestall or minimize such overruns.

However as a result of time and resource constraints, this study is concentrated primarily on analyzing the varied reasons that caused software project time overruns and suggested recommendations to overcome or mitigate some of these reasons.

**Objective of the Project**

Main objectives of the study:

1) To study various reasons of software project time overruns that cause project failures.

2) To provide recommendations in order to overcome known failure factors.

**Project Questions**

After successfully completing the analysis, explanations on the following topics will be obtained.

1) What factors influence a software project?

2) What are the time overruns of failed projects?

3) How do we mitigate these failures?
4) Was agile methodology effective in completing the software development project on time and within the budget?

Limitations of the Study

Research investigation is surrounded by time and resource constraints. Research analysis is limited to respondents who are software professionals in United States.

Definition of Terms

The following are the key terms discussed in the project:

1. **Software development methodology** is a framework used in software engineering structure, plan, and control of the software development process.

2. **Project management methodologies** are the techniques used to initiate, plan and execute projects successfully. Specific project management methodologies can be used for different types of industries and projects.

3. **Agile methodology** is widely used best practice for modeling and documenting software systems. This methodology is very adaptable and flexible than traditional software development methods, making it a better fit in the environment where requirements keep changing at each stage (Holler, 2010).

4. **Scrum methodology** is a widely used agile software development technique. It suits a wide range of software development environments. It focuses on communication and collaboration, delivering functioning
software at short intervals, and adapts to changing software requirements (Wutherland, 2004).

5. **Software development life cycle (SDLC)** is the process of developing or modifying software systems by following software development methodologies. People, process and technology play a vital role in success of the software development lifecycle (Pendharkara, Rodgerb, & Subramanian, 2008).

6. **Best practices** are a set of approaches to **software development** which results in developing most efficient software in a productive manner within the budget. A combination of best practices can be used to best fit a specific software development environment. Different organizations follow different best practices to fit their projects and teams (Cooper & Schindler, n.d.).

Some of them are:

- Iterative development.
- Test driven development.
- Quality testing.
- Requirement change management.
- Following standard coding practices.
- Software version control.

**Project management best practices** remind that on successful initiation, planning, execution, monitoring, controlling and closing out projects, will
illustrate better results. Best practices represent the practical application of the concepts, processes, and tools defined in the project.

Summary

This chapter emphasizes the importance of the above mentioned problem and proposed solution using a step by step evaluation methodology that would be discussed in the coming chapters.
Chapter 2: Review of Literature

Introduction

This chapter explains the background related to project and research area where it is conducted. Literature related to project problem and methodologies used to conduct the project are discussed in detail. This chapter will provide the readers a complete and broader aspect of the project.

Background

A project can be defined as a plan or proposal of a scheme or an undertaking requiring concerted effort.

Softwares are the programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation. A series of instructions that perform a particular task is called a "program." The two major categories of software are "system software" and "application software." System software is made up of control programs such as the operating system and database management system (DBMS). Application software is any program that processes data for the user (inventory, payroll, spreadsheet, word processor, etc.). Software tells the hardware how to process the data (Whitten, Bentley, & Dittman, n.d.).

Software can also be defined as written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory.
Unlike a building, software can’t be seen, touched, felt or visualized and it is hard for the layman to get a conceptual grip of its size or cost or how long it might take to build.

Hence software projects can be defined as an undertaking to develop computer programs routines, and symbolic languages that control the functioning of the hardware and direct its operation.

**Software creation.** Software is created with programming languages and related utilities, which may come in several forms:

- Single programs like script interpreters, packages containing a compiler, linker, and other tools; and
- Large suites (often called Integrated Development Environments) that include editors, debuggers, and other tools for multiple languages.

Software creation requires concerted effort from the software development teams. A software project team ideally consists of a team leader, software developers and software testers.

However for a software project to be successful it is essential that the project is completed on time and on budget, with all features and functions originally specified. There are a couple of metrics for project success/failure. Some are binary, some are quantitative, and some are qualitative. Many may be influenced by outside factors. Some are not direct indicators, but probably have a strong correlation. Some of them are:
1. Actual development time vs. projected development time.
2. Actual cost vs. projected cost
3. Project deployed/introduced, or cancelled?
4. Revenue, profit, or productivity gains realized, vs projected
5. Customer/user satisfaction
6. Effect on reputation of company/department/project team
7. Were project staff rewarded (promoted, given bonuses), punished (sacked, demoted, given pay cuts), or neither? (Schiesser & Hall, n.d.)

**Purpose of Literature Review**

Literature is reviewed to analyze the historical and present situation of software projects and as a whole to interpret the software project management. Over and above, the purpose of the literature review is to understand in totality the foundation of the research problem, understand the data that has been gathered in the field of study and to make new findings on the problem statement. The abstracts in the following pages are some of the key literature reviewed in this project.

**Literature Related to the Methodology**

These have been documented from various sources in lieu with reasons behind software time overruns. Information has been gathered from various sources extending from Internet, magazines, journals, and textbooks to articles. This analysis has thrown light on the articles and research that have been written and published on software project time overruns. The following are the conclusions drawn from the articles used for literature review:
**How to be agile:** This article concerns with various factors which lead to project success. Some of the factors which influence the project success and to be Agile are:

**Slash the budget:** Small budgets force developers to focus only on the essentials. Small budgets also make it easier to kill failing projects.

**If it doesn’t work, kill it:** Right decision at the right time is an imperative trait for a successful project. Appropriate decisions are made by bringing marketing, program, project management and IT executives together at the beginning of the project and evaluate its progress by setting up timelines. If the project seems going out of way kill it. Killing it in the initial stages, avoid unnecessary time and money over runs.

**Keep requirements to a minimum:** Don’t start with everything you want the software to do, start only with the minimum requirements which absolutely needed to be done and don’t worry about writing all your needs down, because requirements keep changing from time to time.

**Build on success, not hope:** Segregate the project work into multiple milestones and implement stringent deadlines. Have your business stakeholders test and approve these milestones periodically. This will ensure steady timelines and maintain quality of the project. This is an agile concept’s most radical departure from traditional development.
Keep your development teams small: Fewer the developers, the better it is. Developers should team up and code. Agile Development proponents swear that team coding is much more efficient and produces stronger code than individual effort.

Assign non-IT executives: Non-IT executives should coordinate with technical project managers, test iterations to make sure they meet requirements, and act as liaison between executives and IT.

Project management: Factors for project success: Compared to the projects in 2009, the one’s in 2013 are more successful in terms of maintaining budget, timely completion and with specific requirements. The article further goes and explains about 10 factors that contribute to a projects success. While all the projects do not require all of the 10 factors to be successful, more the factors present in the project strategy, the higher is the chance of project success level. The 10 factors are firm basic requirements, clear business objectives, reliable estimates, standard software infrastructure, formal methodology, experienced project manager, executive support, minimized scope, and others.

Failure statistics: The project success rates were least for the large software companies when compared to that of the small and medium software companies. Project restarts is the major reason for the software project failures that cause both cost and time overruns. Again large companies have more cost and time overruns when compared with the medium and small companies.

Standish categorizes software projects into three resolution types:
Successful: The project which is completed on time and within the allocated budget, with all features and functions originally specified.

Challenged: The project is completed, but with budget and time overrun, with fewer features and functions than initially specified.

Failed: The project is canceled or aborted before completion, or never implemented.

The Standish Group partitioned the large, medium and small companies based on the results obtained. A large company is any company with greater than $500 million dollars in revenue per year, a medium company is defined as having $200 million to $500 million in yearly revenue, and a small company is from $100 million to $200 million.

Figure 1 shows the percentage of success rates for small software development projects.
Summary

Only 9% of the projects in large companies, 16.2% of projects in medium companies and 28% of projects in small companies were successful. Sixty-one and five tenths percent of all large company projects were challenged when compared to 46.7% for medium companies and 50.4% for small companies. Most of the failed projects are with the medium companies which is 37.1% when compared to large companies and small companies which were 29.5% and 21.6%, respectively.
Chapter 3: Methodology

Introduction

This chapter discusses the methodology used to conduct the project. The most suitable research method was applied to the project. The reason behind selecting this research method are explained. The framework applied to the project and data collection processes are discussed in detail. Various tools and technologies used to analyze the data collected are also discussed. The budget and timeline involved to conduct this entire project are also specified.

Design of the Study

1. **Scope of the study.** This study is restricted to only software projects. It does not include other types of projects.

2. **Type of research.** The project employees qualitative research which abides to systematic, formal and descriptive techniques. The study is based on data collected from structured questionnaire and informal interviews with key personnel.

3. **Data collection.** Data consists of primary and secondary sources. Primary data is sourced from Software project leads, team leads, developers and testers with more than three years of experience.

   A structured questionnaire is developed and administered to generate the primary data.

   Secondary data is sourced from information provided by various journals, reference books and internet. Internet has been a major secondary source for the extraction of the expert’s opinion.
4. **Sampling technique.** Snowball sampling (Patton, 1990) and stratified sample technique have been used. The respondents for the study have been selected based on experience and expertise. Contacts were established using these respondents.

5. **Sample populations.** The sample populations for the purpose of our study are only software professionals.

6. **Sampling frame.** The sampling frame for the study comprises of software professionals in United States of America who are currently employed in the software companies with experience of 32 years or more.

7. **Sample size.** The composition of the study sample consisted of Software professionals with three or more than 3 years of experience. The respondents I approached are 50 in number.

8. **Sample descriptions.** Respondents have been selected from across a cross section of software product and services.

9. **Instrumentation techniques.** Structured Questionnaire—The primary data analysis has been conducted through structured Questionnaires, which were sent to the respondents who were software professionals. Set of Questionnaire has been built to identify the factors which cause project time overruns.

10. **Tools for data analysis.** Since the research topic is highly qualitative in nature, I preferred to use simple percentages so as to enable the data to be more concise and susceptible for easy interpretation and understanding. Simple treatment
of data will be more useful and accessible in drawing inferences from the data obtained.

11. **Limitations of the study.** Research investigation is surrounded with time and resource constraints. Research analysis is limited to respondents who were software professionals in United States of America and thus the limitation of generalization becomes obvious.

**Budget**

No budget was involved to conduct this project. Research was primarily based on personal interviews conducted within the development team. So the exploratory nature of the project made the research, data collection and analysis very simple to evaluate the Scrum methodology and accomplish the project objectives.

**Project Timeline**

The following is the anticipated timeline to complete this project:

**Timeline.**

<table>
<thead>
<tr>
<th>OVERVIEW PROJECT SCHEDULE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MILESTONE</strong></td>
<td><strong>DATE</strong></td>
</tr>
<tr>
<td>Project Initiation</td>
<td>08/25/2014</td>
</tr>
<tr>
<td>Project Proposal Write-up</td>
<td>09/20/2014</td>
</tr>
<tr>
<td>Analyzing the Questionnaire</td>
<td>11/15/2014</td>
</tr>
<tr>
<td>Data Analysis and Interpretation</td>
<td>01/05/2015</td>
</tr>
<tr>
<td>Evaluating the Agile Methodology</td>
<td>02/15/2015</td>
</tr>
<tr>
<td>Recommendations</td>
<td>03/05/2015</td>
</tr>
<tr>
<td>Project Closure</td>
<td>03/25/2015</td>
</tr>
</tbody>
</table>
Summary

This typically illicit the qualitative methodology that was used to address the central problem that drives the cost of the project. It also talks about the basis on which the data was collected from the development team and evaluation of the efficiency using qualitative data and Scrum methodology. It also covers the cost estimation and budgeting of the project and timeline taken to accomplish this project.
Chapter 4: Data Presentation and Analysis

After collaborating with respondents, it was analyzed about each respondent’s professional things like experience, number of projects they had worked through, the time taken to complete a project, team size etc., and their response about the factors leading for project time overrun also recorded. The analyzed data from the predefined methodology is clearly illustrated below.

1. **Average number of projects undertaken:**

![Figure 2: Number of Projects Undertaken](image)

Figure 2 depicts that as the experience level of all respondents is not same, average number of projects undertaken by the respondents is 7. The number of projects are ranged between 1 and 40.
2. Average time spent by the respondents on a typical project:

Figure 3 shows that on interpretation, the average time to complete a project was 10 person months spent by the respondents. The time taken ranged between 3 and 36 person months.
3. Average team size of the respondents:

Figure 4: Team Size

Figure 4 shows that the team size of the respondents is varied between 2 and 35 members with an average of 9 members in a team.
4. Project completion:

Figure 5: Project Completion

Figure 5 shows that in response to the completion of project, 65% of the respondents reported that projects mostly completed on time. Twelve percent reported that the projects always completed on time while 6% reported that the projects were never completed on time.
5. Product delivered with all the initial requirements:

Figure 6 shows that on the aspect of product delivery, majority of the respondents agree that the products they delivered met all the requirements as per the initial specifications. 63% said yes to the query while 35% negated the query and 1 respondent did not respond to the query.
6. Proportion of features added later:

Figure 7 shows that 63% of the respondents said that 10-30% of the features were added later to the product. Twenty-nine percent said that less than 10% of features were added later while 8% said that 30-50% of the features were added later. None of the respondents said that more than 50% of the features were added later.
7. Managing large teams:

Figure 8: Managing Large Teams

Figure 8 shows that majority of the respondents claimed managing large a team is “Not so easy”. Only 8% consider it “Easy” and 4% consider it “Very difficult”.
8. Problems faced in managing teams:

Figure 9: Problems Faced in Managing Teams

Figure 9 shows that 44% of the respondents found communication as the major problem in handling teams. The next problem faced is lack of experience of the team members followed by friction between members and lack of management support.

The other problems identified by the respondents were:

1. People not taking end-to-end ownership.
2. Making them believe in what they are doing and explain them the need of following some processes.
3. Consistency in Competence and Commitment of the team members.
4. Skill of every team members is not at same level, so many times some are more overloaded than others.
5. Team members not co-located (global team), Cultural differences.
6. Co-ordination of various interfaces among various components.
7. Co-ordination and resource sharing.
8. Timely reporting of issues in the project.
10. Managements support to set realistic targets and convince customer of the same.
11. Member’s personal problems, immaturity, attitude, attrition.

9. Ways of overcoming team management problem:

Figure 10 shows that in response to the ways of overcoming management problems, 31% of the respondents opted for more training courses as a viable option to overcome the problems faced in managing teams while 29% opted for more
project reviews and 19% opted for smaller teams. Twenty-one percent of the respondents provided other solutions to overcome the problems faced in managing teams. This include the following:

1. Motivation and making sure that each employee blends with the organizational goals and the processes it follows.
2. Groom team for Company and also manager style.
3. Define the processes very clearly during the planning stages and keep the team updated about the same.
4. Explain the significance of the various steps and processes followed in the project to the project team. This will help the team members answer the question: “Why are we doing some specific activities?”
5. Management Support on Administrative & Infrastructure.
6. More time should be given in design and requirements phase.
7. Meetings, Constructive feedback, non-penalizing reactions on issues reported, encourage honesty.
8. Leads & Project Managers undergo soft skills training and develop strategies for excellence in the changing environment using innovation, displaying creative leadership and building stronger teams.
9. Managements support to set realistic targets and convince customer of the same.
10. To have back up plans ready always by anticipating problem areas.
11. Team building exercises, Team lunches to reduce friction between team members and improve communication between team members.

12. Have some senior folks, effective training and mentoring.

10. Increase in delivery time:

![Pie chart showing delivery time exceeded by](image)

**Figure 11: Increase in Delivery Time**

Figure 11 shows that none of the projects had time overruns of more than 3 months duration. Fifty-two percent of the projects overshot the time by less than 1 month and 48% overshot time by 1-3 months.
11. Usage of time tracking tools:

Figure 12: Usage of Time Tracking Tools

Figure 12 shows that 67% of the respondents used time tracking tools while the rest didn't use any. Most of the tools used were in-house developed and the others consisted mainly of M.S Office Project, Excel Sheets, Lotus Notes, Rational Portfolio Manager, CONCERTO and Outlook Calendar.
12. Other measures employed to ensure on time delivery of project:

Figure 13: Other Measures

Figure 13 shows that beside using time tracking tools, 31% of the respondents used Checks and Reviews, 30% used Regular meeting, 25% used Careful planning, 8% Allocated extra resources, 1% reduced functionality to ensure that the projects are delivered on time.

Other measures employed include the following:

1. Allocate rightly skilled resources.
2. Identify and track Milestones.
3. Proactively identify the possible risks and take the necessary mitigation steps.
4. Good next-level leadership, empowerment, open-communication.
5. Buffer management.
6. Proper estimation before hand.

7. Push back on some non-priority requirement (Negotiate soft with customer).

8. Proactive Risk identification and mitigation/management.


10. Proper allocation of work.

11. Backup resources.

13. **Factors responsible for project time overrun:** The following charts depict the respondent’s attitude towards various factors that can be held responsible for project time overrun. Each factor is measured over a scale of four: Certainly Responsible, Responsible to a Great Extent, Responsible to Some Extent and Not Responsible.

**1. Lack of User input**

![Lack of user input chart](image)

*Figure 14: Lack of User Input*
Figure 14 shows that lack of user input, marking this as a point for the project time overrun, 37% of the respondents feel that lack of user input is to some extent responsible for project time overrun while 36% feel that this factor is certainly responsible. Only 4% feel that lack of user input is not responsible for project time over run.

2. **Incomplete requirements and specifications:**

![Incomplete requirements and specifications](image)

Figure 15: Incomplete Requirements and Specifications

Figure 15 shows that incomplete requirements and specifications are considered by the majority of the respondents, i.e., 48% of the total, as certainly responsible for project time overruns followed by 31% of respondents who feel that this factor is responsible to a great extent for project time overrun. Seventeen percent and 4% voted for responsible to some extent and not responsible respectively. Hence this factor turns out to be a major area of concern.
3. Changing requirements and specifications:

Figure 16 shows that changing requirements and specifications, rated with 46% of the respondents consider it as responsible to a great extent for project time overrun and 31% feel that this factor is certainly responsible. Only 4% feel that this factor is not responsible for project time overrun while 19% feel that it is responsible to some extent.

Though intermediate project updates leads to changing requirements and specification, overall this factor can also be considered as a major area of concern.
4. Lack of executive support:

Figure 17: Lack of Executive Support

Figure 17 shows that lack of executive support is considered by 52% of respondents as responsible to some extent for project time overruns and 29% feel that it is not responsible. Only 13% and 6% feel that it is responsible to great extent and certainly responsible, respectively.
5. Technology incompetence:

Figure 18: Technology Incompetence

Figure 18 shows that 41% of respondents feel that technology incompetence of the team members is to some extent responsible for project time overrun, 31% feel that it is responsible to a great extent, 15% feel that it is responsible certainly and 13% feel that it is not responsible for project time over run.
6. Lack of resources:

Figure 19: Lack of Resources

Figure 19 shows that 33% of respondents feel that lack of resources is to great extent responsible for project time overrun, 29% feel that it is certainly responsible, 21% feel that it is responsible to some extent and 17% feel that it is not responsible for project time over run.

Again, this factor can be considered as an important factor to take care of to ensure on time delivery of projects.
7. Unrealistic expectations:

Figure 20 shows that unrealistic expectations, as the factor leads to project time overrun, rated with 34% of the respondents, they feel that this factor is certainly responsible for project time overrun. While 25% feel that it is responsible to a great extent. Only 10% feel that it is not responsible and 31% feel that it is responsible to some extent.

Overall, since majority of respondents feel that it is certainly responsible, this factor is critical to avoid project time overrun.
8. Unrealistic objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:

Figure 21: Unrealistic Objectives

Figure 21 shows that when discussed about the effect of unclear objectives on a software project, 38% of the respondents feel that unclear objectives are responsible to some extent in causing project time overrun. Twenty-seven percent opted for “Certainly”, 25% opted for “To a great extent” and 10% opted for “Not responsible”.

unrealistic
figures
objects
8.

objectives:
9. Unrealistic time frames:

Figure 22: Unrealistic Time Frames

Figure 22 shows that majority of the respondents feel that fixing an unrealistic time frame is responsible for project time overruns. Forty-two percent voted for responsible “To a great extent”, 29% voted for “Certainly” responsible, 19% voted for responsible “To some extent” and only 10% voted for “Not responsible”.
10. New technology:

Figure 23 shows that in the software environment as technology update is quite obvious and vigorous; half of the respondents feel that this factor is to some extent responsible while 19% feel that it is not responsible. Only 25% and 6% feel that it is responsible to great extent and certainly responsible, respectively.
11. Lack of planning:

Figure 24: Lack of Planning

Figure 24 shows that 36% feel that lack of planning is certainly responsible for project time overrun, 21% feel that it is responsible to a great extent, 35% feel that it is responsible to some extent and only 8% feel that this factor is not responsible for project time overrun.

As better planning is the basic formula for the proper workout and outcome of a project, most of the respondents also claimed that planning plays a key role for successful project implementation.
12. So many or complicated standards:

Figure 25 shows that this factor has not been given much importance by the respondents. 44% feel that too many or too complicated standards are to only some extent responsible for project time overrun, 27% feel that it is not at all responsible, 17% feel that it is responsible to a great extent while only 12% feel that it is certainly responsible for project time overrun.
### Project Success Factors:

<table>
<thead>
<tr>
<th>Project Success Factors</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User Involvement</td>
<td>15.8%</td>
</tr>
<tr>
<td>2. Executive Management Support</td>
<td>13.8%</td>
</tr>
<tr>
<td>3. Clear Statement of Requirements</td>
<td>13.0%</td>
</tr>
<tr>
<td>4. Proper Planning</td>
<td>9.5%</td>
</tr>
<tr>
<td>5. Realistic Expectations</td>
<td>8.5%</td>
</tr>
<tr>
<td>6. Smaller Project Milestones</td>
<td>7.6%</td>
</tr>
<tr>
<td>7. Competent Staff</td>
<td>7.3%</td>
</tr>
<tr>
<td>8. Ownership</td>
<td>5.2%</td>
</tr>
<tr>
<td>9. Clear Vision &amp; Objectives</td>
<td>2.9%</td>
</tr>
<tr>
<td>10. Hard-Working, Focused Staff</td>
<td>2.5%</td>
</tr>
<tr>
<td>Other</td>
<td>13.9%</td>
</tr>
</tbody>
</table>

According to the survey User Involvement, Executive Management Support, Clear Statement of Requirements are the major factors which influence project success.
### Project Challenged Factors:

<table>
<thead>
<tr>
<th>Project Challenged Factors</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Lack of User Input</td>
<td>12.9%</td>
</tr>
<tr>
<td>2. Incomplete Requirements &amp; Specifications</td>
<td>12.4%</td>
</tr>
<tr>
<td>3. Changing Requirements &amp; Specifications</td>
<td>11.6%</td>
</tr>
<tr>
<td>4. Lack of Executive Support</td>
<td>7.4%</td>
</tr>
<tr>
<td>5. Technology Incompetence</td>
<td>7.0%</td>
</tr>
<tr>
<td>6. Lack of Resources</td>
<td>6.3%</td>
</tr>
<tr>
<td>7. Unrealistic Expectations</td>
<td>5.8%</td>
</tr>
<tr>
<td>8. Unclear Objectives</td>
<td>5.6%</td>
</tr>
<tr>
<td>9. Unrealistic Time Frames</td>
<td>4.2%</td>
</tr>
<tr>
<td>10. New Technology</td>
<td>3.8%</td>
</tr>
<tr>
<td>Other</td>
<td>23.0%</td>
</tr>
</tbody>
</table>

Lack of User Input, Incomplete Requirements & Specifications, Changing Requirements & Specifications are the major factors which are responsible for a project to be challenged.
### Project Impaired Factors:

<table>
<thead>
<tr>
<th>Project Impaired Factors</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Incomplete Requirements</td>
<td>13.3%</td>
</tr>
<tr>
<td>2. Lack of User Involvement</td>
<td>12.2%</td>
</tr>
<tr>
<td>3. Lack of Resources</td>
<td>10.5%</td>
</tr>
<tr>
<td>4. Unrealistic Expectations</td>
<td>9.8%</td>
</tr>
<tr>
<td>5. Lack of Executive Support</td>
<td>9.4%</td>
</tr>
<tr>
<td>6. Changing Requirements &amp; Specifications</td>
<td>8.7%</td>
</tr>
<tr>
<td>7. Lack of Planning</td>
<td>8.1%</td>
</tr>
<tr>
<td>8. Didn't Need It Any Longer</td>
<td>7.4%</td>
</tr>
<tr>
<td>9. Lack of IT Management</td>
<td>6.3%</td>
</tr>
<tr>
<td>10. Technology Illiteracy</td>
<td>4.4%</td>
</tr>
<tr>
<td>Other</td>
<td>9.9%</td>
</tr>
</tbody>
</table>

Incomplete Requirements, Lack of User Involvement, Lack of Resources are the major factors responsible for project failure.

**Summary**

This chapter presents the analyzed data that is collected from the development team. Upon analysis, efficiencies and risks of scrum methodology were identified. The team suggested certain improvements for the scrum methodology by implementing new phases and tasks to the existing Scrum framework. The following
chapter will discuss the results of this project in detail along with research and future recommendations.
Chapter 5: Results, Conclusion, and Recommendations

The Methodology, Questionnaire and every interaction with all respondents, leads the following.

Results

1) What factors influence a software project?
   A. Requirements, Technical resource availability and leveraging technology for business needs are the major factors that influence a project. When these three parameters are aligned and implemented using agile methodology then, the project had higher chances of success.

2) What are the time overruns of failed projects?
   A. Time overruns depends on multiple factors that were explained above. The amount of time depends on factors like stage of the project and the deviation from the requirement. There is no definite time for overruns.

3) How do we mitigate these failures?
   A. From the reviews and analysis, it was identified that planning is the key. Effective planning and design that factor’s in all parameters of project scope will help technical resources understand and cater to the project needs. To mitigate project failure business requirements needs to be effectively captured and scope has to be laid out.

4) Was agile methodology effective in completing the software development project on time and within the budget?
A. Yes, there was a direct evidence for the positive implication of adapting agile methodology in the project. Especially the iterative design model had a positive impact on the overall functioning of the project. The series of iterations provides an opportunity to review the requirements at various stages and recalibrate if there are any missing pieces. Thus, saving cost that could incur from making changes to or completely building an application at later stages.

Summary of findings:

- The average number of projects undertaken by the respondents were 7.
- On an average 10 person months were spent by the respondents on a project.
- The time taken to complete a software project is ranged between 3 and 36 months.
- Respondents worked in teams with different sizes. The average of the respondents worked teams was about 9 members.
- Most of the respondents said that the projects mostly completed on time, signifying that the project failure due to time overrun was decreased still further from that in the year 2013.
- Majority of the respondents agreed that the products that they delivered met all the requirements as per the initial specifications and hence project failure due to the inability to meet all the requirements as per initial specification has also reduced from the same in the year 2013.
• Majority of the respondents said that only 10%-30% of the features were added later to the product. Hence we can probably say that the time and effort, both in human effort and in cost, spent on adding new features to the released product has significantly decreased.

• A majority of respondents considered managing large teams was not so easy. Having large teams working on a project could be a hindrance to the successful completion of the project.

• Communication was considered by most of the respondents as a major problem in managing the teams, as the communication is a primary base for official camaraderie. The next most popular problem in handling teams seems to be lack of experience of the team members.

• More training courses for the team members and more project reviews were considered by the respondents as the most effective ways to overcome the problems faced in managing project teams. Among various other ways that can help make the team management easier, significant emphasis was put on the need for clarification and detailed explanation of the objective of the project and the need for each process and step undertaken to complete the project.

• As per the respondents’ replies, delivery time of the project exceeded by at the most 3 months. This was again a significant improvement for the findings in the year 2009.
- Majority of the respondents said that they used time tracking or accounting tools to ensure that the project met the deadlines. Most of the tools used were in-house developed. Among the other tools used M/S Office Project, Excel Sheets and Lotus Notes were the most favored ones. However, those that do not use any time tracking or accounting tool constitute a significant proportion. Hence some work needed to be done in this sphere to ensure that such tools are used to effectively manage the software projects.

- Checks and reviews and regular meetings were the most used measures adopted to ensure that the projects met their deadlines. Other measures used included proper allocation of resources, both human and technical resources and Buffer management.

- Among the various factors that cause project time overrun, incomplete requirements and specifications is considered to be certainly responsible for such overrun by the respondents. Changing requirements and specifications and unrealistic time frames are considered as to a great extent responsible for project time overrun. Factors like lack of executive support, technology incompetence, unclear objectives, new technology and too many or too complicated standards are considered to be responsible to some extent for project time overrun. Other factors received mixed responses for the respondents.
Conclusion

The most important part of the research was to discover why projects tend to fail. According to the survey by the opinion of the IT associates the major reasons that a project would succeed are user involvement, executive management support, and a clear statement of requirements. There were other factors for the success, but with these three in place, the chances of success are much greater. Without them, chance of failure increases significantly.

The survey participants were also asked about the factors that cause projects to be challenged. The three major factors causing a project challenged are Lack of User Input, Incomplete Requirements & Specifications, Changing Requirements & Specifications.

The participant were also asked about the factors which involve in impaired and failed projects. The major three reasons which causes project failure were Incomplete Requirements, Lack of User Involvement, Lack of Resources.

Recommendations

- Smaller teams in a software projects were desired as they are easy to manage and communicate.

- Every project should have well define and managed Change management process (CRQ) to keep a track of all the changes requested and made to the initial Specs. CRQ provides enough time to plan for the change as well as the disaster recovery. CRQ ensure that the project adheres to the timelines and provides a platform to communicate with all the stake holders.
and helps creating end user manuals. Since resources are the most expensive element of an IT project and since they are paid on the basis of man hours, it is important to reduce the lag and deploy resources necessary to ensure the project deadlines are met.

- Effective communication is the key. At any given point of time, it is imperative to bring every resource on par with the whereabouts of the project. It would be a wise move to employ measure to ensure an effective flow of communication channel.

- Team composure is a crucial element that can curb investment. A team should be a balanced blend of rookies and experts which promotes higher motivation and spirits among the team. This has two positive implication: 1) The experts act as the spear heads that guides the project and force it through its way and the rookies roll it further with proper guidance. 2) The rookies gain experience from their superiors and also identify the bottle necks and come up with out of the box solutions.

- Sufficient amount of training and self-explanatory courses should make their way into the team members as well as the team leaders and project managers. The training should be imparted in both technical and soft skills. Modern and latest innovation has to be a part of the Training that is exclusively used to target the implantation of the technology rather than it merely theoretical.
• If the project is a development or an implementation project. An iterative model like agile is well suited as it involves multiple iterations for the plan and multiple reviews of a single task. This also ensure the timelines and will elevates delays if any.

• Time tracking or accounting tool should be used to track the project progress. Some recommended tools most widely used are M/S Office Project, Excel Sheets and Lotus Notes.

• While planning the project schedule, sufficient buffer and back up should be allocated to account for team members going on leave due to illness, personal emergencies, trainings etc. Often, time spent on training is not taken into consideration while planning the project schedule.

• Majority of the efforts and time should be spent for the project design, specification and requirement gathering stages so that they are complete and have negligible changes of being included at the development stage. Client partnering is essential that builds client developer relationship based on trust, credibility and relevance.

• The organizational climate also plays an important role in the performance of its employees. It represents the employees’ perceptions of the way thing are done in the organization. The climate that is created in the organization can make a difference between the winning and losing in the market place. The various elements that make up the organizational climate are clarity in
mission and direction, improvement in standards, responsibility, autonomy, flexibility, rewards, recognition, team commitment, etc.

- Team should be raised on the basis of adaptability and creativity, for solving critical unexpected business solutions.
References


Appendix

Questionnaire

Objective: To analyze the reasons behind software project time overruns

1. Name of the organization:
2. Designation:
3. Years of Experience:  < 4years  4-8years  > 8 years
4. Number of Projects undertaken:
5. Average time spent on a typical project (in person months):
6. What was the average team size? _____________
7. Did the project get completed on time? 
   Always  Mostly  Sometimes  Never
8. Was the product delivered with all the features/requirements as per initial specifications? 
   Yes  No
9. What proportion of features/requirements was added later? 
   <10%  10%-30%  30%-50%  30%-50%  >50%
10. Do you consider managing a large team as 
    Easy  Not so easy  Difficult  Very difficult
11. What problems do you face in organizing, coordinating and monitoring your team? 
    Communication  Lack of experience  Friction between members 
    Lack of management support  Others 
    Specify others ______________________________
12. What can be done to overcome the above problem? 
    More training courses  Smaller teams  More project reviews
    Others  Specify others ______________________________
13. Delivery times were exceeded by 
    <1 month 1-3 months  3-6 months  >6 months
14. Do you use any time tracking/accounting tools? 
    □ Yes  □ No
15. If yes, which? ______________________________
16. What other measures do you take to ensure that projects meet delivery deadlines?

- [ ] Regular meetings
- [ ] Checks and reviews
- [ ] Careful planning
- [ ] Allocate extra resources
- [ ] Reduce functionality
- [ ] Others

Specify others ______________________________

17. To what extent do you consider the following factors responsible for project time overruns? (Chaos, 2014)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Certainly responsible</th>
<th>To a great Extent</th>
<th>To some extent</th>
<th>Not Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of user input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incomplete requirements and specifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing requirements and Specifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of executive support</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology incompetence</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unrealistic expectations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unclear objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unrealistic time frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Too many/complicated Standards</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>