

Managing the Quality of Project Deliverables: A Process Framework Based on ISO 21500 and ISO 21502

Dejan IGOV, Duško ČUČKOVIĆ

Faculty of Project and Innovation Management prof. dr Petar Jovanović, Belgrade, Republic of Serbia

Faculty of Project and Innovation Management prof. dr Petar Jovanović, Belgrade, Republic of Serbia

dejan.igov@pmc.edu.rs, dusko.cuckovic@pmc.edu.rs

Corresponding Author: Dejan IGOV

Abstract— Project deliverable quality is a key determinant of project success and user satisfaction. This paper develops a process framework for managing deliverable quality within ISO 21500 and ISO 21502. Based on qualitative analysis of standards and relevant literature, the framework connects user requirements, quality criteria, scope and deliverables, control points, corrective actions, formal acceptance, and benefits evaluation. It is conceived as an adaptable cycle for predictive, Agile, Scrum, and hybrid environments. The framework is complemented by Quality 4.0 mechanisms, ISO/IEC 25000 SQuaRE metrics for ICT deliverables, and Success Management activities. It provides an integrative basis for transparent, measurable, and adaptive quality management and identifies priorities for empirical validation.

Keywords— project management, quality management, project deliverables, ISO 21502, ISO/IEC 25000, Quality 4.0

I. INTRODUCTION

Project management increasingly serves as a mechanism for value creation, organizational change, and strategic implementation rather than only as a temporary arrangement for completing predefined activities. In project-oriented and engineering-oriented organizations, this broader role makes the quality of project deliverables especially important because deliverables are the visible and usable outputs through which users, clients, and the organization judge project performance. Quality-related managerial practices are also linked with business performance and competitiveness [1]. Consequently, deliverable quality should be managed as a strategic and managerial concern throughout the project life cycle, not only as a technical inspection at project closure.

Traditional assessments of project success have emphasized time, cost, and scope. Although this so-called iron triangle remains useful for operational control, it is insufficient for judging whether a project has created useful and acceptable results [2], [3]. The distinction between project management success and broader project success therefore remains essential: a project may be efficiently executed and still fail to satisfy users or realize expected benefits [4], [5]. Deliverable quality provides an important link between these perspectives because it connects what the project produces with user

requirements, acceptance criteria, usability, and organizational value.

A standardized approach is useful because ISO 21500:2021 establishes the context and concepts of project, programme, and portfolio management, while ISO 21502:2020 provides guidance for managing individual projects [6], [7]. ISO 9001:2015 adds customer focus, the process approach, risk-based thinking, performance evaluation, and continual improvement [8]. Together, these standards support the view that quality should be planned, monitored, verified, and improved from requirements definition to formal acceptance and subsequent benefits evaluation. The research question is therefore: How can ISO 21500 and ISO 21502 contribute to more systematic management of project deliverable quality in project-oriented organizations?

The proposed approach should not be interpreted as a rigid, plan-driven sequence. In Agile and Scrum environments, quality is embedded in repeated refinement, implementation, review, retrospective, and the Definition of Done [9]. In digitally mature organizations, Quality 4.0 mechanisms such as automated testing, process mining, predictive analytics, and AI-enabled anomaly detection can strengthen control and support earlier corrective action [10]. The framework is therefore treated as a repeatable management logic that can be adapted to predictive, iterative, and hybrid project environments.

II. THEORETICAL AND STANDARDIZED BACKGROUND

Project quality can be examined at two related but analytically distinct levels. The first is the quality of the project management process: how the project is initiated, planned, organized, monitored, controlled, and closed. The second is the quality of project deliverables: the degree to which products, services, results, or changes meet explicit requirements, relevant standards, acceptance criteria, and stakeholder expectations. A disciplined process increases the probability of acceptable outputs but does not guarantee them; conversely, an acceptable output produced through an unstable process may be difficult to reproduce. Effective quality management therefore requires both process discipline and explicit attention to deliverable characteristics.

The broader literature on project success reinforces this position. Time, cost, and scope describe important execution constraints, but project success also depends on usefulness, stakeholder satisfaction, strategic contribution, and benefits realization [2]–[5]. Jugdev and Müller showed that understandings of project success evolved toward multidimensional and longer-term criteria [11]. Deliverable quality is therefore neither identical to project management efficiency nor reducible to technical conformity. It is the point at which management processes, stakeholder expectations, and project value intersect. Digital quality approaches further extend this logic by enabling more continuous monitoring and earlier identification of deviations [12].

ISO 21500 places projects within a broader organizational system and links project outputs with organizational objectives and expected benefits [6]. ISO 21502 provides more operational guidance for project organization, scope, deliverables, stakeholders, risks, changes, and controls [7]. ISO 9001 complements this structure through customer focus, process management, evidence-based evaluation, and continual improvement [8], while ISO 10006 applies quality-management principles specifically to projects [13]. Their combined use does not create a universal methodology; rather, it provides a coherent management logic that can be tailored to project size, complexity, lifecycle, regulatory context, and stakeholder expectations.

For software and ICT deliverables, broad quality expectations should be translated into domain-specific characteristics. The ISO/IEC 25000 SQuaRE series provides a structured basis for quality requirements and evaluation [14], while ISO/IEC 25010:2023 defines a product quality model that supports specification and assessment of characteristics such as functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability [15]. These models are useful when general stakeholder expectations must be converted into measurable or verifiable acceptance criteria.

III. METHODOLOGY

The paper is a conceptual, review-based study grounded in qualitative document analysis. Document analysis is appropriate when the objective is to identify meanings, relationships, and conceptual connections across standards, professional guidance, and scientific literature [16], [17]. Sources were selected for their relevance to project success, deliverable quality, acceptance criteria, standardized project management, quality management, Agile delivery, and digital quality mechanisms. The PMBOK Guide was also considered as a professional reference for contemporary project-management principles and practices [18].

The analysis followed four steps. First, the literature was examined to distinguish project management success from the quality and value of project results. Second, ISO 21500, ISO 21502, ISO 9001, ISO 10006, and the ISO/IEC 25000 series were analyzed to identify principles relevant to requirements, scope, deliverables, controls, acceptance, and improvement. Third, these

insights were synthesized with Agile, continuous-delivery, Quality 4.0, and Success Management perspectives. Fourth, the resulting concepts were organized into a seven-element process framework. The framework is conceptual rather than empirically validated, which is consistent with the role of review-based studies in synthesizing knowledge and proposing directions for research and practice [19].

IV. PROCESS FRAMEWORK FOR MANAGING PROJECT DELIVERABLE QUALITY

The proposed framework treats deliverable quality as a continuous management cycle comprising seven connected elements: user requirements, quality criteria, project scope and deliverables, control points, deviations and corrective actions, deliverable acceptance, and benefits evaluation. The sequence is a logical rather than strictly chronological structure. In predictive projects, elements may correspond to phases and milestones; in Agile and hybrid projects, the same logic can be repeated for each sprint, increment, release, or major deliverable.

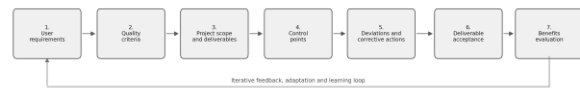


Fig. 1 Process framework for managing project deliverable quality

Source: Authors, based on [6], [7], [8], [9], [13], [15].

The first element is the definition of user requirements. Quality cannot be assessed reliably unless the project team understands what users need, what problem the project should solve, and which benefits are expected. Requirements may be functional, technical, organizational, regulatory, safety-related, financial, or user-oriented. They should be defined with relevant stakeholders and should include the context in which the deliverable will be used. Ambiguous requirements create downstream uncertainty and make later acceptance subjective.

The second element translates requirements into quality criteria. Statements such as reliable, secure, usable, or high performance must be operationalized through measurable or otherwise verifiable indicators. Acceptance criteria have both a technical and a communication function because they define what will be checked, how it will be checked, and on what basis the deliverable will be accepted. For ICT deliverables, SQuaRE characteristics can support this translation [14], [15].

The third element connects quality criteria with project scope and specific deliverables. Scope defines what is to be delivered, but it does not by itself guarantee acceptable quality. Each key deliverable should therefore be traceable to relevant requirements and acceptance criteria. This linkage reduces the risk that a project formally fulfils scope while producing an output that does not satisfy users and reflects the integrated management of scope, deliverables, stakeholders, risks, and changes emphasized by ISO 21502 [7].

The fourth element establishes control points: predefined opportunities to verify whether work and deliverables are progressing in accordance with requirements and criteria. Depending on the lifecycle, these may include stage gates, technical reviews, work-package completion, sprint reviews, automated tests, user testing, audits, or client approvals. Their main purpose is preventive: deviations should be detected before they become expensive defects, schedule delays, or acceptance disputes.

The fifth element manages deviations and corrective actions. Deviations may arise between planned and actual performance, specified requirements and produced deliverables, or initial expectations and changing project circumstances. They should be documented, assessed for impact, prioritized, and connected with risk, scope, time, cost, and change management. Corrective action may involve technical repair, process adjustment, requirement clarification, scope change, or renewed stakeholder alignment. Quality 4.0 tools can strengthen this element through data-supported monitoring, defect analysis, predictive alerts, and anomaly detection [10], [12].

The sixth element is formal deliverable acceptance. Acceptance should confirm that predefined criteria have been met rather than merely record that an activity is complete. Responsibilities for verification and approval should be clear, and the basis for acceptance or return for correction should be documented. This creates traceability and reduces the likelihood of later disagreement between the project team, client, and users.

The seventh element is benefits evaluation. Formal acceptance does not necessarily mean that the project has generated the expected value. After initial use, organizations should evaluate whether the deliverable has contributed to user satisfaction, problem solving, process improvement, strategic objectives, or other expected benefits. This phase can be linked with formal Success Management activities, including identification of success criteria and factors, planning of evaluation, monitoring of success, review of results, and reporting at closure [20]. Benefits evaluation also creates feedback for future requirements, criteria, and project practices.

A. Adaptation of the Framework to Contemporary Project Environments

The framework is intentionally adaptable. In Agile and Scrum environments, the seven elements can be repeated at sprint or increment level, with the Definition of Done functioning as an agreed quality threshold and sprint reviews providing frequent stakeholder validation [9]. Continuous integration and delivery pipelines can act as recurring control points by automatically verifying builds, tests, and deployment conditions [21]. In plan-driven projects, the same logic can be aligned with formal stage gates and milestone reviews.

Quality 4.0 extends the framework from episodic review toward continuous, data-supported assurance. Automated testing, process mining, sensor data, defect logs, real-time dashboards, predictive analytics, and AI-enabled anomaly detection can improve the timeliness of decisions without replacing managerial judgment [10], [12]. The extent of digitalization should depend on data

availability, infrastructure, analytical competence, governance, and project risk. Organizations with lower digital maturity can apply the same framework through structured checklists, review meetings, shared deviation registers, and standardized acceptance forms.

Domain-specific tailoring is equally important. Construction projects may define control points around design approvals, structural milestones, inspections, safety requirements, and regulatory compliance. Software projects may emphasize reliability, usability, security, automated testing, and release readiness. The framework should therefore be treated as a common management logic, not as a substitute for technical standards, lifecycle-specific methods, or professional judgment.

B. Implementation Logic and Managerial Use

For practical application, the framework should be embedded in the organization's existing project governance rather than introduced as a parallel quality system. At project initiation, expected benefits and key user requirements should be recorded together with the initial scope. During planning, these requirements should be translated into acceptance criteria, verification methods, responsible roles, and planned control points. This creates a traceability chain from the reason for initiating the project to the evidence used to accept its outputs. The chain is especially important where several organizational units, suppliers, or external clients participate in delivery, because it reduces ambiguity over who defines, verifies, and approves quality.

During execution, control points should be selected according to risk and the cost of late correction. High-risk, safety-critical, technically novel, or customer-visible deliverables require earlier and more frequent verification than routine outputs. Control evidence can combine quantitative indicators, test results, inspection records, review decisions, and stakeholder feedback. When a deviation is identified, the project team should distinguish between a defect in execution and a change in the underlying requirement. The first usually calls for corrective action; the second may require formal change control and renewed agreement on scope, schedule, cost, risk, and acceptance criteria. This distinction prevents teams from treating changing expectations as simple defects or, conversely, using change requests to conceal avoidable quality failures.

The framework also supports organizational learning when acceptance and benefits evaluation are connected. Acceptance answers whether the deliverable satisfies the agreed criteria at a defined point in time, whereas benefits evaluation asks whether the accepted output creates the intended value in use. The two decisions should therefore be related but not conflated. Evidence from benefits evaluation can reveal that formally correct requirements did not fully represent user needs, that certain quality characteristics were under-specified, or that operating conditions changed after delivery. Such findings should feed lessons learned, future requirement templates, risk registers, and quality-control practices. In this way, the final element of one project becomes an input to the planning of subsequent projects.

A scalable implementation is preferable to an administratively heavy one. Small projects may use a concise requirements-and-acceptance checklist, a small number of review points, and a simple deviation log. Large or regulated projects may require formal traceability matrices, independent verification, configuration control, audit evidence, and post-implementation benefit reviews. Digital tools can increase visibility and speed, but the management logic remains the same: define what quality means for the specific deliverable, decide how evidence will be obtained, act on deviations while correction is still feasible, and confirm both acceptance and value. This principle preserves the adaptability emphasized by ISO-based project management while maintaining sufficient control for accountable decision-making.

C. Quality Evidence, Traceability, and Decision Rules

The framework becomes operational only when quality criteria are connected with evidence and explicit decision rules. For each major deliverable, an organization can establish a simple traceability structure linking the originating requirement, the relevant quality criterion, the verification method, the person or role responsible for verification, the evidence produced, and the acceptance decision. This structure need not be administratively complex. Its purpose is to ensure that acceptance is based on information defined before the decision is made. Traceability is particularly valuable when requirements change, because the project team can identify which deliverables, tests, risks, and approval decisions are affected and can update them consistently rather than relying on informal communication.

Verification evidence should be proportionate to the significance of the deliverable. In some projects, an inspection record or user review may be sufficient; in others, evidence may include test reports, regulatory certificates, performance measurements, security assessments, configuration records, or independent verification. The framework therefore supports a risk-based approach to quality control. The greater the consequence of failure, uncertainty, technical novelty, or difficulty of late correction, the stronger the evidence and the earlier the control point should be. Conversely, low-risk routine deliverables should not be burdened with controls whose cost exceeds their decision value. This proportionality helps maintain the balance between standardization and adaptability that is central to the proposed approach.

Clear decision rules are also important when criteria are not fully met. The result of a control point should not automatically be reduced to a binary pass-or-fail judgment. Depending on the deviation, the project may accept the deliverable, accept it conditionally with a documented action, return it for correction, or initiate formal change control when the underlying requirement is no longer appropriate. The decision should consider the effect on scope, time, cost, risk, usability, and expected benefits. By separating verification evidence from the managerial decision that follows, the framework preserves technical objectivity while allowing project governance to address trade-offs transparently. This is

especially relevant in complex projects where a technically minor nonconformity may have major contractual or safety implications, while a larger deviation may be acceptable if stakeholder needs have legitimately changed.

D. Benefits Feedback and Organizational Learning

Benefits evaluation extends the quality perspective beyond the formal handover of a deliverable. Acceptance demonstrates that the agreed criteria were satisfied at a particular point, but it does not necessarily show whether the output solved the intended problem or created sustainable value. For this reason, post-acceptance evaluation should use a limited set of benefit indicators that can be observed after implementation. Depending on the project, these may include user adoption, reduction in processing time, defect frequency, service reliability, safety performance, revenue contribution, cost reduction, or stakeholder satisfaction. The indicators should be related to the objectives defined at project initiation so that the organization can distinguish between a deliverable that was correctly produced and one that also achieved its intended purpose.

The feedback loop is important because quality problems are not always caused by poor execution. A deliverable may conform perfectly to its specification and still create limited value if the original requirement was incomplete, if assumptions changed, or if the operating environment evolved. Benefits evaluation can therefore reveal weaknesses in requirement definition, acceptance criteria, stakeholder involvement, or benefit assumptions. These findings should be captured as organizational knowledge and used to improve future requirement templates, risk assessments, quality plans, and control-point design. Linking this feedback with Success Management reinforces the long-term perspective on project success [20] and turns project closure from an administrative endpoint into a source of evidence for continual improvement.

This learning mechanism is also compatible with iterative delivery. In Agile and continuous-delivery settings, benefit signals can be collected earlier and more frequently, allowing product priorities and acceptance criteria to evolve while delivery is still underway. In predictive projects, the same logic can be applied through post-implementation reviews and planned benefit-realization checkpoints. The frequency differs, but the principle is unchanged: evidence from use should inform future decisions. This closes the cycle proposed in the framework and explains why benefits evaluation is positioned not merely as a final measurement activity but as the connection between completed deliverables and improved management of subsequent projects.

V. DISCUSSION AND LIMITATIONS

The framework integrates two issues that are often treated separately: the quality of project management and the quality of project outputs. Process discipline supports predictable execution, but project success ultimately depends on whether deliverables are accepted, usable, and capable of generating value. Conversely, a technically sound deliverable produced through an

unstable process may be difficult to repeat and may expose the organization to unnecessary risk. The model therefore connects requirements, scope, controls, acceptance, and benefits instead of treating quality as an isolated verification activity.

A second contribution is the integration of standardization with iterative and digital delivery practices. ISO 21500 and ISO 21502 provide a stable conceptual and managerial structure [6], [7], while ISO 9001 and ISO 10006 strengthen customer focus, risk-based thinking, process control, and improvement [8], [13]. Agile practices, continuous delivery, SQuaRE models, and Quality 4.0 mechanisms provide different ways to operationalize that structure [9], [10], [12], [14], [15], [21]. This combination can help organizations avoid two extremes: rigid standardization that becomes an administrative burden and flexible delivery without sufficient traceability or acceptance discipline.

The framework has several limitations. First, it is derived from literature and standards and has not yet been empirically validated. Second, the seven elements are presented as a sequence for clarity, but direct linear implementation would be inappropriate in many high-velocity or exploratory projects. Third, domain-specific quality models and regulatory requirements remain necessary for technically complex deliverables. Fourth, Quality 4.0 mechanisms depend on data quality, digital infrastructure, analytical competence, and governance. Finally, benefits evaluation extends beyond deliverable acceptance and may require dedicated roles and Success Management routines. These limitations define a research agenda rather than invalidate the framework.

Future studies should test the framework through case studies, surveys, longitudinal designs, and comparative analyses across sectors. Relevant outcomes include user satisfaction, frequency of corrective requests, defect detection time, cost of rework, acceptance-cycle duration, and benefits realization. Comparative research could also examine how project size, lifecycle, technological complexity, and organizational maturity influence the relative importance of individual framework elements.

VI. CONCLUSIONS

Managing the quality of project deliverables requires a broader perspective than the traditional emphasis on time, cost, and scope. Project management efficiency remains important, but overall project success also depends on whether outputs meet user requirements, are formally acceptable, remain usable in practice, and contribute to expected organizational benefits. Deliverable quality should therefore be treated as a continuous management responsibility rather than as a final inspection event.

The paper proposes a seven-element process framework that connects user requirements, quality criteria, scope and deliverables, control points, deviations and corrective actions, deliverable acceptance, and benefits evaluation. ISO 21500 and ISO 21502 provide the principal project-management structure, while ISO 9001 and ISO 10006 reinforce process and quality-management principles. The framework can be

complemented by ISO/IEC 25000 SQuaRE models for technical quality criteria, Agile and Scrum practices for iterative validation, continuous delivery for recurring automated controls, Quality 4.0 analytics for early detection of deviations, and Success Management for explicit evaluation of benefits.

The framework is intended as an adaptable management logic for project-oriented and engineering-oriented organizations. Its practical value lies in making the relationship between requirements, quality evidence, acceptance, and benefits more transparent. Empirical validation is the next necessary step, particularly to determine which combinations of controls and metrics are most effective across different project environments.

REFERENCES

- [1] M. Bakator, D. Đorđević, and D. Čočkalović, "Developing a model for improving business and competitiveness of domestic enterprises", *Journal of Engineering Management and Competitiveness (JEMC)*, vol. 9, no. 2, pp. 87-96, 2019, doi: 10.5937/JEMC1902087B.
- [2] R. Atkinson, "Project management: Cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria", *International Journal of Project Management*, vol. 17, no. 6, pp. 337-342, 1999, doi: 10.1016/S0263-7863(98)00069-6.
- [3] D. Baccarini, "The logical framework method for defining project success", *Project Management Journal*, vol. 30, no. 4, pp. 25-32, 1999, doi: 10.1177/875697289903000405.
- [4] A. de Wit, "Measurement of project success", *International Journal of Project Management*, vol. 6, no. 3, pp. 164-170, 1988, doi: 10.1016/0263-7863(88)90043-9.
- [5] A. J. Shenhar and D. Dvir, *Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation*. Boston, MA: Harvard Business School Press, 2007.
- [6] International Organization for Standardization, *ISO 21500:2021, Project, programme and portfolio management - Context and concepts*. Geneva, Switzerland: ISO, 2021.
- [7] International Organization for Standardization, *ISO 21502:2020, Project, programme and portfolio management - Guidance on project management*. Geneva, Switzerland: ISO, 2020.
- [8] International Organization for Standardization, *ISO 9001:2015, Quality management systems - Requirements*. Geneva, Switzerland: ISO, 2015.
- [9] K. Schwaber and J. Sutherland, *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org, 2020. <https://scrumguides.org/scrum-guide.html>, Accessed: 08.09.2026.
- [10] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, "Data analytics in Quality 4.0: Literature review and future research directions", *International Journal of Computer Integrated Manufacturing*, vol. 36, no. 5, pp. 678-701, 2023, doi: 10.1080/0951192X.2022.2128219.
- [11] K. Jugdev and R. Müller, "A retrospective look at our evolving understanding of project success", *Project Management Journal*, vol. 36, no. 4, pp. 19-31, 2005, doi: 10.1177/875697280503600403.
- [12] A. M. Dias, A. M. Carvalho, and P. Sampaio, "Quality 4.0: Literature review analysis, definition and impacts of the digital transformation process on quality", *International Journal of Quality & Reliability Management*, vol. 39, no. 6, pp. 1312-1335, 2022, doi: 10.1108/IJQRM-07-2021-0247.
- [13] International Organization for Standardization, *ISO 10006:2017, Quality management - Guidelines for quality management in projects*. Geneva, Switzerland: ISO, 2017.
- [14] International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC 25000:2014, Systems and software engineering - Systems and software Quality*

Requirements and Evaluation (SQuaRE) - Guide to SQuaRE. Geneva, Switzerland: ISO, 2014.

- [15] International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC 25010:2023, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Product quality model*. Geneva, Switzerland: ISO, 2023.
- [16] G. A. Bowen, "Document analysis as a qualitative research method", *Qualitative Research Journal*, vol. 9, no. 2, pp. 27-40, 2009, doi: 10.3316/QRJ0902027.
- [17] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed. Thousand Oaks, CA: SAGE Publications, 2019.
- [18] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th ed., and The Standard for Project Management*. Newtown Square, PA: Project Management Institute, 2021.
- [19] H. Snyder, "Literature review as a research methodology: An overview and guidelines", *Journal of Business Research*, vol. 104, pp. 333-339, 2019, doi: 10.1016/j.jbusres.2019.07.039.
- [20] N. Takagi and J. Varajão, "ISO 21502 and Success Management: A required marriage in project management", *SAGE Open*, vol. 15, no. 3, 2025, doi: 10.1177/21582440251355046.
- [21] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Boston, MA: Addison-Wesley, 2010.