Revisiting IT Project Uncertainty: Operationalizing the Sambamurthy–Zmud Model for De-Risking Digital Transformation Projects

Irshad Abdulla[1]

[1]Beedie School of Business, Simon Fraser University, Vancouver, Canada
irshad_abdulla@sfu.ca

Abstract

Uncertainty in project deliverables remains a pervasive source of IT project failure, yet its structural origins are rarely operationalized. Sambamurthy and Zmud (2017) proposed a conceptual model linking IT project uncertainty to two fundamental dimensions: (1) the clarity or ambiguity of project deliverable specifications and (2) the number, diversity, and power of stakeholders involved. Despite its strong resonance with practice, this framework has not been empirically developed or tested.

This paper extends and operationalizes the Sambamurthy–Zmud model by defining measurable constructs for project deliverable specification clarity and stakeholder structure complexity and by theorizing their joint effect on IT project risk. Drawing on information processing theory, stakeholder theory, and socio-technical systems perspectives, the paper argues that IT project uncertainty is not merely a descriptive condition, but a primary driver of project risk. A conceptual model and testable hypotheses are proposed to guide future empirical research and managerial practice in digital transformation projects.

Keywords

Digital Transformation, Technology Project Uncertainty, Technology Project Risk, Deliverable Specification, Project Stakeholders

1. Introduction

Information technology (IT) projects continue to experience high rates of failure, budget overruns, and benefit shortfalls. Decades of empirical and practitioner studies identify “uncertainty” as one of the most persistent causes of risk (Lyytinen et al., 1998; Wallace et al., 2004; Tiwana & Keil, 2004). Yet, the term *uncertainty* remains conceptually diffuse and operationally underdeveloped.

Sambamurthy and Zmud (2017) proposed a compelling and practice-aligned model that frames IT project uncertainty along two dimensions: (1) the *clarity or ambiguity of project deliverable specifications* and (2) the *number, diversity, and power of stakeholders* who influence project outcomes. According to this model, projects characterized by ambiguous deliverables and complex stakeholder structures exhibit higher levels of uncertainty, which in turn heightens project risk.

While their framework provides a clear conceptual foundation, it remains largely pedagogical — introduced in their book *Guiding the Digital Transformation of Organizations* — and has not yet been elaborated or validated in the academic literature. This paper seeks to extend that model by defining the constructs, grounding them in established theories, and linking them explicitly to project risk outcomes.

The central premise is that *reducing uncertainty reduces risk*. Clarifying deliverable specifications and managing stakeholder structure structure complexity are therefore not administrative niceties, but core mechanisms of *de-risking* IT projects. The research question motivating this study is:

> **How do deliverable specification clarity and stakeholder structure complexity jointly influence IT project risk?**

2. Theoretical Background

2.1 IT Project Risk and Uncertainty

Risk in IT projects is commonly defined as the likelihood and impact of adverse events affecting project objectives (Barki et al., 1993). Uncertainty, by contrast, refers to the *lack of clarity or predictability* regarding those objectives and events. In project management research, uncertainty is viewed as a precursor to risk: ambiguous or unstable conditions increase the probability of negative outcomes (Chapman & Ward, 2003).

IT projects, particularly large-scale digital transformation initiatives, operate under high uncertainty because deliverables evolve alongside emerging technologies and shifting organizational needs. Requirements often remain partially specified or contested across stakeholder groups, creating fertile ground for scope creep, design rework, and schedule delays.

2.2 The Sambamurthy–Zmud Model of Project Uncertainty

Sambamurthy and Zmud (2017) position IT project uncertainty within a two-dimensional framework:

1. **Clarity vs. Ambiguity in Deliverable Specifications** – the degree to which project outputs are well-defined and understood.
2. **Number / Diversity of Stakeholders** – the structural complexity of the stakeholder environment influencing the project.

Projects with *high ambiguity* and *high stakeholder structure complexity* are expected to experience the greatest uncertainty and therefore the highest risk. This framework offers a powerful lens for examining project failure causes but has not been translated into measurable constructs or empirically validated.

This paper advances the model by defining each dimension operationally and theorizing their interaction through established organizational theories.

3. Theoretical Foundations

3.1 Information Processing Theory (Galbraith, 1973)

Organizations must process information to resolve uncertainty. The amount of information processing required increases with task ambiguity and interdependence. Deliverable specification clarity reduces the need for information processing; ambiguity increases it. Projects overloaded with unprocessed information are more prone to coordination failure and risk.

3.2 Stakeholder Theory (Freeman, 1984; Mitchell, Agle & Wood, 1997)

Projects involve multiple stakeholders with varying salience (power, legitimacy, urgency). As stakeholder diversity and power asymmetry increase, alignment becomes harder, leading to contested interpretations of deliverables. This misalignment magnifies the impact of specification ambiguity.

3.3 Socio-Technical Systems Theory

Projects succeed when the technical (specifications, systems, methods) and social (stakeholders, governance, communication) subsystems are aligned. Misalignment—such as ambiguous deliverables coupled with fragmented stakeholder interests—creates systemic project risk.

Together, these theories help explain how ambiguity and stakeholder structure complexity interact to produce risk through elevated information processing demands, goal misalignment, and socio-technical imbalance.

4. Conceptual Model

The proposed model conceptualizes *IT Project Risk* as an outcome of *IT Project Uncertainty*, which emerges from the interaction between **deliverable specification clarity (DSC)** and **stakeholder structure complexity (SSC)**.

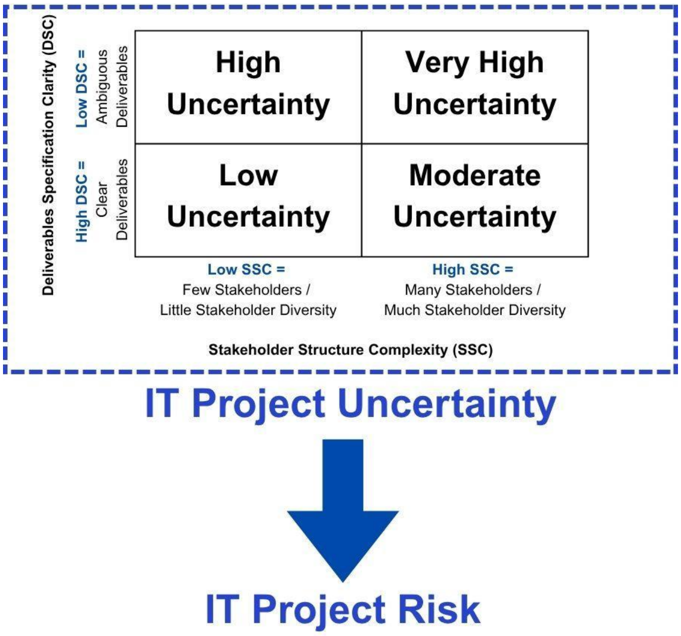

Figure 1. Conceptual Model of IT Project Risk as an outcome of IT Project Project Uncertainty (Adapted from Sambamurthy and Zmud, 2017)

When **DSC is low** (ambiguous deliverables), projects face greater risk due to misinterpretation of what actually needs to be delivered. When **SSC is high** (large number of stakeholders, or stakeholders representing varying interests / parts of the organization), coordination challenges / meeting competing needs of stakeholders amplify the impact of any deliverable ambiguity. Thus SSC by itself is not of significance, the moderating impact of SSC on DSC is relevant. Conversely, **high DSC and low SSC** conditions yield lower uncertainty and reduced project risk.

5. Construct Definitions and Operationalization

This section defines the key constructs underpinning the model of IT project uncertainty adapted from **Sambamurthy & Zmud (2017)**. Following their proposition that *uncertainty in project deliverable specifications* and *stakeholder structure complexity* increase IT project risk, we expand each construct and provide preliminary operationalization guidance for empirical testing.

5.1 Project Deliverable Specifications

Definition

Project deliverable specifications (also referred to plainly as deliverables) are the formal outputs that define what an IT project must produce to achieve its intended outcomes. In an IT project, each project deliverable serves a particular purpose that aids in achieving project objectives in some form or another; and there may be relationships between deliverables such that one deliverable informs or is a prerequisite for another. Each deliverable typically comprises different content required to satisfy the objectives of that particular deliverable. For example project charter deliverables may detail a set of decisions about project scope, functional design specifications may include descriptions of how the system functionality will be set up, and data design deliverables describe decisions about what data will be migrated from the old system to the new system being implemented in the project.

Uncertainty in deliverable specifications arises from incomplete, inconsistent, or ambiguous definitions of "what success looks like" (Wallace, Keil & Rai, 2004; Tiwana & Keil, 2007). Reducing ambiguity in these deliverables is central to de-risking large-scale IT projects (Sambamurthy & Zmud, 2017).

Building on Sambamurthy & Zmud's (2017) proposition that uncertainty in project deliverable specifications contributes to IT project risk, we can operationalize this construct by identifying the *key deliverables* in large-scale technology projects and examining how clarity or ambiguity manifests in each. Table 1 in Appendix A synthesizes findings from the IS project management and systems engineering literature, combined with practical insights from large-scale ERP and digital transformation programs (e.g., Tiwana & Keil, 2007; Nelson, 2007; Barki et al., 2001; Keil et al., 2013). It lists and describes key deliverables in a digital transformation project, along with potential sources of ambiguity and mechanisms that can be applied to reduce such ambiguity.

Operationalization of Deliverable Specification Clarity (DSC)

Deliverable Specification Clarity (DSC) is conceptualized as a multi-dimensional construct comprising perceived clarity, documented ambiguity, and contextual interpretation. To assess this construct in the context of IT projects, three complementary components are proposed: the *Deliverable Clarity Index (DCI)*, the *Ambiguity Ratio (AR-0)*, and *Qualitative Indicators (QI)*. Each serves a distinct yet reinforcing analytical role.

Deliverable Clarity Index (DCI) is a qualitative composite indicator measuring how clear, complete, specific, and consistent key project deliverables are in the eyes of project participants. It reflects the shared understanding of deliverables among project participants. DCI can be measured as a mean of Likert-scale survey responses (1–5) on items assessing core deliverables listed in Appendix A (e.g., project charter, requirements, data, integration, training, change management). Data for DCI can be gathered by surveying project team members, business stakeholders, and system integrators.

The Initial Ambiguity Ratio (AR-0) can be seen as a diagnostic indicator quantifying the extent of unresolved, conflicting, or incomplete elements in deliverable documentation. It captures "objective" ambiguity present in project artefacts. This is a quantitative indicator counting ambiguous, conflicting, or incomplete items ÷ total documented items. AR-0 data can be gathered through project requirements logs, issue trackers, meeting minutes and design documentation. Gaps in AR-0 can manifest if a deliverable has never been developed in the project in the first place and hence does not show up as a denominator in the "incomplete items ÷ total documented items" AR-0 calculation. Similarly, if the deliverable is developed but is missing important content which project participants do not realize, these ambiguities are never documented in the issue tracker, meeting minutes etc. and may not show up in the numerator of "incomplete items ÷ total documented items" AR-0 calculation.

Qualitative Indicators (QI) are thematic and narrative evidence providing contextual understanding of how clarity or ambiguity manifests in practice. It explains the *why* behind DCI and AR values. Measures include thematic coding of qualitative data to identify instances of misunderstanding, rework, or disagreement regarding deliverables. Interviews, project retrospectives, industry expertise and lessons-learned documentation are potential data sources. QI can be used to compensate for potential data gaps in AR-0 described above.

Integration of DSC Components

DCI, AR-0 and QI components are not redundant but rather complementary, leading to an overall Ambiguity Ratio (AR). DCI provides a subjective perception of clarity across stakeholders whilst AR-0 offers an initial objective documentation-based measure of ambiguity. QI supplies contextual insight, explaining patterns and relationships observed in DCI and AR-0. QI also allows for consideration of relevant project deliverables which may not have been developed at all in a particular project, or which may have been developed but are missing important deliverable content and such deficiencies do not show up in DCI or AR-0 assessments.

Together, AR enables both quantitative assessment and qualitative interpretation of how clearly deliverables are defined and communicated within IT projects. Using both perceptual (DCI, QI) and empirical (AR-0) measures aligns with mixed-methods approaches in IS research (Venkatesh et al., 2013; Mertens, 2015). This triangulation ensures construct validity and enables multi-perspective understanding—critical in complex, multi-stakeholder IT environments where clarity is socially constructed and often evolves dynamically (Lyytinen & Newman, 2008; Turner & Cochrane, 1993).

5.2 Stakeholder Characteristics

Stakeholders are individuals or groups with a direct interest in or influence over project deliverables, and by extension, the overall realization of intended project outcomes. Their number and diversity shape the level of uncertainty during project execution (McKeen & Smith, 2003; Lyytinen & Hirschheim, 1987). We distinguish core stakeholders (those defining, approving, or directly impacted by deliverables) from peripheral stakeholders (those indirectly affected but not influential in specification clarity).

The number of stakeholders can be defined as the count of distinct core stakeholders and stakeholder groups directly involved in shaping project deliverables. Risks introduced by the number of stakeholders include coordination overload, increased communication noise, amount of effort required for training and change management. These risks can be measured and potentially mitigated by using stakeholder analysis, communication network density metrics and change impact assessments (Mitchell, Agle & Wood (1997); Barki & Hartwick (2001)).

Diversity of stakeholders is another dimension of SSC (Sambamurthy and Zmud, 2017). This can be seen as the degree of heterogeneity in the core stakeholders' functional role in the business, their desired project outcomes, their desire or time available to participate in project activities, competing priorities, culture, geography, or cognition. Misaligned frames of reference and conflicting decision logics are some of the risks that diverse stakeholders introduce. These risks can be measured and mitigated through stakeholder diversity index assessments (Shannon index), alignment workshops and running shared understanding sessions (Reich & Benbasat (2000); Martinsuo & Hoverfält (2018)).

Sambamurthy and Zmud (2017) also include power as a dimension of stakeholder structure complexity. Power is excluded from this initial operational model due to its conceptual and measurement complexity but remains a critical area for future exploration (Mitchell et al., 1997; Aaltonen & Kujala, 2010).

Operationalization of Stakeholder Structure Complexity (SSC)

Stakeholder Complexity Index (SCI) can be defined as the weighted composite measure of number of stakeholders × diversity of stakeholders. SCI can be informed by both qualitative and quantitative data sources.

Qualitative data sources informing include stakeholder maps, governance records, and meeting transcripts coded for influence, conflict, and alignment patterns. Quantitative measures can be adapted from scales from Reich & Benbasat (2000) and Barki & Hartwick (2001) for evaluating stakeholder alignment and communication quality.

6. PROPOSED ANALYTICAL MODEL AND HYPOTHESES

Building on Sambamurthy and Zmud's (2017) conceptualization of IT project uncertainty, this study proposes a model in which **deliverable ambiguity** and **stakeholder structure complexity** jointly influence **project risk**. Specifically, project risk (PR) is modeled as a function of the *Ambiguity Ratio (AR)*, the *Stakeholder Complexity Index (SCI)*, and their interaction:

$$PR = \beta 0 + \beta 1(AR) + \beta 2(SCI) + \beta 3(AR \times SCI) + \varepsilon$$

Interpretation of the Model

1. **β_0** represents the baseline level of project risk when both ambiguity and stakeholder structure complexity are minimal.
2. **β_1** captures the direct effect of ambiguity in project deliverable specifications on project risk.
3. **β_2** captures the direct effect of stakeholder structure complexity—defined by the number, diversity, and power of stakeholders—on project risk.
4. **β_3** represents the *interaction effect*, indicating whether the risk impact of ambiguity becomes stronger (or weaker) as stakeholder structure complexity increases.
5. **ε** captures all other sources of project risk not explained by ambiguity or stakeholder structure complexity (random noise, unmeasured variables, etc.).

Theoretical Rationale

Ambiguity in project deliverable specifications creates uncertainty in scope, requirements, and expected outcomes, which elevates the likelihood of rework, schedule delays, and misalignment between business and technical teams (Barki, Rivard, & Talbot, 2001; Wallace, Keil, & Rai, 2004). Meanwhile, stakeholder structure complexity—through competing interests, communication breakdowns, and decision bottlenecks—further amplifies these risks (Jiang, Klein, & Balloun, 1998; McLeod et al., 2012).

When both ambiguity and stakeholder structure complexity are high, their joint influence is expected to **compound project uncertainty**, creating an environment where clarity deteriorates faster than mitigation mechanisms can respond.

Hypotheses

H1: *Deliverable ambiguity (AR) is positively associated with project risk (PR).*
Projects with more ambiguous deliverables will experience higher levels of perceived and realized risk.

H2: *Stakeholder complexity (SCI) is positively associated with project risk (PR).*
Projects involving a greater number, diversity, or power imbalance among stakeholders will face higher levels of project risk.

H3: *Stakeholder complexity moderates the relationship between deliverable ambiguity and project risk (AR × SCI).*
The positive effect of ambiguity on project risk will be **stronger** when stakeholder structure complexity is high.

7. Research Design and Methodological Pathways

A **multi-method research strategy** is recommended to test the hypotheses.

Phase 1: Construct Development

Conduct a Delphi study with experienced IT project managers and architects to validate construct dimensions and generate measurable indicators for deliverable specification clarity and stakeholder structure complexity.

Phase 2: Survey Study

Administer a cross-sectional survey across multiple organizations. Use validated scales or newly developed measures to test the hypothesized relationships using regression or SEM.

Phase 3: Case Studies

Complement quantitative findings with qualitative case studies of large digital transformation projects to understand mechanisms in context.

Potential dependent variables include cost and schedule performance, requirement change frequency, and stakeholder satisfaction.

8. Expected Contributions

8.1 Theoretical Contributions

1. **Operationalization of IT Project Uncertainty:** This paper provides measurable constructs for deliverable specification clarity and stakeholder structure complexity, transforming a conceptual model into an empirically testable framework.
2. **Integration of Risk and Uncertainty:** It explicitly positions uncertainty as a *causal mechanism* driving risk, bridging two historically distinct literatures.
3. **Extension of the Sambamurthy–Zmud Model:** The work extends their framework from conceptual to empirical terrain, contributing to the digital transformation and IT governance literatures.

8.2 Practical Contributions

1. **De-Risking through Clarity:** Offers project managers diagnostic tools to assess and reduce ambiguity early in the lifecycle.
2. **Stakeholder Mapping for Risk Mitigation:** Provides a framework for analyzing stakeholder diversity and power to anticipate coordination challenges.

3. **Governance Implications:** Suggests that organizations can mitigate risk by dynamically matching governance intensity to levels of deliverable ambiguity and stakeholder structure complexity.

9. DISCUSSION

This conceptualization reframes IT project uncertainty not as an abstract descriptor but as a *risk-inducing condition* that can be systematically measured and managed. Clarifying deliverables reduces the interpretive latitude available to diverse stakeholders; aligning stakeholders reduces the amplification effect of residual ambiguity.

Projects often fail not simply because requirements are incomplete, but because *uncertainty interacts with stakeholder structure*. A moderately ambiguous requirement can be manageable in a cohesive stakeholder environment but catastrophic in a fragmented one. Thus, managing uncertainty is inherently a *social* as well as a *technical* challenge.

10. FURTHER RESEARCH

The model proposed in this paper extends and operationalizes the conceptual framework introduced by **Sambamurthy and Zmud (2017)** on IT project uncertainty. While the current work focuses on clarifying and measuring the dimensions of **deliverable specification clarity (DSC)** and **stakeholder structure complexity (SSC)**, several avenues for further research remain to advance and empirically validate the model.

First, this study assumes that the **IT project business case**—including its anticipated benefits—is well-defined prior to project mobilization. In practice, however, the business case itself may be uncertain or incomplete, thereby introducing additional layers of ambiguity that propagate throughout the project. Future research could therefore examine how uncertainty in the business case influences downstream deliverables and overall project risk. Second, future work should seek to develop **granular definitions of ambiguity** for each key project deliverable. This may include identifying *explicit questions or criteria* that a deliverable must answer to be considered "clear" rather than "ambiguous." Establishing such diagnostic checklists could strengthen both theoretical precision and practical applicability.

Third, further investigation is needed into the **appropriate level of detail** required for deliverables to achieve clarity. Greater detail does not necessarily equate to greater clarity—particularly for complex deliverables such as business processes and requirements specifications. Researchers could explore how modeling standards (e.g., BPMN levels L1–L4) and the balance between *current state* and *future state* representations affect clarity and alignment among stakeholders. Fourth, the **quality dimension** of deliverables merits closer attention. Determining what constitutes "adequate quality" for clarity—along with the methods and metrics to assess it—remains an open question. This includes exploring how technical accuracy, completeness, and stakeholder validation jointly contribute to perceived and actual clarity.

Fifth, IT project deliverables are **dynamic and temporal** in nature: they evolve as business conditions, technologies, and stakeholder expectations change. Future studies should examine how shifts in context over the project lifecycle alter the clarity–ambiguity balance, and whether adaptive governance mechanisms can mitigate associated risks. Sixth, comparative studies across **different project management methodologies**—such as waterfall versus agile or hybrid models—could reveal how varying degrees of iteration, documentation, and stakeholder engagement influence the relationship between ambiguity, complexity, and project risk.

Finally, while the present study excluded **stakeholder power** from the proposed model due to its conceptual and measurement complexity, Sambamurthy and Zmud (2017) emphasize power as an important dimension of stakeholder influence. Future research should explore how power asymmetries—both formal and informal—shape perceptions of clarity, decision authority, and the ability to manage uncertainty in large-scale IT projects. Mitchell et al. (1997); Keil et al. (2013); Tiwana (2010) provide useful insights that need to be further explored.

11. CONCLUSION

Uncertainty in project deliverables is a central driver of IT project risk. Building on Sambamurthy and Zmud's (2017) conceptual model, this paper defines and operationalizes two foundational dimensions of uncertainty — deliverable specification clarity and stakeholder structure complexity — and theorizes their joint influence on risk outcomes.

By positioning uncertainty as a de-risking target, rather than a descriptive variable, the model provides both a theoretical bridge and a managerial toolkit for improving IT project success. Future empirical work can validate these constructs, test the proposed hypotheses, and refine predictive models of digital transformation risk.

REFERENCES

[1] Aladwani, A. M. (2001). Change management strategies for successful ERP implementation. *Business Process Management Journal, 7*(3), 266–275.

[2] Aaltonen, K., & Kujala, J. (2010). A project lifecycle perspective on stakeholder influence strategies in global projects. *Scandinavian Journal of Management, 26*(4), 381–397.

[3] Armenakis, A. A., Harris, S. G., & Mossholder, K. W. (1999). Creating readiness for organizational change. *Human Relations, 46*(6), 681–703.

[4] Barki, H., Rivard, S., & Talbot, J. (2001). An integrative contingency model of software project risk management. *Journal of Management Information Systems, 17*(4), 37–69.

[5] Bass, L., Clements, P., & Kazman, R. (2022). *Software Architecture in Practice* (5th ed.). Addison-Wesley.

[6] Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed.* Addison-Wesley.

[7] Browning, T. R. (2014). Managing complex project process models with a process architecture framework. *Systems Engineering, 17*(3), 241–261.

[8] Chapman, C., & Ward, S. (2003). *Project risk management: Processes, techniques, and insights* (2nd ed.). Wiley.

[9] Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering.* Springer.

[10] Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of Business Process Management* (2nd ed.). Springer.

[11] Freeman, R. E. (1984). *Strategic management: A stakeholder approach.* Pitman.

[12] Galbraith, J. R. (1973). *Designing complex organizations.* Addison-Wesley.

[13] Galster, M., Avgeriou, P., Weyns, D., & Becker, S. (2013). Variability in software architecture: Current practice and challenges. *Computer, 46*(7), 90–94.

[14] Gregor, S., Hart, D., & Martin, N. (2006). Enterprise architectures: Enablers of business strategy and IS/IT alignment in public sector organizations. *Information Technology & People, 20*(2), 96–120.

[15] ISO/IEC. (2011). *Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—System and software quality models* (ISO/IEC 25010:2011). ISO.

[16] Jiang, J. J., Klein, G., & Balloun, J. L. (1998). Systems analysts' perceptions of job skills. *Information & Management, 34*(1), 45–53.

[17] Khatri, V., & Brown, C. V. (2010). Designing data governance. *Communications of the ACM, 53*(1), 148–152.

[18] Keil, M., Rai, A., & Liu, S. (2013). How user risk and requirements risk moderate the effects of formal and informal control on the process performance of IT projects. *European Journal of Information Systems, 22*(6), 650–672.

[19] Kotter, J. P. (1996). *Leading change.* Harvard Business School Press.

[20] Lyytinen, K., & Hirschheim, R. (1987). Information systems failures: A survey and classification of the empirical literature. *Oxford Surveys in Information Technology, 4*(1), 257–309.

[21] Lyytinen, K., & Newman, M. (2008). Explaining information systems change: A punctuated socio-technical change model. *European Journal of Information Systems, 17*(6), 589–613.

[22] Lyytinen, K., Mathiassen, L., & Ropponen, J. (1998). Attention shaping and software risk—A categorical analysis of four classical risk management approaches. *Information Systems Research, 9*(3), 233–255.

[23] Mairiza, D., Zowghi, D., & Nurmuliani, N. (2010). An investigation into the notion of non-functional requirements. *Proceedings of the 2010 ACM Symposium on Applied Computing*, 311–317.

[24] Martinsuo, M., & Hoverfält, P. (2018). Change program management: Toward a capability for managing value-oriented, integrated multi-project change in its context. *International Journal of Project Management, 36*(1), 134–149.

[25] McKeen, J. D., & Smith, H. A. (2003). *Making IT happen: Critical issues in IT management.* Wiley.

[26] McLeod, L., Doolin, B., & MacDonell, S. G. (2012). A perspective-based understanding of project success. *Project Management Journal, 43*(5), 68–86.

[27] Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a theory of stakeholder identification and salience. *Academy of Management Review, 22*(4), 853–886.

[28] Nelson, R. R. (2007). IT project management: Infamous failures, classic mistakes, and best practices. *MIS Quarterly Executive, 6*(2), 67–78.

[29] Otto, B. (2011). A morphology of the organisation of data governance. *ECIS 2011 Proceedings*.

[30] PMI. (2021). *A guide to the project management body of knowledge (PMBOK® Guide)* (7th ed.). Project Management Institute.

[31] Prosci. (2018). *Best practices in change management* (11th ed.). Prosci Research.

[32] Reich, B. H., & Benbasat, I. (2000). Factors that influence the social dimension of alignment between business and IT objectives. *MIS Quarterly, 24*(1), 81–113.

[33] Sambamurthy, V., & Zmud, R. W. (2017). *Guiding the digital transformation of organizations.* Cognella Academic.

[34] Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems, 17*(4), 5–36.

[35] Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997). Data quality in context. *Communications of the ACM, 40*(5), 103–110.

[36] Tiwana, A. (2010). *Software platforms: Ecosystems of innovation.* Morgan Kaufmann.

[37] Tiwana, A., & Keil, M. (2007). Does peripheral knowledge complement control? *Strategic Management Journal, 28*(6), 623–634.

[38] Turner, J. R., & Cochrane, R. A. (1993). Goals-and-methods matrix: Coping with projects with ill-defined goals and/or methods. *International Journal of Project Management, 11*(2), 93–102.

[39] van der Aalst, W. M. P. (2016). *Process mining: Data science in action* (2nd ed.). Springer.

[40] Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative–quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly, 37*(1), 21–54.

[41] Vidal, L. A., & Marle, F. (2008). Understanding project complexity: Implications on project management. *Kybernetes, 37*(8), 1094–1110.

[42] Wallace, L., Keil, M., & Rai, A. (2004). Understanding software project risk: A cluster analysis. *Information & Management, 42*(1), 115–125.

Author

Irshad Abdulla is an enterprise technology leader with over two decades of experience shaping digital transformation strategies and large-scale architecture practices across energy, utilities, consulting, and public sector organizations. He has held senior roles including Chief Technology Architect at a public utility and Director of Digital Transformation at a public health authority. He has also worked in management consulting where he advised clients on maximizing value from enterprise platforms, cloud adoption, and business-technology alignment.

Irshad's research interests lie in better understanding how technology can be applied as a driver of strategic innovation as well as a practical enabler of organizational outcomes. He is a frequent contributor to thought leadership on digital strategy, governance, and enterprise transformation. Irshad holds a PhD in Management Information Systems and Public Administration and teaches at Simon Fraser University, Beedie School of Business.

APPENDIX A: AMBIGUITY IN KEY TECHNOLOGY PROJECT DELIVERABLES

Ambiguity in any of the deliverables below amplifies *project uncertainty* by weakening shared understanding (Reich & Benbasat, 2000) and reducing ability to predict outcomes (Tiwana, 2010). Thus, improved deliverable specification clarity functions as a risk mitigation mechanism—a pathway to lower project execution risk (Sambamurthy & Zmud, 2017).

Table 1. Key Project Deliverables, Sources of Ambiguity, and Clarity Mechanisms

Deliverable	Description / Scope	What Needs to Be Clear	What May Be Ambiguous	Mechanisms to Reduce Ambiguity / Increase Clarity	Representative References
1. Project Charter & Plan	Foundational document defining project purpose, scope, delivery model, roles, responsibilities, resource plans, budget, and project management tools	Scope boundaries, role clarity (incl. vendors/ system integrators), governance model decision making processes, project record management practices, milestones	Diffuse scope statements, unclear accountabilities, incomplete resourcing assumptions. Business case and anticipated benefits are unclear, making it difficult to firm up the project charter deliverable.	RACI matrices, detailed work breakdown structures (WBS), integrated project plans, formal governance charters	PMI (2021); Barki et al. (2001); Nelson (2007)
2. Business Requirements	Description of business problems, objectives, and needs that must be satisfied.	Business outcomes, value drivers, success metrics framed within the context of business processes.	Overly generic goals, scope creep, conflicting stakeholder interests, statements about business requirements with business process / workflow context. Confusing business requirements with system functionality.	Joint requirements sessions, business case validation, business process modelling, prioritization matrices	Tiwana & Keil (2007); Schmidt et al. (2001)
3. Functional Design Specifications	System set-up tasks and design, system process flows, and user interactions	User journeys, process variants, exception handling, system	Assumptions about inherent system capabilities, lack of description of	Prototypes, user stories, iterative walkthroughs	Browning (2014); Boehm & Turner (2004)

Deliverable	Description / Scope	What Needs to Be Clear	What May Be Ambiguous	Mechanisms to Reduce Ambiguity / Increase Clarity	Representative References
	with the IT system	configurations and customizations	how the business requirements will be met by the system.		
4. Non-Functional Requirements (NFRs) & Designs	System performance, reliability, usability, supportability, portability, accessibility, device compatibility	Performance SLAs, response times, usability standards, device coverage	Implicit expectations, lack of quantitative benchmarks, inadequate description of how the NFRs will be met	ISO/IEC 25010 frameworks, usability testing, operational readiness reviews	Chung et al. (2000); Mairiza et al. (2010); ISO/IEC 25010:2011
5. Solution Architecture	Conceptual and logical structure of system components and relationships	End-to-end description of the solution including technology stack, interfacing system, user access methods, backup and recovery procedures, security model.	Undefined system boundaries, conflicting architectural patterns, unclear decision making processes	Architecture review boards, reference models, architectural documentation standards	Gregor et al. (2006); Zachman (1987)
6. Data Design	Data models, data ownership, master data definitions, metadata, and data lifecycle management	Data definitions, ownership, lineage, retention, and quality rules. Decisions about data to be migrated to the new system, decisions about how un-migrated data will be handled.	Data overlap, inconsistent definitions, unowned data domains and unclear decision-making rights related to data, inadequate description of how data will be validated	Data governance boards, data dictionaries, master data management policies	Khatri & Brown (2010); Otto (2011); Strong et al. (1997)
7. Integration Design	Interfaces and data exchanges between systems, APIs, and message flows	Integration protocols, data mapping, synchronization frequency	Assumed API behavior, missing error-handling design, undefined	Interface control documents (ICDs), integration testing	Barki et al. (2001); Wallace et al. (2004)

Deliverable	Description / Scope	What Needs to Be Clear	What May Be Ambiguous	Mechanisms to Reduce Ambiguity / Increase Clarity	Representative References
			volumes of data to be interfaced	frameworks, API versioning	
8. Testing and Acceptance Criteria	Test cases, defect classification, acceptance conditions	Acceptance thresholds, UAT ownership, regression scope	Disputed quality standards, differing “fit for purpose” views	Acceptance criteria templates, third-party QA, validation sign-offs	Wallace & Keil (2004); Lyytinen et al. (1998)
9. Training Deliverables	Materials, exercises, and simulations to enable end-user readiness	Training scope, role coverage, delivery method	Gaps in training coverage, unclear performance expectations	Role-based training, learning analytics, feedback mechanisms	Aladwani (2001); Prosci (2018)
10. Organizational Change Management Deliverables	Change impact assessments, stakeholder engagement, communication plans, adoption measures	Change impacts, adoption readiness, engagement responsibilities	Misjudged resistance, misaligned communication cadence	Stakeholder engagement plans, readiness assessments, change heatmaps	Kotter (1996); Armenakis et al. (1999)
11. Project Dependencies	Identification and sequencing of related or prerequisite initiatives	Dependency list, critical path alignment, ownership	Overlooked interdependencies, timing mismatches	Dependency matrices, integrated master schedules, risk-adjusted plans	Vidal & Marle (2008); PMBOK (2021)
12. Transition and Deployment Plans	Migration to production, go-live, support handover	Go-live criteria, rollback strategies, operational support model	Undefined readiness indicators, ownership gaps	Simulation runs, stage-gate reviews, post-implementation reviews	Nelson (2007); PMI (2021)
13. The IT System	Built solution meeting the users needs, which will be at varying stages of completion in different project stages	Where the system will be hosted (cloud, on-premises, hybrid), how updates to the build will be applied, how functional and non-functional	Multiple components may make up the whole system and each component may not be clearly defined	Hosting model, technical specifications, as-built documents	Galster et al. (2013); Bass et al. (2022)

Deliverable	Description / Scope	What Needs to Be Clear	What May Be Ambiguous	Mechanisms to Reduce Ambiguity / Increase Clarity	Representative References
		requirements are achieved			