HOW DO I KEEP MY PROJECT FROM SLIPPING? IF IT DOES, HOW DO I RECOVER ITS SCHEDULE?

Sadiqali Shaik 40277437

Concordia University

SOEN 6841 Software Project Management

Professor Pankaj Kamthan

Date *(for student papers)*

TABLE OF CONTENTS

Abstract

Project Slippage is the process of a software project failing to meet its deadline as per schedule. This is a relatively common occurrence which is found in the world of software project management but it is something that can only be minimized and rather rarely completely nullified. Slippage can occur due to various conditions such as technical complexity, changing business conditions, evolving business requirements, poor estimation and several others. However, it is to be duly noted that the project manager can work with their best efforts and make necessary adjustments and reevaluations to avoid project slippage and meet the required schedule deadlines. A software project manager is the key individual responsible for gathering a team of technically proficient people to plan, analyze and execute the development of a software project. They are also responsible for communicating with all the stakeholders of said software project and cooperate with their team of developers to ensure the successful completion and submission of the software project to the customer, who is also in turn a part of the stakeholders. This report will talk about what project slippage is and how the software project manager of a company can minimize if not avoid schedule slippage. In addition to Schedule Slippage, a software project also succumbs to Cost Slippage which is not discussed but is deemed important enough to mention since it is one of the key factors which contributes to a successful Software Project.

Keywords: Software Project Management, Software Project Manager, Project Slippage, Schedule

Introduction

Software Projects are an amalgamation of several stakeholders, factors, evaluations and adjustments which in unison deliver a successful software project. One major factor that plays into the synthesis of a successful software project is time. To be precise, Schedule. A project schedule is the timeline drafted by the software project manager which explicitly describes how the given time is used to finish several components or subtasks in the software project. A schedule is not designed and finalized in one go, it is something that undergoes several reevaluations by multiple stakeholders involved with the project.

There are also cases where the deadline for delivering the project is set by the customer itself and this has its own consequences. In such a situation, a manager and their team of developers can only negotiate with the customer to get a better deadline. However, customers have their own technically proficient teams who know how long a project can take to deliver. There are also cases when the customer has a strict deadline which needs to be followed no matter what due to the fact that they have OTHER customers, i.e., the customers of our customer who have set the strict deadline. In a way, it can all be looked at as a cog in a system of several synchronously rotating gears.

Project Slippage is the term used to describe a situation where the project is behind schedule and is going to take longer to deliver than the discussed and agreed upon delivery date. This occurs due to several factors such as:

1. Scope Changes
2. Technical Challenges
3. Poor Estimation
4. Team Productivity
5. Risk Occurrences

In the next subchapter the in-depth analysis of Project Slippage is given, which forms the base of the problem statement at hand.

Problem Statement

The problem statement for the topic "How do I keep my project from slipping? If it does, how do I recover its schedule?" involves addressing the challenges of project schedule management in software development. It seeks to explore strategies and techniques for preventing project delays and, in case of slippage, implementing effective recovery measures. This problem statement emphasizes the need for proactive project management practices to ensure projects stay on track and the ability to respond to setbacks swiftly and efficiently to maintain project timelines.

Objectives

For the given problem statement, this report will look at the following objectives

1. What exactly is project slippage?
2. What are the factors or conditions that could lead to Project Slippage?
3. How can one avoid project slippage in the early stages of development?
4. If project slippage is identified, how can one circumvent it in order to minimize the slippage?

Background Study

Now that we have an understanding as to what is Schedule Slippage, there are quite a few resources that talk about it and provide an insight into the statistics of Schedule Slippage.
The following paper talks about the various factors that lead to schedule slippage with supporting statistics.

Schedule Slippage, its prevention factors & their adherence [1]

This paper talks about what Slippage is and how relatively common it is. It also mentions a Standish Group findings of major causes of project failures.

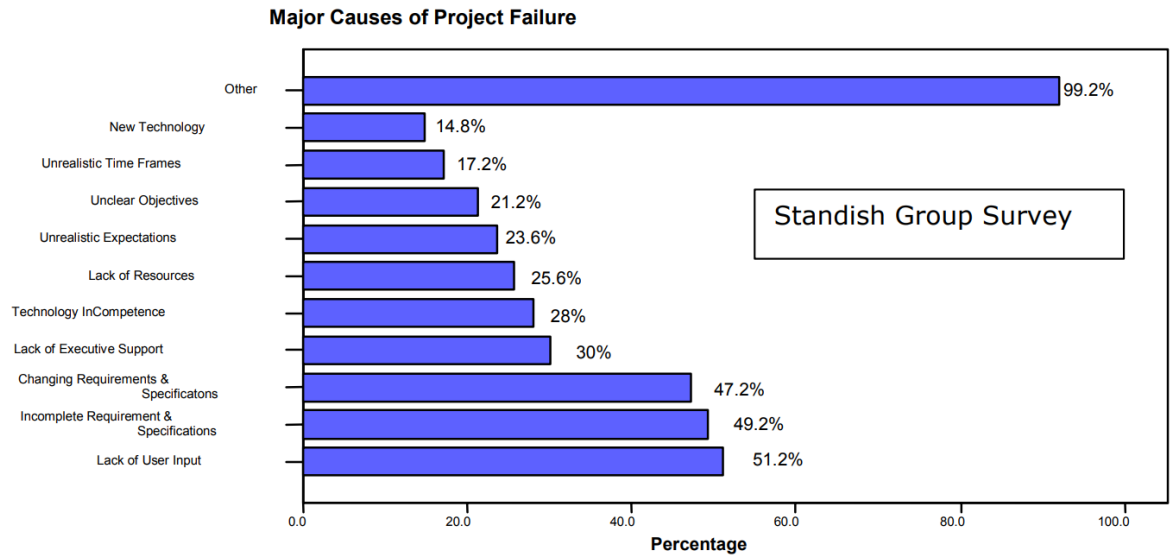


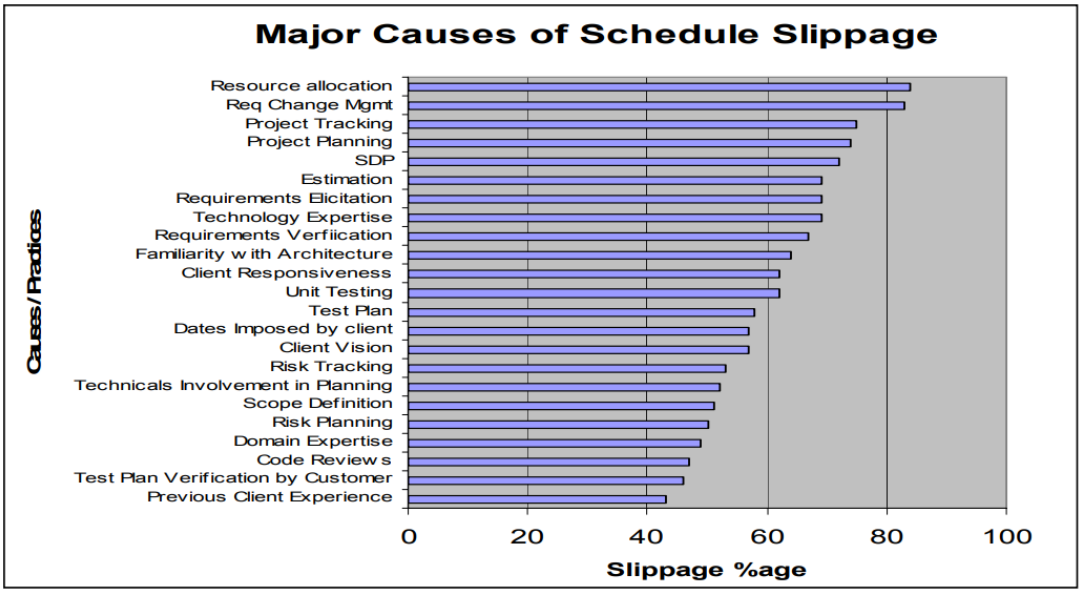
Figure 1: Standish Group Survey on Major Causes of Project Failure [1]

From the above figure, it is clear how various factors can cause a project failure.
The largest cause after "Other" is "Lack of User Input". Any software will only thrive when enough users make use of it. This is the reason a software is developed. Similarly, "Incomplete Requirements & Specifications" forms the third contender for being a factor for project failures. A crystal clear outline on the requirements and specifications is ideal however it is not always the case that a project development team is lucky enough to get one. Sometimes, it may be incomplete or lacking and other times, it just may not be comprehensible.

The authors of this paper have also mentioned and briefed on the major factors that cause schedule slippage:

1. Software Development Plan (SDP): is a document developed in the very early stages of the development. It is observed by the findings made by the authors that 72% of the projects on the local scope were designing an SDP. On an average only 20% of the projects observed slippage while having an SDP whereas the projects without an SDP were at 72%.
2. Project Plan (Gantt Chart): Almost 95% projects had a Gantt Chart designed in order to keep track of all the tasks meant to be completed. 33% of the projects were found to have slipped where a project plan was drafted whereas on the other hand there were 74% projects which observed slippage without a project plan.
3. Project Tracking: Keeping close track of the project can cause a significant change on its success. The results of the study conducted by the authors found that about 67% of the projects were keeping track of their projects. An average of 15% of the projects observed slippage while keeping track of project tracking whereas on the other hand, 75% of the projects had slipped with no tracking.

Similarly, several other factors were given along with the findings of their studies to support how basic yet necessary activities could help projects be on track with their schedule. Some other factors were, Estimation, Team Involvement in Estimation, Risk Management Plan and Dates Imposed by Clients.



Consolidated View of Factors affection Schedule Slippage [1]

With these findings we have covered the first and second objective being “What exactly is project slippage?” and “What are the factors or conditions that could lead to project slippage?”.

Causes and solutions for schedule slippage: a survey of software projects [2]

This paper discusses the causes of software schedule slippage, common solutions to overcome it, and a research approach to identify effective solutions for delayed software projects. For the purpose of this report, we will only be focusing on the solution aspects of the research done by the authors.

Upon inspecting the causes that can lead to schedule slippage, it is quite evident as to the hows of fixing these particular causes. This paper identifies quite a few common solutions which can help overcome a slipped schedule. The following are the common solutions as identified by the authors of the paper:

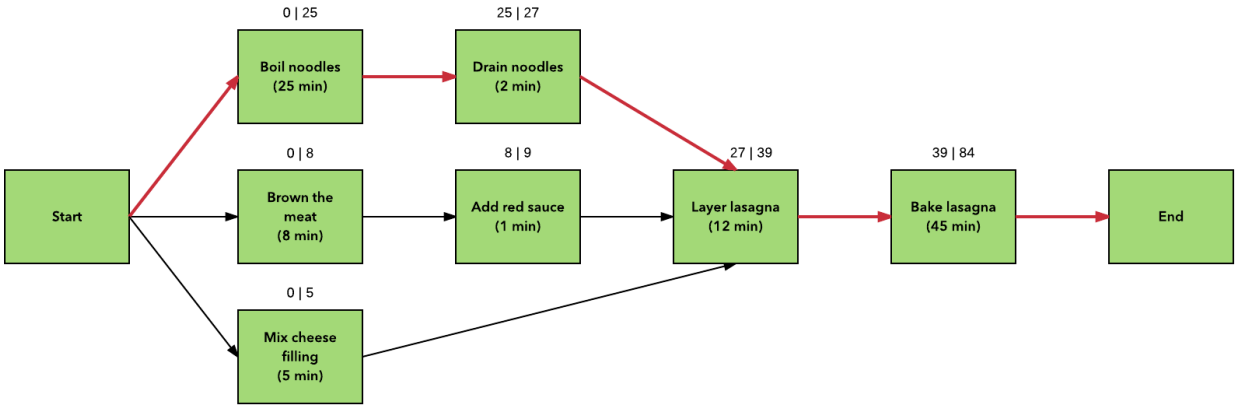
1. Code Reuse:
   The practice of incorporating existing code modules or components into the development of new software is gaining prominence. This trend is attributed to the advancements in object-oriented tools and methodologies witnessed over the past decade.
2. Test Reuse:
   Leveraging existing test cases or test suites in the testing phase of new software is an emerging strategy. This is particularly notable in the context of the advancements in object-oriented tools and methodologies in recent years.
3. Re-prioritizing Tasks:
   The strategic alteration of task order or priority within the project plan is employed to emphasize critical or time-sensitive tasks. This proactive approach ensures that pivotal work is accorded precedence, thereby contributing to on-time delivery.
4. Obtaining Hardware with Better Performance:
   Upgrading the hardware infrastructure utilized in software development or testing is undertaken to enhance performance and expedite task completion. This initiative is geared towards optimizing project efficiency.
5. Postponing Features to Next Revision:
   The deliberate deferment of the implementation of specific features or functionalities to subsequent software versions is a prudent measure. This strategic decision serves to streamline project scope and enhance manageability.
6. Maintaining Software Development or Testing Tools:
   Ensuring the currency and optimal functionality of tools employed in software development or testing is imperative. This practice contributes to heightened productivity and minimizes the occurrence of errors.
7. Maintaining Accurate and Up-to-Date Documentation:
   Rigorous attention to the accuracy and currency of project documentation is fundamental. This ensures that project information is reliable, current, and readily accessible to all team members, fostering improved communication and reducing errors.

These are the common solutions that can be used to overcome Schedule Slippage after it has been identified. Following the above solutions will practically minimize the slippage however nullifying is not a possible feat.

Methodology

CPM in Project Management [3]

Critical Path Method (CPM) is a sophisticated project management technique used to plan, schedule, and control complex projects. It identifies all necessary activities, determines their sequence and duration, and calculates the critical path—the longest sequence required for on-time project completion. CPM enables project managers to optimize time and cost, identify risks and delays, and make informed decisions for project success.



An example of what a simple CPM could look like for making Lasagna [7]

Once schedule slippage is identified, Critical Path Method (CPM) becomes a pivotal tool for recovery. By precisely delineating the critical path—the sequence of activities determining project duration—CPM allows for targeted efforts on crucial tasks. Project managers can optimize resources, resequence tasks, and employ resource leveling to efficiently allocate workforce, mitigating delays. CPM's visual representation aids in strategic decision-making, facilitating the identification and management of risks associated with slippage. Through these measures, CPM enables a focused and informed approach to address schedule deviations, promoting a more effective and timely project completion.

Another effective approach to minimize schedule slippage is to first step back and take in the look of the bigger picture of the schedule and identifying where slippage is occurring. Upon identifying the situation and the causes, a new schedule can be designed while keeping everyone in the loop. This includes, team members, customers and any other stakeholders. Keeping everyone acknowledged of the current situation while trying to fix it is the ethical approach.

Results

Maecenas id luctus ligula. Cras condimentum eleifend nibh sit amet iaculis. Suspendisse placerat sollicitudin mi, vel ornare augue hendrerit ac. Nulla sed suscipit sapien. Cras pellentesque orci lectus, eu consequat enim.

Outcome 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Outcome 2

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

References

[1] S. R. Ayyubi, M. Ahmad and F. Faiz, "Schedule Slippage, its prevention factors & their adherence (Assessment of the Project Management Best Practices which contribute in successful completion of projects in the Software Industry of Pakistan)," 2007 International Conference on Information and Emerging Technologies, Karachi, Pakistan, 2007, pp. 1-8, doi: 10.1109/ICIET.2007.4381331.

[2] Z. Ma, J. S. Collofello and D. E. Smith-Daniels, "Causes and solutions for schedule slippage: a survey of software projects," Conference Proceedings of the 2000 IEEE International Performance, Computing, and Communications Conference (Cat. No.00CH37086), Phoenix, AZ, USA, 2000, pp. 373-379, doi: 10.1109/PCCC.2000.830340.

[3] Aliyu, A. M. "Project management using critical path method (CPM): a pragmatic study." Global Journal of Pure and Applied Sciences 18.3-4 (2012): 197-206.

[4] Atin, S., and R. Lubis. "Implementation of critical path method in project planning and scheduling." IOP Conference Series: Materials Science and Engineering. Vol. 662. No. 2. IOP Publishing, 2019.

[5] ChatGPT

[6] What to do when your project management schedule slips | D&AD New Blood

[7] CPM Example