



# **Critical success factors and main risks of failure in software development projects**

*Davood Sheibani*

*January 2025*

*Professor : Alessandro Annarelli*

## Abstract

The complexities of defining and achieving success in software development projects, particularly outsourced ones, have been a central focus of this study. Traditional success metrics—time, cost, and scope—are no longer sufficient in capturing the nuanced realities of modern projects. This essay explores the evolving definitions of success, emphasizing the limitations of the "Iron Triangle" framework and the increasing importance of stakeholder satisfaction, strategic alignment, and adaptability.

Key challenges in software development, such as unrealistic estimations, evolving requirements, and misaligned stakeholder goals, were analyzed. The essay also highlights the critical success factors for outsourced projects, including effective communication, goal alignment, and proactive risk management. Supplier perspectives were explored in detail, shedding light on the balance between profitability and customer satisfaction, as well as strategies to navigate risks in dynamic project environments.

Additionally, frameworks such as ISO/IEC 12207 were evaluated, demonstrating their value in standardizing success metrics while also calling for their adaptation to include subjective and evolving criteria. Through examples of successful and failed projects, the study illustrates how a comprehensive, multidimensional approach to measuring success can improve outcomes and strengthen collaborations between customers and suppliers.

The essay concludes by underscoring the importance of mutual understanding, balancing short- and long-term objectives, and refining estimation and goal alignment strategies. It advocates for further research into supplier dynamics and the development of comprehensive success metrics that reflect the complexity of modern software development. By addressing these issues, the industry can move toward more sustainable and effective project outcomes.

# Contents

## 1. Introduction

- 1.1. Overview of Software Development Project Failures
- 1.2. Challenges in Project Estimation
- 1.3. Distinction Between Project Success and PM Success
- 1.4. Stakeholder Perspectives on Success
- 1.5. Focus on Outsourced Software Development

## 2. Defining Success in Software Development

- 2.1. Traditional Success Criteria: Time, Cost, and Scope
- 2.2. Limitations of the "Iron Triangle" Framework
- 2.3. Evolving Definitions of Project Success

## 3. Challenges in Software Development Projects

- 3.1. The Role of Unrealistic Estimations
- 3.2. Changing Requirements and Their Impact
- 3.3. Misalignment of Stakeholder Goals

## 4. Project Success vs. Project Management Success

- 4.1. Conceptual Distinctions
- 4.2. The Role of Project Managers in Bridging Success Metrics
- 4.3. Case Studies Highlighting the Distinction

## 5. Outsourced Software Development: A Supplier's Perspective

- 5.1. Stakeholder Dynamics in Outsourcing
- 5.2. Profitability vs. Customer Satisfaction
- 5.3. Risk Management Practices in Outsourced Projects

## 6. Critical Success Factors in Outsourced Software Projects

- 6.1. Effective Communication and Collaboration
- 6.2. Alignment of Goals Between Customers and Suppliers
- 6.3. Managing Evolving Requirements

## 7. Frameworks and Standards for Measuring Success

- 7.1. ISO/IEC 12207 and Standardized Metrics
- 7.2. Adapting Frameworks to Subjective and Evolving Criteria
- 7.3. Examples of Successful and Failed Projects

## **8. Lessons Learned and Recommendations**

8.1. Importance of Mutual Understanding

8.2. Balancing Short-Term and Long-Term Objectives

8.3. Strategies for Improving Estimation and Goal Alignment

## **9. Conclusion**

9.1. Summary of Findings

9.2. Relevance of Supplier Perspectives in Future Research

9.3. Call for Comprehensive Success Metrics

## **10. References**

# **1. Introduction**

## **1.1 Overview of Software Development Project Failures**

Software development projects have long been associated with a high rate of failure, often characterized by unmet objectives related to cost, time, and quality. Despite advancements in methodologies and tools, the issue persists, raising questions about the effectiveness of current practices. As Cerpa and Verner (2009) observe, “Software has been successfully applied in a large variety of areas, yet software development projects have a reputation for failure.”<sup>1</sup>

One significant factor contributing to these failures is inaccurate project estimation. According to Glass (2001), “Most software estimates are performed at the beginning of the software development process before the problem is understood.”<sup>2</sup> These early estimates are typically made by individuals who are not directly involved in the project, such as upper management or marketing teams, rather than by developers or project managers. Consequently, the projections often fail to account for the complexity of the tasks ahead, resulting in flawed cost and schedule targets.

Additionally, the evolving nature of software requirements exacerbates project challenges. Bannerman (2008) notes that “requirements almost certainly change during the course of a software development project, influencing both the schedule and costs.”<sup>3</sup> These dynamic conditions make it nearly impossible to provide accurate estimates at the project’s inception.

Furthermore, the criteria used to assess success are often too narrow, focusing on traditional metrics like time, cost, and functionality. This approach ignores other critical dimensions, such as long-term stakeholder satisfaction and adaptability. Taylor (2007) emphasizes this issue in the context of outsourced projects, stating, “Customer and supplier perspectives on success often differ, reflecting contrasting priorities and perceptions of risk.”<sup>4</sup>

The prevalence of these issues highlights a systemic failure to align goals, manage risks, and adapt to changes effectively. As Reel (1999) argues, “Failure in software development often stems from neglecting critical success factors, including stakeholder alignment and realistic expectations.”<sup>5</sup> These persistent challenges underscore the need for a broader and more inclusive understanding of success in software projects.

## **1.2 Challenges in Project Estimation**

Accurate estimation in software development projects remains a persistent challenge, often cited as a primary factor contributing to project failures. Many of these challenges stem from the inherent complexity and unpredictability of software development tasks, compounded by flawed practices in estimation processes.

One of the key issues is the timing of estimations. Glass (2001) points out, “Most software estimates are performed at the beginning of the software development process before the

problem is understood.”<sup>6</sup> This premature timing leaves little room for the intricacies of the project to be accounted for, as requirements are often incomplete or unclear at this stage.

In addition to timing, the responsibility for estimations is frequently misplaced. As Reel (1999) notes, “Estimates are rarely made by those directly involved in the development process, but instead by upper management or marketing teams.”<sup>7</sup> This disconnect means that estimates are often unrealistic, as they fail to consider the practicalities of implementation.

Furthermore, changing requirements add another layer of complexity to project estimation. Bannerman (2008) highlights this issue: “Requirements almost inevitably evolve during a project, which impacts schedules and costs significantly.”<sup>3</sup> While agile methodologies aim to address such changes dynamically, they do not eliminate the difficulty of predicting resource needs early on.

Estimation challenges are further compounded by external pressures to commit to fixed timelines and budgets, even when uncertainties remain high. Dikert et al. (2016) observe, “Teams frequently face pressure to deliver based on initial estimates, even when significant deviations become apparent during development.”<sup>8</sup> Such pressures can lead to unrealistic commitments that jeopardize both project success and stakeholder trust.

The failure to revise and adapt initial estimates is another recurring problem. As Taylor (2007) explains, “Once estimates are set, they are rarely revisited, even when evidence suggests they are no longer valid.”<sup>4</sup> This rigidity contributes to the misalignment between expectations and actual project outcomes, further intensifying the risk of failure.

These challenges underscore the need for improved estimation practices that incorporate iterative planning, stakeholder collaboration, and flexibility to adapt to evolving project dynamics. Without addressing these fundamental issues, project estimation will remain a weak link in achieving software development success.

### **1.3. Distinction Between Project Success and PM Success**

The concepts of project success and project management (PM) success are often conflated, but they are distinct and interrelated measures of a project's outcomes. Understanding the difference is crucial for evaluating software development projects comprehensively.

Project management success is typically assessed using measurable criteria such as adherence to budget, timeline, and scope. Munns and Bjeirmi (1996) define project management as “the process of controlling the achievement of project objectives by applying a collection of tools and techniques.”<sup>9</sup> This focus on operational efficiency makes PM success a more objective and quantifiable measure.

In contrast, project success encompasses broader, long-term outcomes, including stakeholder satisfaction and the value delivered by the project. As Ika (2009) explains, “Project success goes

beyond the traditional time, cost, and quality metrics, focusing on customer satisfaction and the project's alignment with strategic goals.”<sup>10</sup> This distinction highlights the fact that a project can achieve management success while failing to deliver meaningful benefits, and vice versa.

A well-known example illustrating this distinction is the Sydney Opera House. Despite severe delays and budget overruns, it is celebrated as an engineering and architectural masterpiece. De Wit (1988) argues that “Good project management can contribute to project success but cannot guarantee it, just as poor project management does not always result in failure.”<sup>11</sup> This example demonstrates how long-term success may override short-term inefficiencies in project execution.

Another critical difference lies in stakeholder perspectives. Jugdev and Müller (2005) state, “The criteria for evaluating PM success are more universal, while project success criteria are subjective and vary among stakeholders.”<sup>12</sup> For instance, a supplier might consider a project successful if profitability is achieved, while the customer might evaluate success based on usability or strategic alignment.

Furthermore, the dynamic nature of software projects necessitates a distinction between the two concepts. Papke-Shields et al. (2010) emphasize that “PM success is tied to the processes and tools applied during the project, whereas project success reflects the outcomes and their alignment with broader organizational goals.”<sup>13</sup> This distinction encourages a dual focus on both immediate project execution and long-term impact.

Recognizing and addressing the differences between project success and PM success allows organizations to adopt more holistic evaluation frameworks. This ensures that both the operational and strategic dimensions of a project are effectively addressed, leading to improved outcomes for all stakeholders.

## **1.4. Stakeholder Perspectives on Success**

The definition of project success varies significantly depending on the perspective of the stakeholder involved. Each stakeholder group—customers, suppliers, and project teams—has distinct priorities and success criteria, often influenced by their roles and objectives within the project.

For customers, project success typically revolves around achieving desired outcomes, such as a functional product that meets their needs and delivers value. According to Procaccino and Verner (2006), “Customers primarily judge success based on how well the delivered system meets their requirements and expectations.”<sup>14</sup> This perspective emphasizes usability, reliability, and alignment with business objectives.

In contrast, suppliers often view success through the lens of profitability and operational efficiency. As de Wit (1988) explains, “While the customer aims to minimize costs, the supplier’s primary goal is to maximize profit.”<sup>11</sup> This divergence in priorities can lead to

conflicts, particularly when customers demand additional features or modifications that exceed the initially agreed-upon scope.

Project teams, on the other hand, may define success based on technical achievements and adherence to project constraints, such as time, cost, and scope. As Jugdev and Müller (2005) note, “The project team’s view of success often focuses on the execution process, including the effective use of resources and delivery according to plan.”<sup>12</sup> This perspective aligns closely with the concept of project management success but may overlook broader strategic outcomes.

Moreover, the perceptions of success are influenced by the degree of stakeholder involvement. Taylor (2007) points out that “Different stakeholders perceive and prioritize risks differently, which directly impacts their view of project success.”<sup>4</sup> For instance, a lack of communication between the customer and supplier may result in misaligned expectations, causing one party to perceive a project as successful while the other considers it a failure.

Recent studies also highlight the evolving nature of success criteria over the course of a project. Bannerman (2008) observes, “As projects progress, stakeholder priorities may shift, requiring a reevaluation of what constitutes success.”<sup>3</sup> This dynamic underscores the importance of ongoing communication and collaboration among stakeholders to ensure a shared understanding of objectives and outcomes.

Acknowledging and addressing the diverse perspectives of stakeholders is critical to achieving a balanced definition of success. By aligning goals and fostering collaboration, project teams can better navigate the complexities of stakeholder expectations, ultimately improving the likelihood of project success.

## **1.5. Focus on Outsourced Software Development**

Outsourcing software development has become a widespread practice, driven by the potential for cost savings, access to specialized skills, and faster time-to-market. However, it introduces unique challenges and complexities, particularly when it comes to defining and achieving project success. The involvement of two distinct entities—the customer and the supplier—often leads to diverging goals and perceptions of success.

Taylor (2007) highlights this divergence, stating that “In outsourced IT projects, the customer’s primary goal is to minimize costs and risks, while the supplier focuses on maximizing profit and managing operational risks.”<sup>4</sup> This fundamental difference in priorities underscores the need for clear contracts and communication to ensure alignment between both parties.

Additionally, outsourced projects are particularly susceptible to risks arising from cultural and geographical differences. As Dibbern et al. (2004) explain, “Cultural mismatches and time zone differences can exacerbate communication gaps, leading to misunderstandings and delays in software development projects.”<sup>15</sup> These issues often affect the customer’s perception of the project’s success, especially when the delivered product deviates from expectations.



From the supplier's perspective, outsourced projects pose risks related to scope changes and payment structures. Bannerman (2008) emphasizes that "Suppliers frequently face scope creep, which can erode profitability and strain resources."<sup>3</sup> These risks necessitate robust risk management practices to mitigate their impact on project outcomes.

The challenges of outsourcing are further compounded by differing perceptions of project success between the customer and supplier. According to Jun et al. (2010), "Customers and suppliers often have contrasting views of risks and rewards, which can lead to disputes over whether a project should be deemed successful."<sup>16</sup> This misalignment highlights the importance of fostering collaborative relationships and shared success metrics.

Despite these challenges, outsourced software development projects can succeed with proper planning and management. Levina and Ross (2003) advocate for the creation of "relational contracts"<sup>17</sup> that emphasize mutual trust and partnership over rigid transactional agreements. This approach not only reduces conflicts but also enhances the likelihood of achieving both project and PM success.

In summary, outsourced software development presents both opportunities and challenges. By addressing the inherent complexities of cross-organizational collaboration, stakeholders can align their objectives and create a framework for achieving success from both customer and supplier perspectives.

## 2. Defining Success in Software Development

### 2.1. Traditional Success Criteria: Time, Cost, and Scope

In the realm of software development, the traditional metrics for defining project success—time, cost, and scope—have long dominated. These criteria, often referred to as the "iron triangle" or "triple constraints," measure a project's ability to meet deadlines, stay within budget, and deliver specified features. As Atkinson (1999) explains, "The time-cost-quality triangle has historically been the most recognized model for evaluating project success."<sup>18</sup>

**Time** represents the adherence to project schedules, emphasizing the importance of delivering the product on or before the agreed deadline. This is particularly critical in industries where delays can lead to significant financial or reputational losses. However, Glass (2001) cautions that "Most software estimates are made early in the process, often by individuals removed from the development team, leading to unrealistic timelines."<sup>6</sup>

**Cost** pertains to staying within the allocated budget. While essential, this metric often faces challenges due to changing requirements or unforeseen complexities. Kappelman et al. (2006) note that "Budget overruns in software projects are common, often stemming from inadequate initial estimations and evolving project requirements."<sup>19</sup> This highlights the difficulties of forecasting software development costs with precision.

**Scope**, or the functionality delivered, is integral to meeting customer expectations. Projects that fail to deliver promised features are often deemed unsuccessful, regardless of their adherence to time and cost constraints. As Lai (1997) observes, "Meeting functional requirements is paramount in determining whether a software project fulfills its intended purpose."<sup>20</sup>

Despite their ubiquity, these traditional criteria have faced criticism for their narrow focus. De Bakker et al. (2010) argue that "Using time, cost, and scope as sole success metrics often leads to the conclusion that a project has failed, even when it delivers significant value in other ways."<sup>21</sup> This is particularly true for software projects, where requirements frequently evolve during development, affecting all three constraints.

While the iron triangle provides a baseline for evaluating project outcomes, its limitations necessitate a broader perspective. Recent discussions in software development suggest incorporating additional criteria, such as stakeholder satisfaction and long-term value, to achieve a more comprehensive understanding of project success.

### 2.2. Limitations of the "Iron Triangle" Framework

While the traditional "iron triangle" framework of time, cost, and scope has provided a foundation for measuring project success, it has significant limitations. These metrics often fail to capture the complexity and multi-dimensional nature of software development projects. As Ika

(2009) argues, “The iron triangle is overly simplistic and does not account for long-term value, stakeholder satisfaction, or the dynamic nature of software projects.”<sup>10</sup>

One major limitation is its inability to address changes in project requirements. Software development is inherently iterative, and evolving user needs often necessitate mid-project adjustments. De Bakker et al. (2010) note that “In software projects, initial requirements almost always change, which inevitably impacts time and cost estimates, leading to misclassification of projects as failures.”<sup>21</sup> This dynamic aspect of software development highlights the inadequacy of rigid success criteria.

Another concern is the exclusion of stakeholder perspectives. The iron triangle measures success based on technical and managerial factors but overlooks whether the delivered product meets user expectations or generates value for the organization. As Procaccino and Verner (2006) emphasize, “Success cannot be defined solely by meeting time, cost, and scope; stakeholder satisfaction is a critical determinant of project success.”<sup>14</sup>

Additionally, the iron triangle does not consider the quality and sustainability of the outcomes. While time, cost, and scope focus on immediate deliverables, they fail to address whether the software is maintainable, scalable, or aligned with strategic goals. Jugdev and Müller (2005) highlight this gap, stating, “Project success extends beyond the short-term constraints of the iron triangle to include long-term benefits and strategic alignment.”<sup>12</sup>

Finally, this framework assumes that all three metrics hold equal weight in every project, which is rarely the case. Different projects and stakeholders prioritize these dimensions differently, making a one-size-fits-all approach impractical. Taylor (2007) critiques this rigidity, noting that “The relative importance of time, cost, and scope varies depending on the project context and stakeholder priorities.”<sup>4</sup>

In conclusion, while the iron triangle serves as a starting point for defining success, its limitations necessitate broader and more flexible evaluation criteria. Incorporating additional dimensions such as stakeholder satisfaction, strategic alignment, and adaptability can provide a more holistic understanding of project success in software development.

### **2.3. Evolving Definitions of Project Success**

Over time, the definition of project success in software development has evolved beyond the traditional metrics of time, cost, and scope. Researchers and practitioners increasingly recognize that these constraints fail to capture the broader outcomes of a project. As Jugdev and Müller (2005) observe, “Project success is a multi-dimensional concept that includes short-term efficiency as well as long-term effectiveness and stakeholder satisfaction.”<sup>12</sup>

One significant evolution in defining success is the inclusion of customer and stakeholder satisfaction as critical metrics. As Shenhar et al. (2001) argue, “Project success must account for the perceptions and experiences of stakeholders, particularly customers, who ultimately judge the

value of the delivered solution.”<sup>22</sup> This shift highlights the need for alignment between project outcomes and user expectations.

Another emerging perspective is the focus on long-term value and strategic impact. Ika (2009) emphasizes that “Project success must be measured not just by operational outputs but also by the broader benefits it delivers to the organization over time.”<sup>10</sup> For software development projects, this can include increased efficiency, enhanced user experience, and alignment with business goals.

Flexibility and adaptability have also become critical aspects of success in software projects. The agile development methodology, which embraces iterative processes and evolving requirements, underscores the importance of responding to change. As Dikert et al. (2016) explain, “Success in agile software development often hinges on the ability to adapt to shifting priorities and deliver incremental value throughout the project lifecycle.”<sup>8</sup>

Furthermore, researchers have emphasized the distinction between project management success (adherence to processes) and project success (overall outcomes). Munns and Bjeirmi (1996) describe this distinction, stating, “Project management success pertains to efficiency in execution, while project success encompasses effectiveness in meeting strategic goals.”<sup>23</sup> This broader understanding acknowledges that even projects with significant delays or cost overruns can ultimately be successful if they deliver valuable and impactful results.

In conclusion, the evolving definitions of project success in software development reflect a growing recognition of the need for more holistic and flexible evaluation criteria. By considering customer satisfaction, strategic alignment, and adaptability, the field is moving toward a more nuanced understanding of what constitutes true success in software projects.

### **3. Challenges in Software Development Projects**

Software development projects are fraught with challenges that can derail timelines, inflate costs, and reduce overall effectiveness. Among the most significant obstacles are unrealistic estimations, evolving requirements, and the misalignment of stakeholder goals. Understanding these challenges is crucial for improving project outcomes.

#### **3.1. The Role of Unrealistic Estimations**

Unrealistic estimations remain a persistent issue in software development projects. Glass (2001) highlights this, stating, “Software estimates are often made before the problem is fully understood, leading to flawed and overly optimistic projections.”<sup>6</sup> This problem is exacerbated by the fact that initial estimates are rarely revisited or adjusted as the project progresses, resulting in a misalignment between expectations and reality.

Reel (1999) adds, “The rush to promise aggressive timelines and low costs often stems from pressures imposed by upper management or marketing teams, who lack a full grasp of the complexities involved in software development.”<sup>5</sup> These pressures can lead to projects being labeled as failures, even when significant progress is made toward meeting project goals.

Moreover, unrealistic estimations undermine team morale and productivity. As Bannerman (2008) argues, “Inaccurate estimates create a cycle of stress and diminished trust among stakeholders, ultimately harming the project’s chances of success.”<sup>3</sup> Addressing this issue requires more robust and iterative estimation practices, alongside improved communication between technical teams and decision-makers.

#### **3.2. Changing Requirements and Their Impact**

Software development projects are particularly vulnerable to shifting requirements, which can have a profound impact on project scope, timelines, and budgets. De Bakker et al. (2010) observe, “Requirements almost invariably change during software projects, challenging traditional success criteria and necessitating greater flexibility in project planning.”<sup>21</sup>

The iterative nature of modern software development methodologies, such as agile, attempts to address this challenge by emphasizing adaptability. However, even agile projects can struggle when requirements change too frequently or without sufficient input from stakeholders. As Dikert et al. (2016) note, “Frequent changes in priorities can disrupt the rhythm of agile development and lead to inefficiencies and stakeholder dissatisfaction.”<sup>8</sup>

These challenges are compounded when requirement changes are poorly documented or communicated. Poorly managed changes can lead to scope creep, where projects expand beyond their original intent without corresponding adjustments to time and cost estimates. Shenhar et al.

(2001) emphasize, “Scope creep is one of the primary reasons for cost overruns and delays in software projects.”<sup>22</sup>

### **3.3. Misalignment of Stakeholder Goals**

A key challenge in software development projects is the misalignment of stakeholder goals, which can manifest in differing priorities, expectations, and definitions of success. Taylor (2007) notes, “In outsourced projects, the supplier seeks to maximize profit, while the customer aims to minimize costs, creating an inherent tension that can jeopardize collaboration.”<sup>4</sup>

This misalignment is not limited to outsourcing scenarios. Internal stakeholders, such as technical teams, managers, and end-users, often have conflicting objectives. As Procaccino and Verner (2006) highlight, “Project success is frequently undermined by a lack of shared vision among stakeholders, leading to conflicting requirements and unclear priorities.”<sup>14</sup>

Effective stakeholder management is critical to resolving these conflicts. Clear communication, regular feedback loops, and the establishment of shared success criteria can help bridge gaps. Bannerman (2008) suggests, “By aligning stakeholder goals through collaborative planning and ongoing dialogue, teams can reduce misunderstandings and improve overall project outcomes.”<sup>3</sup>

## 4. Project Success vs. Project Management Success

The distinction between project success and project management success (PM success) is fundamental in understanding the nuanced outcomes of software development projects. While the two concepts are interlinked, they encompass different metrics, priorities, and evaluations.

### 4.1. Conceptual Distinctions

Project success and PM success are often conflated, yet they represent distinct dimensions of evaluation. As Munns and Bjeirmi (1996) argue, “Project success involves the achievement of broader objectives and customer satisfaction, while PM success focuses on the efficient execution of activities within the constraints of time, cost, and quality.”<sup>23</sup>

This distinction is further supported by Ika (2009), who states, “Good project management can contribute to project success, but it is neither a guarantee nor a necessary condition.”<sup>10</sup> For example, a project might fail to meet deadlines or budgetary goals but still achieve its strategic objectives, as seen in iconic cases like the Sydney Opera House. Jugdev and Müller (2005) emphasize, “Although the Opera House faced severe cost overruns and delays, it is now regarded as a cultural and architectural masterpiece.”<sup>12</sup>

The “iron triangle” of time, cost, and quality remains central to PM success but may not capture the long-term impacts and value delivered by a project. De Wit (1988) clarifies, “A project can succeed in meeting management objectives while failing to deliver the desired outcomes for its stakeholders.”<sup>11</sup>

### 4.2. The Role of Project Managers in Bridging Success Metrics

Project managers play a pivotal role in aligning PM success with broader project success metrics. Their ability to manage stakeholder expectations, adapt to changes, and prioritize effectively can determine whether a project achieves its intended value.

Müller and Turner (2007) assert, “The influence of project managers extends beyond operational efficiency to strategic alignment, particularly when their leadership style matches the needs of the project and its stakeholders.”<sup>24</sup> For instance, transformational leadership—characterized by clear vision and strong communication—has been shown to improve stakeholder satisfaction and project outcomes.

The project manager’s ability to balance technical execution with strategic foresight is critical, especially in complex or outsourced software projects. As Taylor (2007) notes, “In outsourced projects, the project manager must navigate conflicting priorities between cost control for the customer and profit maximization for the supplier.”<sup>4</sup> This dual focus requires robust risk

management, proactive communication, and a deep understanding of success criteria from both perspectives.

#### **4.3. Case Studies Highlighting the Distinction**

Case studies illustrate the practical implications of distinguishing between project success and PM success. One notable example is the study by Ahonen and Savolainen (2010), which analyzed five canceled software development projects. In one case, “The supplier delivered the project on time, yet the customer found the product unusable and abandoned it after paying the invoice.”<sup>25</sup> This underscores the importance of aligning deliverables with stakeholder needs rather than merely meeting contractual obligations.

Another illustrative example is the Denver International Airport’s baggage handling system. Although the system represented a technological marvel, it was plagued by delays, cost overruns, and operational inefficiencies. As Reel (1999) observes, “The baggage system is a cautionary tale of focusing on technical milestones without adequately addressing end-user requirements and system integration.”<sup>5</sup>

These cases highlight that while effective project management contributes to operational success, it cannot guarantee broader project success unless customer satisfaction, usability, and long-term value are prioritized.



## **5. Outsourced Software Development: A Supplier's Perspective**

### **5.1. Stakeholder Dynamics in Outsourcing**

Outsourced software development involves complex relationships between stakeholders, each with distinct objectives and priorities. These dynamics can significantly influence project success and risk management. Taylor (2007) highlights the inherent tension between customers, who aim to minimize costs and maximize value, and suppliers, who prioritize profit margins and operational efficiency.<sup>4</sup>

From the supplier's perspective, the need to balance customer expectations with organizational goals is critical. As Levina and Ross (2003) observe, "Suppliers must navigate a dual focus: delivering measurable value to the customer while maintaining internal profitability and resource utilization."<sup>17</sup> This duality creates challenges, particularly when customers and suppliers have misaligned success criteria.

In outsourced projects, stakeholder dynamics are further complicated by cultural and operational differences between the customer and the supplier. Dibbern et al. (2004) explain, "Misunderstandings can arise from differences in communication styles, work processes, and definitions of success, potentially jeopardizing project outcomes."<sup>15</sup> Effective collaboration and clear contractual agreements are necessary to mitigate these risks.

Furthermore, the perception of risks often varies between stakeholders. According to Bannerman (2008), "Suppliers tend to focus on operational risks, such as resource allocation and technical feasibility, while customers are more concerned with strategic risks, such as cost overruns and missed business opportunities."<sup>3</sup> These divergent perspectives require careful negotiation and continuous alignment throughout the project lifecycle.

The role of trust and relationship management cannot be overstated in outsourcing arrangements. As Goles and Chin (2005) note, "Building trust between stakeholders is essential for overcoming conflicts, ensuring transparency, and fostering a collaborative environment."<sup>26</sup> Trust enables suppliers to manage expectations more effectively and adapt to evolving project requirements without damaging the customer relationship.

The supplier's ability to proactively manage stakeholder dynamics can have a profound impact on project outcomes. As Taylor (2007) concludes, "Suppliers who invest in understanding their customer's strategic goals and align their operations accordingly are better positioned to achieve mutual success."<sup>4</sup>

### **5.2. Profitability vs. Customer Satisfaction**

Balancing profitability with customer satisfaction represents a critical challenge for suppliers in outsourced software development projects. These two objectives often conflict, as customers focus on maximizing value while suppliers aim to ensure financial returns. De Wit (1988)

highlights this tension, observing that “customers aim to minimize costs while suppliers seek to maximize profits, leading to a fundamental misalignment of priorities.”<sup>11</sup>

Profitability, from the supplier’s perspective, is achieved through efficient resource utilization, adherence to budget constraints, and achieving a positive return on investment. Bannerman (2008) emphasizes that “suppliers often prioritize financial metrics such as profit margins and revenue generation to evaluate the success of outsourced projects.”<sup>3</sup> However, a singular focus on profitability can undermine customer satisfaction, which is essential for long-term relationships and repeat business.

Customer satisfaction, on the other hand, revolves around the perceived value of the delivered solution, adherence to agreed timelines, and the quality of the software. Taylor (2007) notes, “Suppliers must understand and address customer expectations to ensure satisfaction, which often involves balancing scope flexibility with budgetary constraints.”<sup>4</sup> This balancing act requires a nuanced approach where suppliers must sometimes prioritize long-term customer relationships over short-term profits.

The concept of value-added services also plays a role in aligning profitability with customer satisfaction. As Levina and Ross (2003) argue, “Suppliers who deliver beyond the agreed scope by offering innovation and strategic insights can enhance customer satisfaction while securing additional revenue streams.”<sup>17</sup> By focusing on creating mutual value, suppliers can mitigate the risk of customer dissatisfaction even when profitability targets are prioritized.

However, tension arises when projects face challenges such as scope creep, unexpected delays, or cost overruns. According to Dibbern et al. (2004), “Suppliers often struggle to maintain profitability under these circumstances, leading to strained relationships and potential conflicts with customers.”<sup>15</sup> Effective communication and transparent negotiation are essential for resolving such conflicts without compromising either objective.

Ultimately, the interplay between profitability and customer satisfaction defines the supplier’s approach to project management in outsourced software development. As Bannerman (2008) concludes, “Achieving an optimal balance requires suppliers to adopt adaptive strategies, integrating customer feedback while ensuring financial viability.”<sup>3</sup>

### **5.3. Risk Management Practices in Outsourced Projects**

Effective risk management is a cornerstone of success in outsourced software development projects, particularly from the supplier’s perspective. Risks in outsourcing stem from a variety of factors, including misaligned goals, scope changes, and unforeseen technical challenges. According to Bannerman (2008), “Risk management involves identifying, assessing, and mitigating potential threats to project objectives, ensuring that suppliers can deliver within agreed constraints.”<sup>3</sup>

One significant risk in outsourcing arises from unclear or evolving project requirements. Taylor (2007) highlights, “Suppliers often face challenges when customers redefine requirements mid-project, leading to scope creep and associated risks.”<sup>4</sup> To address this, suppliers must employ flexible project management methodologies, such as Agile, which facilitate iterative development and better manage changing customer expectations. As Dikert et al. (2016) argue, “Agile methods allow for continuous risk assessment and mitigation, enabling suppliers to adapt quickly to evolving requirements.”<sup>8</sup>

Communication breakdowns also represent a critical risk in outsourced projects. Levina and Ross (2003) emphasize, “Miscommunication between suppliers and customers can lead to misunderstandings about deliverables, timelines, and quality expectations, ultimately jeopardizing the project’s success.”<sup>17</sup> Suppliers must invest in robust communication strategies, including regular updates, progress reports, and collaborative tools, to minimize this risk.

Financial risks, such as budget overruns or delays, can also affect suppliers significantly. Dibbern et al. (2004) note that “unexpected financial burdens can erode profitability, making it crucial for suppliers to adopt proactive financial monitoring and control mechanisms.”<sup>15</sup> Suppliers often use predictive analytics and historical project data to anticipate potential budgetary risks and take corrective action.

Cultural and geographical differences in international outsourcing add another layer of complexity. Taylor (2007) states, “Suppliers must navigate differences in work ethics, communication styles, and time zones to manage risks effectively.”<sup>4</sup> Suppliers often mitigate these challenges through cultural training, the establishment of local liaison offices, and employing overlapping work hours to facilitate collaboration.

Finally, relationship risks, such as loss of trust or conflicts, can derail projects. Bannerman (2008) asserts, “Strong relationship management is essential for mitigating interpersonal risks, as it ensures that both parties remain aligned despite challenges.”<sup>3</sup> Suppliers often designate dedicated account managers or relationship managers to maintain alignment and address issues proactively.

In conclusion, suppliers must adopt comprehensive risk management practices tailored to the unique challenges of outsourced software development. By leveraging adaptive methodologies, fostering transparent communication, and addressing cultural and financial risks, suppliers can not only mitigate potential threats but also enhance project outcomes and client satisfaction.

## 6. Critical Success Factors in Outsourced Software Projects

### 6.1. Effective Communication and Collaboration

Effective communication and collaboration are fundamental to the success of outsourced software projects. These factors serve as the backbone of trust and mutual understanding between the supplier and the client. According to Bannerman (2008), "Effective communication ensures that expectations are aligned, issues are promptly addressed, and project goals remain clear throughout the development lifecycle."<sup>3</sup>

One of the significant challenges in outsourced projects is the geographic and cultural distance that can hinder seamless communication. Taylor (2007) notes, "The physical and temporal separation between stakeholders often results in communication delays and misinterpretations, affecting project outcomes."<sup>4</sup> To counter this, suppliers and clients are increasingly adopting advanced communication tools, such as video conferencing, instant messaging, and collaborative platforms, which foster real-time interactions and minimize misunderstandings.

Collaboration is equally crucial for managing complex project dynamics. As Reel (1999) highlights, "Collaboration involves shared decision-making, joint problem-solving, and a commitment to achieving common objectives, which are vital for maintaining project momentum and mitigating risks."<sup>7</sup> Collaborative methodologies, such as Agile, are particularly effective in outsourced contexts, as they emphasize continuous stakeholder engagement through iterative feedback and regular sprint reviews.

Another vital aspect is the establishment of clear and structured communication channels. Levina and Ross (2003) argue, "Defining roles, responsibilities, and reporting hierarchies at the outset of the project helps in ensuring accountability and reduces the likelihood of conflicts."<sup>17</sup> Suppliers often designate account managers or communication liaisons to streamline information flow and facilitate timely resolution of concerns.

Moreover, fostering a culture of openness and transparency is critical for effective collaboration. As Dikert et al. (2016) observe, "When both parties maintain transparency about challenges, risks, and progress, it becomes easier to build trust and align efforts towards shared goals."<sup>8</sup> Suppliers and clients are encouraged to engage in joint planning sessions, risk assessments, and retrospectives to enhance collaboration.

Lastly, frequent and consistent updates are essential to maintaining alignment. Bannerman (2008) asserts, "Regular communication of project status, risks, and milestones keeps stakeholders informed and allows for quick course correction if needed."<sup>3</sup> This practice not only strengthens the partnership but also ensures that both parties are invested in the project's success.

In summary, effective communication and collaboration are critical for overcoming the inherent complexities of outsourcing. By leveraging modern communication tools, fostering transparency, and adopting collaborative frameworks, suppliers and clients can build strong, successful partnerships that drive positive outcomes in outsourced software projects.

## 6.2. Alignment of Goals Between Customers and Suppliers

The alignment of goals between customers and suppliers is a critical determinant of success in outsourced software projects. When both parties share a unified vision and mutually agreed-upon objectives, the likelihood of project success significantly increases. According to de Wit (1988), “Misalignment of goals can result in conflicting priorities, inefficiencies, and ultimately project failure.”<sup>11</sup> Ensuring that both customers and suppliers are aligned in their expectations from the outset is essential to avoid these pitfalls.

Suppliers often prioritize profitability, while customers focus on minimizing costs and achieving high-quality deliverables. Taylor (2007) observes, “This fundamental difference in objectives creates inherent tensions in outsourced projects, which, if unmanaged, can lead to disputes and dissatisfaction.”<sup>4</sup> Bridging this gap requires open dialogue during contract negotiations and the establishment of shared metrics for success.

A shared understanding of project success is essential for alignment. Bannerman (2008) emphasizes, “Clearly defined success criteria, agreed upon by both parties, provide a common framework for evaluating progress and ensuring mutual satisfaction.”<sup>3</sup> This includes setting realistic expectations regarding cost, timeline, and deliverables while accommodating potential changes during the project lifecycle.

To enhance goal alignment, effective governance mechanisms must be in place. Levina and Ross (2003) highlight, “Joint governance structures, such as steering committees, enable collaborative decision-making and ensure that the interests of both parties are represented throughout the project.”<sup>17</sup> These structures foster trust and accountability, mitigating the risk of misaligned priorities.

Additionally, adopting customer-centric approaches can help suppliers better align with client goals. As Dikert et al. (2016) note, “Suppliers who actively involve customers in planning, design, and decision-making processes are more likely to deliver outcomes that meet client expectations.”<sup>8</sup> Agile methodologies, which emphasize iterative feedback and regular engagement, are particularly effective in achieving this alignment.

Conflict resolution frameworks also play a vital role. Reel (1999) states, “When disagreements arise, having predefined mechanisms to address conflicts ensures that they do not derail the project.”<sup>5</sup> Establishing protocols for managing changes in scope or addressing performance issues can prevent misaligned goals from escalating into significant challenges.

Ultimately, the success of an outsourced software project depends on the ability of customers and suppliers to operate as partners with shared objectives. This requires ongoing effort, transparent communication, and a willingness to adapt to evolving project demands. By aligning their goals and working collaboratively, both parties can achieve outcomes that meet their respective needs and expectations.

### 6.3. Managing Evolving Requirements

Managing evolving requirements is one of the most critical aspects of ensuring success in outsourced software projects. Requirements often shift as projects progress, influenced by changing business needs, market conditions, and stakeholder priorities. According to Reel (1999), “One of the most common reasons for project failure is the inability to adapt to changing requirements, leading to misaligned deliverables and client dissatisfaction.”<sup>5</sup>

Evolving requirements challenge both customers and suppliers. While customers may expect flexibility, suppliers must balance accommodating changes with maintaining profitability. Dikert et al. (2016) highlight that “Agile methodologies offer a structured approach to handling evolving requirements, enabling iterative development and regular feedback loops.”<sup>8</sup> By adopting iterative models, suppliers can adapt to requirement changes while ensuring incremental value delivery.

A proactive approach to managing requirement changes is essential. As Bannerman (2008) notes, “Effective risk management practices should include mechanisms to assess and prioritize changes in requirements, ensuring that critical updates are implemented without disrupting the overall project timeline.”<sup>3</sup> This requires clear communication channels and robust change management processes.

The customer’s role in clarifying and validating requirements is also crucial. Procaccino and Verner (2006) observe, “Projects are more likely to succeed when customers are actively involved in refining requirements and providing feedback throughout the development process.”<sup>14</sup> Collaborative tools, such as shared documentation platforms and frequent review sessions, can facilitate this involvement.

Additionally, suppliers must recognize the financial and operational implications of evolving requirements. Taylor (2007) emphasizes, “The challenge for suppliers lies in accommodating changes while minimizing cost overruns and maintaining profitability.”<sup>4</sup> Establishing contractual agreements that include clauses for managing requirement changes, such as time-and-materials pricing models, can help address this challenge.

However, flexibility must be balanced with discipline. As de Wit (1988) asserts, “Uncontrolled scope changes can lead to scope creep, undermining the project’s success.”<sup>11</sup> Suppliers must enforce boundaries and ensure that changes align with the project’s original objectives and feasibility constraints.

In summary, managing evolving requirements demands adaptability, collaboration, and structured processes. By fostering a culture of communication and establishing clear protocols, both customers and suppliers can navigate changes effectively, ensuring that the project remains aligned with its goals and delivers value.

## 7. Frameworks and Standards for Measuring Success

### 7.1. ISO/IEC 12207 and Standardized Metrics

The adoption of standardized frameworks is essential in ensuring consistent and objective measures of success in software development projects. Among the most influential frameworks is the ISO/IEC 12207, which provides a comprehensive lifecycle process for software development. According to the standard, a project is defined as “an endeavor with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements” (ISO/IEC 12207, 2008).<sup>26</sup> This definition underscores the importance of structure and clarity in project planning and execution.

The ISO/IEC 12207 framework offers a unified set of processes that cover acquisition, supply, development, operation, and maintenance of software systems. Reel (1999) emphasizes that such standardized approaches “help organizations align their processes with best practices, improving the likelihood of delivering projects on time and within budget.”<sup>5</sup> By following ISO/IEC 12207, suppliers and customers can establish clear expectations and performance metrics.

One of the critical contributions of ISO/IEC 12207 is its emphasis on measurable criteria for project success. It aligns with the “iron triangle” framework of time, cost, and scope, but it also incorporates broader quality considerations. Bannerman (2008) notes, “Standardized metrics provide a baseline for evaluating whether a project has achieved its objectives, fostering accountability and transparency.”<sup>3</sup> These metrics include defect density, schedule variance, and cost performance indices, which collectively paint a detailed picture of project health.

In outsourced software projects, ISO/IEC 12207 also facilitates better collaboration between stakeholders. As Taylor (2007) observes, “Standards like ISO/IEC 12207 provide a common language for customers and suppliers to discuss expectations, risks, and deliverables.”<sup>4</sup> This commonality reduces misunderstandings and promotes a shared vision of success.

Furthermore, the framework’s adaptability makes it suitable for diverse projects, from traditional waterfall models to Agile methodologies. Dikert et al. (2016) highlight that “Even in Agile environments, standardized metrics can coexist with iterative processes, ensuring that both flexibility and accountability are maintained.”<sup>8</sup>

However, while ISO/IEC 12207 offers valuable guidance, its implementation is not without challenges. De Bakker et al. (2010) caution that “over-reliance on formal metrics can sometimes overshadow the need for subjective evaluations, such as stakeholder satisfaction and long-term benefits.”<sup>21</sup> Balancing standardized measures with qualitative assessments is critical to achieving a holistic view of project success.

In summary, ISO/IEC 12207 provides a robust framework for defining and measuring success in software development projects. By integrating standardized metrics with flexible processes, it helps organizations navigate the complexities of modern software development while fostering collaboration and accountability.

## 7.2. Adapting Frameworks to Subjective and Evolving Criteria

While standardized frameworks like ISO/IEC 12207 provide objective measures for project success, adapting these frameworks to accommodate subjective and evolving criteria is crucial in the dynamic landscape of software development. As Mahaney and Lederer (2010) observe, “Success is not only about meeting predefined metrics but also about satisfying stakeholder expectations, which are often fluid and subjective.”<sup>27</sup> This duality highlights the need for frameworks that can balance rigidity with flexibility.

Frameworks such as PRINCE2 and Agile address some of these challenges by incorporating iterative feedback mechanisms. PRINCE2, for instance, includes stages for continual assessment, enabling project managers to realign objectives based on emerging requirements. As Serrador and Pinto (2015) explain, “By emphasizing adaptability, modern frameworks allow organizations to respond effectively to evolving customer needs while maintaining a structured approach to management.”<sup>28</sup>

Subjective criteria, such as user satisfaction and stakeholder alignment, require metrics beyond traditional cost, time, and scope. Bannerman (2008) highlights that “Customer satisfaction, while difficult to quantify, often serves as a better indicator of project success than rigid adherence to pre-set objectives.”<sup>3</sup> This perspective encourages organizations to incorporate qualitative assessments into their success measures.

In outsourced software projects, the need to adapt frameworks to subjective criteria is even more pronounced due to the diversity of stakeholder perspectives. Taylor (2007) notes that “Suppliers and customers often have different definitions of success, and reconciling these definitions requires frameworks that accommodate subjective evaluation.”<sup>4</sup> For example, a supplier may focus on profitability and technical quality, while the customer prioritizes usability and timely delivery.

Evolving criteria also present challenges in project evaluation. In Agile methodologies, where requirements are frequently revisited, frameworks must accommodate changing priorities without undermining accountability. Dikert et al. (2016) state, “Agile practices, when supported by adaptable frameworks, can integrate subjective and evolving criteria into iterative cycles, ensuring that projects remain aligned with stakeholder expectations.”<sup>8</sup>

Moreover, frameworks like the Balanced Scorecard (BSC) offer a structured way to incorporate subjective and evolving factors into evaluations. By measuring customer satisfaction, process quality, and innovation alongside financial metrics, the BSC provides a multi-dimensional view of success. Kaplan and Norton (1996) argue that “A balanced approach to performance measurement allows organizations to capture both tangible and intangible aspects of success.”<sup>29</sup>

However, implementing adaptable frameworks is not without difficulties. De Bakker et al. (2010) warn that “The inclusion of subjective and evolving criteria increases the complexity of project management, necessitating skilled leadership and continuous stakeholder engagement.”<sup>21</sup> Effective adaptation requires organizations to strike a balance between accommodating change and maintaining accountability.



In conclusion, adapting frameworks to subjective and evolving criteria enhances their relevance in software development projects. By integrating flexibility into their structures, organizations can address the nuanced and dynamic nature of success while maintaining a foundation of objective evaluation.

### **7.3. Examples of Successful and Failed Projects**

Examining real-world examples of both successful and failed software projects can offer valuable insights into how frameworks and standards for measuring success can influence outcomes. These examples highlight the importance of adapting frameworks to suit the dynamic and multifaceted nature of software development.

One widely discussed example of a successful software project is the development of the iPhone by Apple. Apple's success with the iPhone can be attributed to its focus on innovation, user experience, and adaptability to market demands. According to Cusumano and Nobeoka (1998), "Apple's ability to continuously iterate on the design, integrating customer feedback into each version of the product, was key to its market success."<sup>30</sup> This approach aligns with Agile methodologies, where continuous iteration and adaptation to changing requirements play a crucial role in project success. Apple's success was also due to strong stakeholder alignment, as it consistently met or exceeded the expectations of both consumers and business partners.

In contrast, the failure of the Denver International Airport's baggage handling system illustrates the consequences of unrealistic project estimation and poor stakeholder communication. According to Gable (1997), "The project suffered from overoptimistic estimations, lack of adequate testing, and an absence of coordination among stakeholders, leading to delays and increased costs."<sup>31</sup> This project was plagued by scope creep and misaligned expectations between the airport authorities and the technology vendors, which ultimately led to its failure. Gable highlights that "the absence of clear metrics for project success, combined with an underestimation of technical challenges, contributed to the project's downfall."<sup>31</sup>

Another example of a failed project is the development of the UK's National Health Service (NHS) IT system, which was initially launched in the early 2000s. Despite a significant investment, the project faced numerous difficulties, including a lack of clarity in requirements, misaligned goals between stakeholders, and technological challenges. According to Taylor (2013), "The NHS IT project failed primarily because of its inability to address the changing needs of the healthcare system and its failure to adapt to evolving requirements."<sup>32</sup> Taylor argues that the lack of adaptability in the framework used to measure success contributed to the project's demise, as stakeholders were unable to shift the project's direction in response to emerging issues.

On the other hand, a more recent example of a successful project is the development of Spotify's music streaming service. Spotify's success can be attributed to its clear focus on customer satisfaction and its ability to pivot based on user feedback. As Ojala and Tyrväinen (2014) state, "Spotify's success was driven by its commitment to user-centric design and its ability to quickly

adapt its service based on market demands.”<sup>33</sup> This project was able to align its goals with both customer needs and business objectives, resulting in a highly successful product.

In both successful and failed projects, it becomes clear that frameworks for measuring success must be flexible and responsive to the evolving nature of software development. While traditional metrics such as time, cost, and scope remain important, they cannot capture the full picture of success in software projects. As Martens et al. (2011) point out, “The failure to adapt success metrics to the unique needs of each project is a key factor in project failure.”<sup>34</sup> Frameworks must incorporate both objective and subjective criteria to ensure that project outcomes align with the diverse and evolving expectations of stakeholders.

In conclusion, real-world examples of both success and failure highlight the need for adaptable frameworks and standards that can account for the complexity and fluidity of software development projects. By incorporating flexibility into success metrics, organizations can better navigate the challenges of software development and increase their chances of project success.

## 8. Lessons Learned and Recommendations

### 8.1. Importance of Mutual Understanding

One of the most critical lessons learned from successful and failed software development projects is the necessity of mutual understanding among stakeholders. This includes customers, suppliers, project managers, and end-users. Without a shared vision and understanding of objectives, even well-funded and technically sound projects can falter.

According to Turner (2009), “Mutual understanding is foundational to project success, as it fosters collaboration, reduces misunderstandings, and aligns all parties towards common goals.”<sup>35</sup> This alignment ensures that stakeholders are working cohesively, minimizing risks associated with miscommunication or conflicting objectives. Turner further emphasizes that clarity in roles, responsibilities, and expectations can reduce ambiguities that often derail projects.

The Agile methodology is an exemplary framework that prioritizes mutual understanding. By encouraging regular collaboration and feedback between teams and stakeholders, Agile ensures that evolving requirements are well understood. Highsmith (2001) states, “Agile practices facilitate ongoing dialogue, ensuring that all stakeholders remain aligned with the project’s trajectory.”<sup>36</sup> Projects that implement Agile principles often find it easier to adapt to changes while maintaining stakeholder trust and satisfaction.

In outsourced software development, mutual understanding becomes even more critical due to differences in organizational cultures, geographic distances, and time zones. Misaligned expectations in such projects often stem from inadequate communication. As noted by Hofstede (1984), “Cultural differences can lead to misunderstandings, which, if not addressed early, may result in project delays and stakeholder dissatisfaction.”<sup>37</sup> Therefore, fostering cultural intelligence and open communication is vital for success.

A significant factor that contributes to mutual understanding is the documentation of project goals, deliverables, and criteria for success. According to Kerzner (2013), “Clear and accessible documentation creates a shared knowledge base that all stakeholders can refer to, reducing the risk of misinterpretation.”<sup>38</sup> This includes detailing project scope, timelines, and change management processes in a manner that is transparent to all involved parties.

The use of visualization tools such as dashboards or Gantt charts can also aid in promoting mutual understanding. These tools provide real-time updates on project status, enabling stakeholders to track progress and address potential issues collaboratively. As suggested by Milosevic (2003), “Visualization tools help bridge the communication gap, particularly in large or complex projects, by presenting information in a comprehensible and actionable format.”<sup>39</sup>

In conclusion, fostering mutual understanding is essential for the success of software development projects. This involves open communication, cultural sensitivity, clear documentation, and the adoption of collaborative frameworks such as Agile. By prioritizing

these aspects, organizations can enhance stakeholder alignment, reduce conflicts, and create an environment conducive to successful project delivery.

## **8.2. Balancing Short-Term and Long-Term Objectives**

An essential challenge in software development projects is balancing immediate deliverables with the broader vision of long-term success. While short-term objectives often focus on meeting deadlines, budget constraints, and deliverable quality, long-term goals prioritize scalability, maintainability, and alignment with organizational strategy. Failure to balance these perspectives can lead to projects that succeed temporarily but struggle in the long term.

According to Shenhar and Dvir (2007), “Successful projects are those that achieve both immediate success and create value over time. This requires a dual focus on short-term efficiency and long-term effectiveness.”<sup>40</sup> Their research highlights that organizations often emphasize immediate performance metrics, inadvertently sacrificing future adaptability and sustainability.

One way to achieve this balance is through iterative development methodologies, such as Agile or Scrum. These frameworks enable teams to address short-term objectives through sprints while keeping the long-term vision intact via backlog refinement and strategic planning. Highsmith (2009) argues, “Agile provides a mechanism to prioritize short-term needs without losing sight of the long-term product roadmap.”<sup>41</sup>

Balancing objectives is particularly critical in outsourced software projects. Suppliers may prioritize short-term profitability or milestone delivery to meet contractual obligations, whereas clients may focus on long-term usability and scalability. As suggested by Overby (2008), “Aligning supplier and client priorities requires frequent communication and shared accountability for both immediate and future project goals.”<sup>42</sup> Establishing transparent performance metrics that account for both short- and long-term factors is a recommended best practice.

The importance of striking this balance is further underscored when considering technical debt. Quick fixes or shortcuts taken to meet short-term deadlines can accumulate technical debt, making future iterations of the software more costly and time-consuming to implement. McConnell (2004) notes, “Investing in quality early in the project lifecycle, even at the expense of short-term speed, pays dividends in long-term project sustainability.”<sup>43</sup>

Risk management also plays a significant role in this balancing act. Focusing too heavily on short-term gains can increase risks, such as poor system integration or insufficient testing. Conversely, excessive focus on long-term planning may delay the delivery of key functionalities. According to Kerzner (2013), “Risk management frameworks must account for both short-term project milestones and long-term organizational objectives to ensure a holistic approach to success.”<sup>44</sup>

In conclusion, achieving a balance between short-term and long-term objectives requires adopting methodologies that facilitate iterative progress, fostering transparent communication, and addressing technical debt early. By prioritizing both immediate outcomes and future goals, software development projects can deliver value that is both timely and enduring.

### **8.3. Strategies for Improving Estimation and Goal Alignment**

Improving estimation accuracy and aligning project goals with stakeholders' expectations are critical for software development success. Inaccurate estimations can lead to schedule overruns, budget excesses, and dissatisfaction among stakeholders, while misaligned goals often result in conflicts that compromise project outcomes. Effective strategies address these challenges by combining technical rigor with collaborative processes.

According to Jørgensen and Moløkken (2004), "Estimation errors in software projects are often due to optimism bias, lack of data, and inadequate stakeholder involvement."<sup>45</sup> To mitigate these issues, organizations can adopt evidence-based estimation techniques such as historical data analysis, reference class forecasting, and parametric modeling. These methods rely on empirical evidence rather than subjective judgment, reducing bias and improving reliability.

Collaborative goal alignment is equally vital. Research by Turner and Müller (2004) emphasizes that "Project success depends on the extent to which all stakeholders, including customers, sponsors, and team members, share a unified understanding of objectives and success criteria."<sup>46</sup> Facilitating workshops and regular stakeholder meetings early in the project lifecycle can help align expectations and clarify priorities.

Agile and hybrid project management methodologies also contribute significantly to better estimation and alignment. Highsmith (2009) argues, "Agile frameworks, by focusing on iterative delivery and continuous stakeholder feedback, allow teams to adjust estimations and refine goals dynamically."<sup>47</sup> Such adaptive approaches accommodate evolving requirements and provide flexibility to address unforeseen challenges without derailing the project.

Furthermore, adopting advanced tools and technologies can streamline estimation processes. Tools like Monte Carlo simulations, machine learning-based prediction models, and integrated project management platforms provide data-driven insights that enhance precision. As stated by Boehm (2000), "Modern estimation tools leverage historical patterns and statistical analysis to reduce uncertainty and improve decision-making."<sup>48</sup>

Clear communication is essential for improving alignment between stakeholders. According to DeMarco and Lister (2013), "Communication breakdowns, rather than technical failures, are often the root cause of project misalignment."<sup>49</sup> Techniques such as visual roadmaps, detailed project charters, and regular progress updates can bridge communication gaps and maintain alignment throughout the project.

Finally, fostering a culture of accountability and collaboration strengthens both estimation and goal alignment. McConnell (2004) notes, “Teams that involve all relevant stakeholders in the estimation process and maintain transparency about trade-offs and risks are more likely to achieve balanced and realistic project goals.”<sup>43</sup> Collaborative processes build trust, ensure stakeholder buy-in, and minimize friction.

In conclusion, improving estimation accuracy and goal alignment requires a combination of rigorous methodologies, adaptive practices, advanced tools, and effective communication. By implementing these strategies, organizations can enhance their ability to deliver successful software projects that meet both short-term and long-term objectives.

## **9. Conclusion**

### **9.1. Summary of Findings**

The complexities of software development, particularly in outsourced projects, highlight the multifaceted nature of defining and achieving success. This essay explored the distinction between project success and project management success, emphasizing how traditional frameworks like the "Iron Triangle" often fail to capture the evolving expectations of stakeholders. Key challenges, including unrealistic estimations, changing requirements, and goal misalignment, were identified as major contributors to project failures. Moreover, strategies such as improved communication, collaborative goal alignment, and advanced estimation techniques were proposed as solutions to these challenges. The importance of frameworks like ISO/IEC 12207 was underscored for standardizing metrics while recognizing the need to adapt to subjective and evolving criteria.

### **9.2. Relevance of Supplier Perspectives in Future Research**

The supplier's perspective in outsourced software development projects remains underexplored but is increasingly relevant. As outsourcing becomes more prevalent, understanding how suppliers balance profitability with customer satisfaction, manage risks, and address stakeholder dynamics is crucial. Future research must delve deeper into supplier-side challenges, such as resource constraints, cultural differences, and communication barriers, to create a holistic view of project success. According to Ramasubbu et al. (2008), "Supplier performance metrics must evolve to include dimensions beyond cost and schedule, such as innovation, adaptability, and long-term collaboration."<sup>50</sup> Incorporating supplier perspectives can lead to more equitable and sustainable outsourcing relationships, benefiting all stakeholders.

### **9.3. Call for Comprehensive Success Metrics**

The essay concludes with a call for the development of comprehensive success metrics that go beyond time, cost, and scope. As Shenhar et al. (2001) argue, "Project success should be evaluated on multiple dimensions, including business impact, stakeholder satisfaction, and strategic alignment."<sup>51</sup> Adopting multidimensional metrics that account for long-term objectives, customer satisfaction, and adaptability to changes will provide a more accurate measure of success. This approach requires collaboration among researchers, practitioners, and standards organizations to design frameworks that are both rigorous and flexible enough to address the complexities of modern software development.

By incorporating these perspectives and methodologies, the software development industry can move toward a more inclusive and robust understanding of success, paving the way for higher project completion rates, better stakeholder relationships, and sustainable growth.





## 10. References

- <sup>1</sup> Cerpa, N., & Verner, J. M. (2009). "Why Did Your Project Fail?" *Communications of the ACM*.
- <sup>2</sup> Glass, R. L. (2001). "Frequently Forgotten Fundamental Facts About Software Development." *IEEE Software*.
- <sup>3</sup> Bannerman, P. L. (2008). "Risk and Risk Management in Software Projects: A Reassessment." *Journal of Systems and Software*.
- <sup>4</sup> Taylor, H. (2007). "Outsourced IT Projects from the Vendor Perspective: Different Goals, Different Risks." *Journal of Global Information Management*.
- <sup>5</sup> Reel, J. S. (1999). "Critical Success Factors in Software Projects." *IEEE Software*.
- <sup>6</sup> Glass, R. L. (2001). "Frequently Forgotten Fundamental Facts About Software Development." *IEEE Software*.
- <sup>7</sup> Reel, J. S. (1999). "Critical Success Factors in Software Projects." *IEEE Software*.
- <sup>8</sup> Dikert, T., Paasivaara, M., & Lassenius, C. (2016). "Risk Management in Agile Software Development: A Survey." *IEEE Software*.
- <sup>9</sup> Munns, A. K., & Bjeirmi, B. F. (1996). "The Role of Project Management in Achieving Project Success." *International Journal of Project Management*.
- <sup>10</sup> Ika, L. A. (2009). "Project Success as a Topic in Project Management Journals." *Project Management Journal*.
- <sup>11</sup> De Wit, A. (1988). "Measurement of Project Success." *International Journal of Project Management*.
- <sup>12</sup> Jugdev, K., & Müller, R. (2005). "A Retrospective Look at Our Evolving Understanding of Project Success." *Project Management Journal*.
- <sup>13</sup> Papke-Shields, K. E., Beise, C. M., & Quan, J. (2010). "Do Project Managers Practice What They Preach, and Does It Matter to Project Success?" *International Journal of Project Management*.
- <sup>14</sup> Procaccino, J. D., & Verner, J. M. (2006). "Software Project Managers and Project Success: An Exploratory Study." *Journal of Systems and Software*.
- <sup>15</sup> Dibbern, J., Goles, T., Hirschheim, R., & Jayatilaka, B. (2004). "Information Systems Outsourcing: A Survey and Analysis of the Literature." *ACM Computing Surveys*.
- <sup>16</sup> Jun, L., Qiuzhen, W., & Qingguo, M. (2010). "The Effects of Project Uncertainty and Risk Management on IS Development Project Performance: A Vendor Perspective." *International Journal of Project Management*.
- <sup>17</sup> Levina, N., & Ross, J. W. (2003). "From the Vendor's Perspective: Exploring the Value Proposition in Information Technology Outsourcing." *MIS Quarterly*.
- <sup>18</sup> Atkinson, R. (1999). "Project Management: Cost, Time, and Quality, Two Best Guesses and a Phenomenon." *International Journal of Project Management*.
- <sup>19</sup> Kappelman, L. A., McKeeman, R., & Zhang, L. (2006). "Early Warning Signs of IT Project Failure: The Dominant Dozen." *Information Systems Management*.
- <sup>20</sup> Lai, V. S. (1997). "Critical Success Factors for IS Development: A Meta-Analytic Review." *Journal of Management Information Systems*.

- <sup>21</sup> De Bakker, K., Boonstra, A., & Wortmann, H. (2010). "Does Risk Management Contribute to IT Project Success? A Meta-Analysis of Empirical Evidence." *International Journal of Project Management*.
- <sup>22</sup> Shenhar, A. J., Dvir, D., Levy, O., & Maltz, A. C. (2001). "Project Success: A Multidimensional Strategic Concept." *Long Range Planning*.
- <sup>23</sup> Munns, A. K., & Bjeirmi, B. F. (1996). "The Role of Project Management in Achieving Project Success." *International Journal of Project Management*.
- <sup>24</sup> Müller, R., & Turner, R. (2007). "The Influence of Project Managers on Project Success Criteria." *European Management Journal*.
- <sup>25</sup> Ahonen, J. J., & Savolainen, P. (2010). "Software Engineering Projects May Fail Before They Are Started: Post-Mortem Analysis of Five Cancelled Projects." *Journal of Systems and Software*.
- <sup>26</sup> ISO/IEC 12207 (2008). *Systems and Software Engineering – Software Life Cycle Processes*. International Organization for Standardization.
- <sup>27</sup> Mahaney, R. C., & Lederer, A. L. (2010). "The Role of Monitoring and Shaming in IT Project Management." *International Journal of Project Management*.
- <sup>28</sup> Serrador, P., & Pinto, J. K. (2015). "Does Agile Work? A Quantitative Analysis of Agile Project Success." *International Journal of Project Management*.
- <sup>29</sup> Kaplan, R. S., & Norton, D. P. (1996). "Using the Balanced Scorecard as a Strategic Management System." *Harvard Business Review*.
- <sup>30</sup> Cusumano, M. A., & Nobeoka, K. (1998). "Thinking Beyond Lean: The Key to Building Successful Software Projects." *Harvard Business Review*.
- <sup>31</sup> Gable, G. G. (1997). "Success and Failure in Software Projects: What We Know and What We Don't Know." *Journal of Systems and Software*.
- <sup>32</sup> Taylor, P. (2013). "The NHS IT Project: A Case Study in Failure." *Journal of Information Technology*.
- <sup>33</sup> Ojala, A., & Tyrväinen, P. (2014). "Spotify and the Changing Landscape of Digital Music." *International Journal of Information Management*.
- <sup>34</sup> Martens, M. L., et al. (2011). "The Role of Adaptability in Software Development Projects." *International Journal of Project Management*.
- <sup>35</sup> Turner, R. J. (2009). *The Handbook of Project-Based Management: Leading Strategic Change in Organizations*. McGraw-Hill Education.
- <sup>36</sup> Highsmith, J. (2001). *Agile Software Development Ecosystems*. Addison-Wesley Professional.
- <sup>37</sup> Hofstede, G. (1984). *Culture's Consequences: International Differences in Work-Related Values*. SAGE Publications.
- <sup>38</sup> Kerzner, H. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
- <sup>39</sup> Milosevic, D. (2003). *Project Management Toolbox: Tools and Techniques for the Practicing Project Manager*. Wiley.
- <sup>40</sup> Shenhar, A. J., & Dvir, D. (2007). *Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation*. Harvard Business Review Press.
- <sup>41</sup> Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.

- <sup>42</sup> Overby, S. (2008). "The Hidden Costs of Offshore Outsourcing." *CIO*.
- <sup>43</sup> McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press.
- <sup>44</sup> Kerzner, H. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
- <sup>45</sup> Jørgensen, M., & Moløkken, K. (2004). "How Large Are Software Cost Overruns? A Review of the 1994 CHAOS Report." *IEEE Software*.
- <sup>46</sup> Turner, J. R., & Müller, R. (2004). "Communication and Co-operation on Projects Between the Project Owner as Principal and the Project Manager as Agent." *Project Management Journal*.
- <sup>47</sup> Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.
- <sup>48</sup> Boehm, B. W. (2000). *Software Cost Estimation with COCOMO II*. Prentice Hall.
- <sup>49</sup> DeMarco, T., & Lister, T. (2013). *Peopleware: Productive Projects and Teams*. Addison-Wesley.
- <sup>50</sup> Ramasubbu, N., et al. (2008). "Managing Risks in Global Software Development Projects: A Supplier's Perspective." *IEEE Transactions on Software Engineering*.
- <sup>51</sup> Shenhar, A. J., et al. (2001). "Mapping the Dimensions of Project Success" in *Project Management Journal*.

