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

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 themself 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.

Background Study

Now that there is an understanding of 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. 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 in 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.

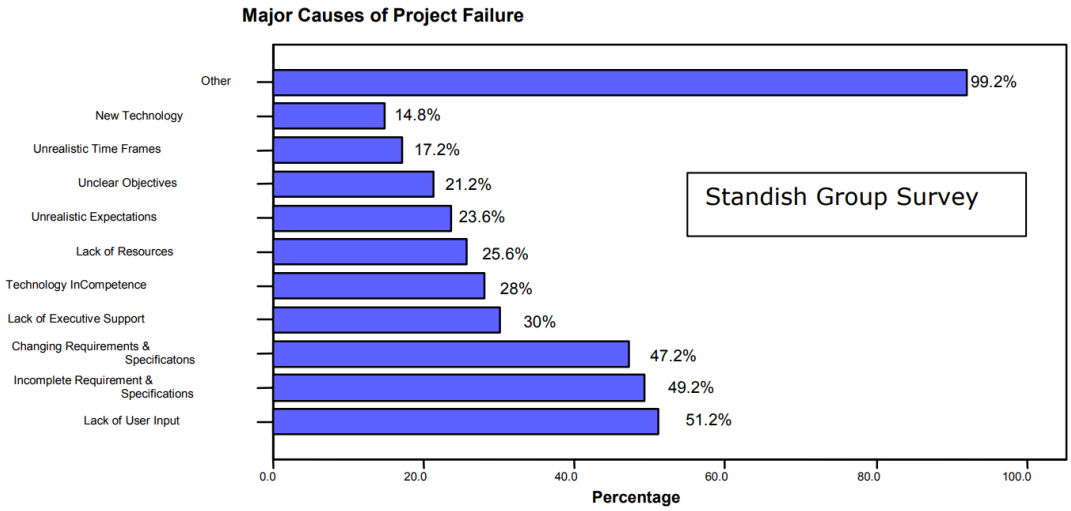
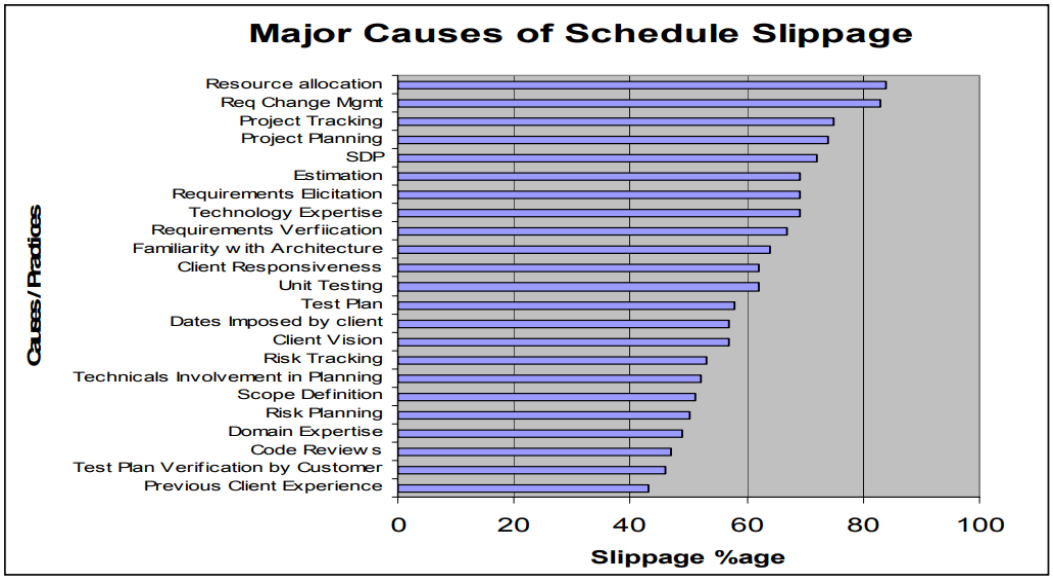


Figure 1: Standish Group Survey on Major Causes of Project Failure [1]

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.



Consolidated View of Factors affecting Schedule Slippage [1]

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.

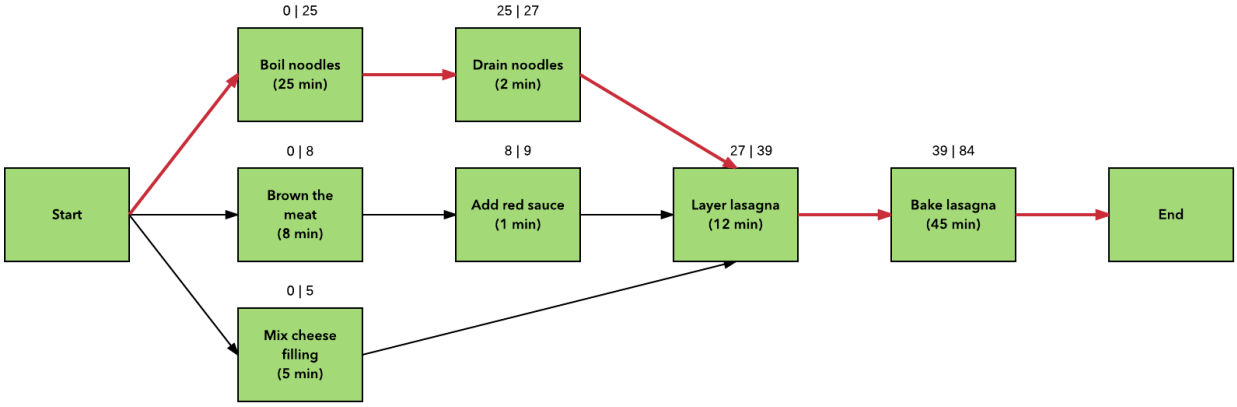
Upon inspecting the causes that can lead to schedule slippage, it is quite evident as to the hows of fixing these particular causes. This paper identifies quite a few common solutions which can help overcome a slipped schedule. The following are the common solutions as identified by the authors of the 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. **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.
3. **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.
4. **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.
5. **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.
6. **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.

Methodology

Project management using critical path method (CPM): a pragmatic study [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 identify 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.

Now, looking at ideas as to how one can avoid slippage altogether is another task, but is it really?
Any impossible task is only impossible as long as one deems it to be impossible. Once an organization starts to look at development and releasing as events that are NOT impossible to manage, there's a sudden clarity to the whole ordeal.

1. **Improve Release Visibility**:
   Enhancing real-time visibility involves implementing tools and processes that allow teams to track progress and potential roadblocks throughout the release cycle. This includes utilizing project management tools, dashboards, and communication channels that provide stakeholders with up-to-date information on the status of the release. Regular status meetings and transparent reporting can further contribute to improved visibility.

2. **Release Workflow**:
   Creating a release workflow is about establishing a systematic and documented sequence of steps from planning to deployment. This workflow serves as a guide for team members, outlining their roles and responsibilities at each stage. It should be a dynamic document that evolves with the project, capturing any changes made during the planning phase. Having a visual representation of the workflow helps in quick identification of progress and potential bottlenecks.

3. **Handle Release Issues**:
   Effectively managing release issues involves having a centralized system for tracking and prioritizing problems. This system should allow for easy collaboration among team members to address issues promptly. Regular issue review meetings and agile methodologies, such as daily stand-ups, contribute to a proactive approach in handling release issues.

4. **Allow Flow Back and Forth**:
   Allowing the flow back and forth in the release process emphasizes an iterative approach. As the product is being built, it should undergo continuous testing in a real-world environment. This iterative cycle enables teams to identify and address issues early in the development process, reducing the likelihood of major setbacks later on.

5. **Use Collaborative Platforms**:
   Collaborative platforms play a crucial role in maintaining transparency and communication within the team. Tools like project management software, collaboration platforms, and version control systems facilitate real-time sharing of information. Team members can track progress, share updates, and collaborate on tasks efficiently. Such platforms also provide a centralized repository for documents, ensuring everyone is working with the latest information. Regular updates and discussions on these platforms contribute to a shared understanding of project status.

Using these techniques it becomes quite clear as to what are the tasks and where each of them resides in the Eisenhower Matrix (Urgent-Important Matrix) and allows all team members to keep track of progress and keep slippage to a minimum from the get-go.

Results and Discussions

Addressing project slippage involves a nuanced discussion that extends beyond mere identification of causes and solutions. It requires a deep understanding of the dynamic interplay among various factors contributing to delays and strategic considerations for building a resilient project management framework.

Encouraging **Open Communication** within the project team and stakeholders is pivotal in navigating project slippage challenges. Regular status updates, transparent reporting, and collaborative discussions create an environment where challenges are identified early, fostering a collective problem-solving approach. This proactive approach ensures that potential issues are surfaced promptly, allowing teams to pivot and adjust strategies in real-time, minimizing the impact of unforeseen challenges.

Organizations stand to gain significantly from **analyzing Historical Project Data**, especially instances of slippage. Understanding patterns, recurring issues, and successful mitigation strategies from past projects provides valuable insights for future endeavors. Cultivating a culture of **Continuous Improvement** involves learning from both successes and setbacks. Project retrospectives and knowledge-sharing sessions contribute to a collective reservoir of best practices, enabling teams to evolve and adapt proactively.

Embracing **Agile Methodologies and Adaptive Strategies** is essential in addressing project slippage. Agile frameworks facilitate iterative development, allowing teams to respond swiftly to changing requirements and unexpected challenges. Integrating agile principles encourages flexibility and responsiveness, empowering teams to adjust project plans based on evolving circumstances, fostering resilience against potential slippage risks.

Robust **Risk Management** [7] is crucial in navigating project slippage challenges. This involves anticipating potential slippage factors and formulating contingency plans. Identifying **Critical Paths**, assessing their vulnerability, and having predefined strategies for risk mitigation contribute to proactive project governance.

Ensuring **Alignment** between project stakeholders and managing expectations is a critical aspect of navigating project slippage challenges. **Transparent Communication** about project timelines, potential risks, and realistic delivery expectations fosters a shared understanding of project complexities. Establishing a clear line of communication with stakeholders prevents misunderstandings and sets realistic expectations. When stakeholders are informed about potential challenges, they are more likely to support adaptive strategies during times of slippage, fostering a collaborative and understanding project environment.

In conclusion, the discussion surrounding project slippage extends beyond technical aspects to encompass the cultural, communicative, and adaptive dimensions of project management. A holistic approach that integrates these elements positions project teams to not only navigate challenges effectively but also to continually learn and evolve in the pursuit of successful project delivery.

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] Soni, Mukesh, and Madhusudan Acharya. "Schedule slippage: Root causes." (2015).

[6] Hulett, D. (2009). Practical Schedule Risk Analysis (1st ed.). Routledge. https://doi.org/10.4324/9781315601885

[7] CONROW, EDMUND H. "Implementing Risk Management."

[8] Why do my project releases slip and how can I fix it? | Jexo

[9] What are the common causes and signs of Schedule Slippage? | Linkedin

[10] ChatGPT

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

[12] CPM Example