"Exploring Software Failure Causes through Project Management "

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering in

Information Technology

by

Anjali punsi - 57 (D20B)

under the guidance of Supervisor (s): **Smita Mane**

Department of Information Technology

Vivekanand Education Society's Institute of Technology

2023-2024

Table of Contents

Table of Contents:

Introduction

During the software development lifecycle, several crucial phases are followed: Communication, Planning, Modeling, Construction, and Deployment. The initial phase is communication, where a Requirement Engineer plays a pivotal role in gathering client/customer requirements to understand their needs. Occasionally, clients may struggle to provide precise requirements, or the requirement engineer may face challenges in comprehending minimal client needs, potentially leading to software failure. After well-defined phases, the final stage is deployment, where the developed software is handed over to the client for evaluation and use. If the software aligns with the client's requirements, it succeeds, bolstering the organization's reputation. However, if it falls short of client expectations, it can tarnish the organization's standing. Software failures have ramifications beyond the client, impacting economic growth and quality of life. Common factors contributing to software failure include unrealistic project goals, inaccurate resource estimations, poorly defined requirements, unmanaged risks, communication breakdowns between stakeholders, inadequate project management, stakeholder politics, and commercial pressures.

Client-reported issues, such as system downtimes or software crashes, often arise from missing components within the software. Detecting and rectifying errors promptly is essential since once software is in the market, turning back becomes impractical. Organizational structure, budget, and resource allocation are critical factors, and upper-level management, development teams, and project managers face immense pressure, which can affect project performance and outcomes. Ultimately, the software's quality and functionality are paramount; even if developed on time and within budget, if it fails to deliver the desired functionality, it remains ineffective.
Success

Numerous factors influence the success or failure of software systems, with both positive and negative consequences. Recognizing these factors that can contribute to project failure is vital. Identifying the causes of software project failures is essential to effectively manage and mitigate them, ensuring a positive organizational reputation. Consequently, the success of an organization is closely linked to the success of its software projects. This paper aims to identify the primary causes of software system failure in the software industry and explore strategies for overcoming these issues by leveraging existing software development models.

Literature Review:

In software project development, personality conflicts within project teams can be a significant factor leading to project failures. When team members fail to fulfill their roles or experience clashes, it can disrupt the project's overall progress. These conflicts not only result in dissatisfaction for the client but also have a detrimental impact on the organization. A study analyzing 70 failed projects underscored that numerous factors contribute to software project failures [2]. Authors in another study emphasized the importance of organizational structure, budget allocation, and resource management, highlighting the immense pressure placed on upper-level management, development teams, and project managers, which can negatively affect team performance and project outcomes [3]. Paul Dorsey and his colleagues identified that software projects often fail to achieve perfection, with failure rates ranging from 50% to 80%. Failures in the software development field can occur despite adherence to established rules and practices [4].

Today, many organizations adopt agile development methodologies, employing various models to develop software incrementally and ensure each phase is well-defined and developed. Alarmingly, out of over 50,000 IT projects, only 29% are deemed successful [5]. Another study by Peter Henderson et al. delved into the failure of large software projects, citing issues such as exceeding budgets, delivering products that don't align with client expectations, and prolonged development timelines due to a lack of knowledge and skills among project members [6]. Kaur et al. explained that software development employs process models to manage concerns related to cost, time, quality, and client requirements, with common causes of failure including estimation errors, unclear project goals, evolving requirements, and a lack of alignment between client specifications and project outcomes [7].

Furthermore, research by different companies reveals that a substantial number of project failures result from issues related to time and budget constraints, with some projects being canceled before completion. Poor project management emerges as the most prevalent factor contributing to software project failures, characterized by inadequate planning, unclear objectives, evolving project goals, unrealistic resource and time estimates, limited user involvement, insufficient executive support, inadequate communication and teamwork, and a shortage of appropriate skills [8]. On the flip side, successful software projects are characterized by their ability to meet

business goals, adhere to schedules and budgets, and satisfy customer requirements [9]. Failure to meet these criteria often leads to project failure. One significant finding suggests that a majority of software projects tend to experience delays, budget overruns, and reduced functionality, with poor management being the primary culprit.

Research Objectives:

We face many problems during the development of a software project. There are many factors that count too much in the failure and success of software projects. If we try to avoid or overcome those causes then the failure ratio of the software system can be decreased to some extent causing a good repute to the organization.

The objectives of our research are:

1. To do a critical analysis of those causes and give reasons due to which most organizations move towards the failure. We will also focus on those causes that how they affect a software system which can cause organizations to have a bad repute.
2. To study and identify the best software development model from the existing ones which can help to overcome the causes of software failure making the status of the organization high which is responsible for making that specific software.

Information Collection:

Information collection or data collection is the far most important part in the research. Due to this phase we are able to find the views of people working in a software house and how software can fail or what are the factors affecting the software failure. We have collected data from Pakistan 26 software houses. We have targeted the people which include developers, QA analysts, Requirement engineers, software engineers, software developers and project managers.

We have analyzed the data by providing the questionnaires to the people and they have selected the options that are suitable according to them. This survey was for Pakistan software houses and by this analysis we have seen the features that are important for software project success and can minimize the failure of software. We need to meet these parameters for the success of the project as these parameters identify the problems for software failure. We have been able to gathered data from Web developers, QA analysts, Requirement engineer, Software developers, Software Engineer, Project managers, Business analyst as shown in figure 1. It is essential to select the audience from which you need to get the views about a specific product. These people have been involved with the type of issues so their feedback counts.

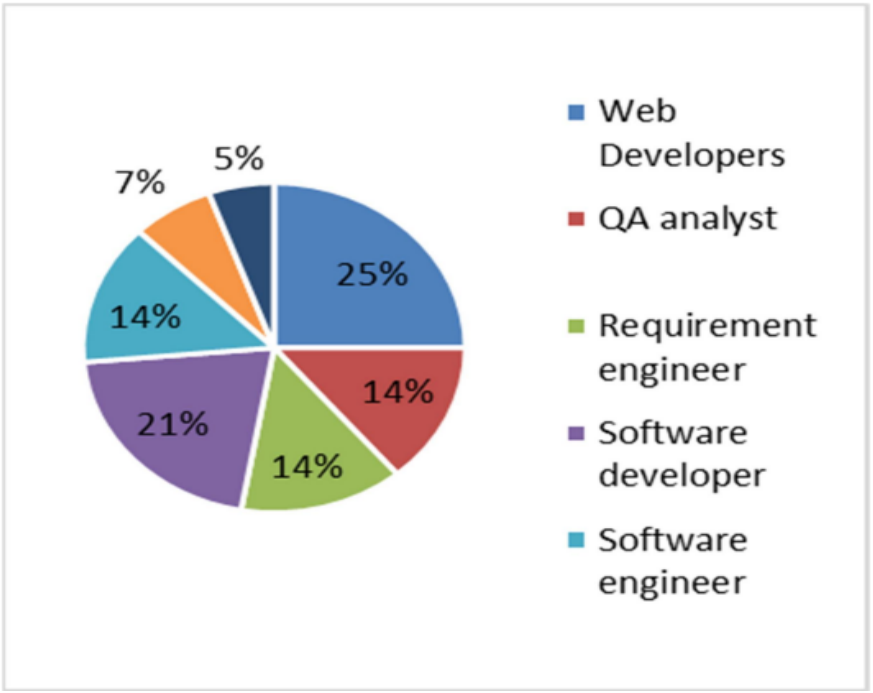

figure 1 : software result

We have gathered information on what type of projects are developing in different software houses. Most of the software houses follow these techniques which I have mentioned in the paper to avoid the failure of the software. The projects which are being developed include Web applications, theme development, Ecommerce sites, Mobile applications and Game development as shown in figure 2.

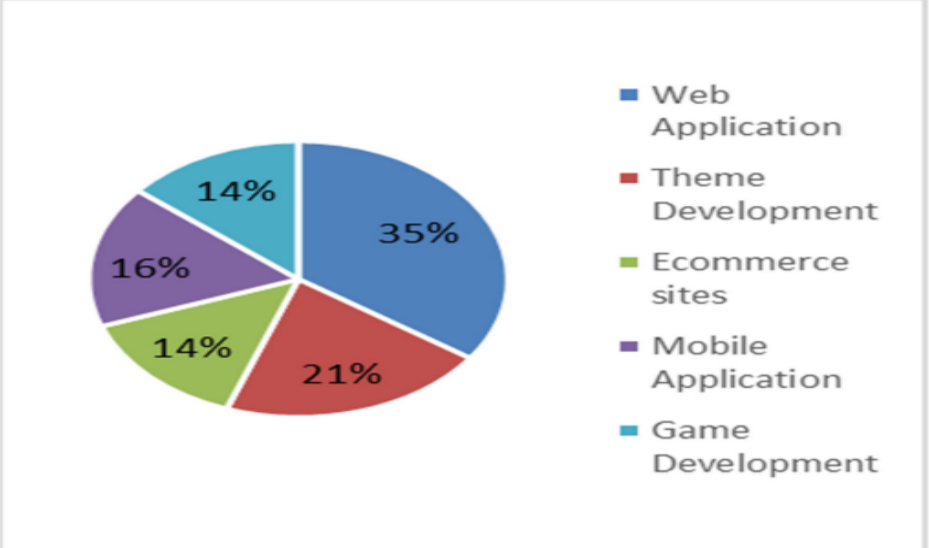

figure 2 : projects in software development

There are a lot of problems which have been identified in the field of software developments but the main issue which most of the users agree on is that Lack of access to end users is a big reason in the failure of software projects. From the graph it is seen that most of the people strongly agreed on this problem as seen

Table 1: Lack of access to end-users is a big reason for the failure of software project

	Frequency	Percent
Strongly Agree	44	61.1
Agree	25	34.7
Disagree	3	4.2
Total	72	100.0

Table 2:The major reason for project failures is the change of project scope during the project's process

	Frequency	Percent
Strongly Agree	40	55.6
Agree	25	34.7
Disagree	4	5.6
Strongly Disagree	3	4.2
Total	72	100.0

we have stated some causes of software failure in software industry which we analyzed from our research and study.

Discussion And Future Work:

The focus of our research lies in identifying the primary reasons and obstacles that have a direct impact on the reputation of software development companies. The ultimate measure of success in this endeavor is the effectiveness of a software system, and our study is geared towards unraveling the key factors contributing to this success or failure. To enrich our investigation, we have sought feedback from employees working within software houses, tapping into their insights and reviews regarding the causes behind software failures. Furthermore, we have diligently explored various factors known to influence software failures. Our future work will revolve around mitigating these faults and errors, with the aim of enhancing software success, benefiting both clients and organizations alike.

Some of the critical factors contributing to software failures, as elucidated in our research, encompass incomplete requirements, ambiguous goals, time constraints, cost management issues, and a dearth of user involvement during the development process. Addressing these issues and keeping them in mind throughout the software development lifecycle is paramount, as failure not only impacts clients and developers but also casts a shadow on the overall reputation of the organization. The significance of organizational reputation cannot be overstated in the software development landscape, which encompasses a multifaceted process from initial requirement gathering to final deployment.

The inception of software development hinges on the meticulous gathering of client requirements, a phase where communication between the requirement engineer and the client is pivotal in creating a comprehensive requirement document. Subsequently, the software undergoes various developmental stages guided by this initial document until deployment. The ultimate success of the software is contingent upon its alignment with the client's stipulated requirements. Many researchers have lent their insights to this topic, underscoring the critical role of requirement gathering. If this primary phase is executed with precision and efficiency, it sets the tone for the accuracy of subsequent phases. Conversely, errors and ambiguities in this initial stage can ripple through the entire development process.

In the context of the software industry in Pakistan, our research delves into the unique challenges faced by the sector. We conducted surveys across 14 to 15 different software houses in Pakistan to gain a comprehensive understanding of software failure

issues within the country's software industry. Our thesis represents a deep dive into the intricacies of software failures in the Pakistani context. We aspire that our findings and analyses will serve as a valuable resource for the Pakistan software industry, aiding in the implementation of best practices and ultimately contributing to improved software development processes in the future.

REFERENCES

[1] Charette, R. N. (2005). Why software fails [software failure]. Ieee Spectrum, 42(9), 42-49.

[2] Verner, J., Sampson, J., & Cerpa, N. (2008, June). What factors lead to software project failure?. In Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on (pp. 71-80). IEEE.

[3] Agarwal, N., & Rathod, U. (2006). Defining 'success' for software projects: An exploratory revelation. International journal of project management, 24(4), 358-370.

[4] 'Top 10 reasons why systems projects fail', http://rrsg.ee.uct.ac.za/courses/EEE4084F/Archive/2014/Assignments/ReadingAssignments/Lect21Dorsey_Top10ReasonsSystemsProjectsFail.pdf, Retrieved Feb., 10 (2018)

[5] Whitney, K. M., & Daniels, C. B. (2013). The root cause of failure in complex IT projects: complexity itself. Procedia Computer Science, 20, 325-330.

[6] Henderson, P. (2006). Why large it projects fail. ACM Trans. Program. Lang. Syst, 15(5), 795-825..

[7] Kaur, R., & Sengupta, J. (2013). Software process models and analysis on failure of software development projects. arXiv preprint arXiv:1306.1068.

[8] Young, S. (2003). Why IT projects fail. Computerworld, 37(34), 44-44.

[9] Berntsson-Svensson, R., & Aurum, A. (2006, September). Successful software project and products: An empirical investigation. In Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering (pp. 144-153). ACM.