PE

ENGINEERING QUALITY ASSURANCE, DESIGN REVIEW & CHANGE CONTROL

4 CEH

PERTINENT

BuilderCE

PROFESSIONAL ENGINEERING CONTINUING EDUCATION

Engineering Quality Assurance, Design Review & Change Control

Requirements, Independent Review & Defensible Records

Self-Paced Course Study Guide



Illustrative engineering coordination scene. This contextual illustration does not document a specific project or incident.

COURSE	CREDIT	INSTRUCTOR
ENG-004	4 CEH / PDH · Pertinent to Engineering	James Barber, P.E.

General engineering practice.

Superior Learning, LLC d/b/a BuilderCE

Course overview

Course description

Engineering quality assurance is the disciplined set of processes used to improve the reliability, traceability, coordination, and defensibility of engineering work. It is not limited to a corporate quality manual, an ISO certification, or a final drawing check. In day-to-day engineering practice, quality is created when requirements are understood, assumptions are controlled, interfaces are identified, calculations and models are checked, design reviews are technically meaningful, revisions are authorized, and the final record allows another qualified professional to reconstruct what was decided and why.

This course examines the practical controls that connect engineering judgment with a reliable design process. It addresses quality assurance and quality control; graded application of review rigor; requirements and design-basis control; interdisciplinary coordination; design review and independent verification; comment resolution; document and configuration control; engineering change management; nonconformance and corrective action; records; and applied scenarios involving mismatched revisions, field changes, incomplete criteria, unresolved comments, and downstream interface failures.

The course is intentionally multidisciplinary. A structural engineer, transportation engineer, mechanical engineer, electrical engineer, controls engineer, environmental engineer, industrial engineer, and engineering manager may use different codes and technical methods, but they face the same control questions: What requirement governs? What assumptions were made? Who checked the work? What changed? Who had authority to approve the change? Did all affected documents remain synchronized? Can the final record show that the released design was reviewed against the correct criteria?

Learning objectives

- Explain the role of quality assurance and quality control in professional engineering work and distinguish process controls from final-product inspection.
- Identify methods for establishing and maintaining requirements, design criteria, assumptions, interfaces, and design-basis traceability.
- Describe effective design-review practices, including appropriate review depth, interdisciplinary coordination, independent checking, verification, comment resolution, and approval.
- Apply document, configuration, and revision-control principles to engineering drawings, calculations, specifications, models, reports, and other controlled technical records.
- Evaluate proposed engineering changes for technical impact, interfaces, downstream consequences, approvals, implementation status, and required record updates.
- Recognize how nonconformances, corrective actions, lessons learned, and quality records support continuous improvement and reduce recurrence of engineering errors.

Study plan and navigation

MODULE	MINUTES	FOCUS
1	35	Engineering Quality Assurance Fundamentals
2	45	Requirements, Design Basis & Traceability
3	45	Design Review, Checking & Independent Verification
4	45	Document Control, Configuration Management & Engineering Change Control
5	35	Nonconformance, Corrective Action & Continuous Improvement
6	35	Applied Quality Scenarios, Records & Professional Accountability

The 240-minute allocation is an instructional plan. Individual reading and application time may vary.

How to complete this course

Read all modules in this course and study the examples and case explanations. No separate module acknowledgement is required. When you have finished reading, select Continue at the bottom of the BuilderCE lesson page to begin the 10-question final examination. If you are reading the PDF, return to that webpage first. A score of at least 70% is required to pass. Retakes are permitted.

Hyatt Regency evidence · Principal references

Scope. This course teaches broadly applicable engineering quality and change-control practices. It does not replace discipline-specific design standards, project contracts, owner procedures, jurisdictional requirements, or the responsible engineer's professional judgment. Where a governing requirement is more specific, that requirement controls.

MODULE 1

Engineering Quality Assurance Fundamentals

Planned instructional time: 35 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

1.1 Quality is produced by the design process, not added at the end

A final drawing review is valuable, but it is too late to be the entire quality system. Many serious engineering errors begin upstream: the design team receives incomplete criteria, a code edition is assumed rather than confirmed, a load or operating condition is misunderstood, a discipline interface is never assigned, a calculation uses an old geometry, or a field change is made without updating the design basis. By the time a final checker sees the deliverable, the error may be embedded across several documents and may appear internally consistent because every affected person worked from the same incorrect assumption.

Quality assurance therefore emphasizes prevention. The organization defines how work is initiated, how requirements are captured, how technical responsibility is assigned, how reviews are planned, what independence is required, how documents are controlled, how changes are authorized, and what evidence must exist before release. Quality control emphasizes detection: checking calculations, comparing drawings to criteria, verifying model outputs, inspecting completed work products, and confirming that comments were resolved. The two functions are complementary. Prevention without

checking can miss execution errors; checking without process control can repeatedly discover the same avoidable problems at the end of the job.

CONTROL	QUALITY ASSURANCE	QUALITY CONTROL
Purpose	Prevent and systematically manage error	Detect errors in the work product
Typical activity	Plan roles, requirements, review and change controls	Check calculations, drawings and model outputs
Evidence	Defined process and its implementation	Technical findings and verified resolution
Relationship	Creates reliable conditions for engineering work	Tests whether the actual work meets criteria

Figure 1. Engineering quality assurance strengthens the process while quality control verifies the work product.

1.2 Process quality, technical quality, and professional judgment

A project can have excellent document-control discipline and still produce a technically weak design. It can also have technically strong individual engineers whose work becomes unreliable because revisions are not coordinated or because the organization cannot identify the current approved documents. Engineering quality therefore has at least three dimensions. Process quality asks whether the required controls were followed. Technical quality asks whether the engineering analysis and deliverables are correct, complete, coordinated, and fit for their intended use. Professional judgment asks whether the controls and technical conclusions were appropriate to the actual consequence, uncertainty, and scope of practice.

The three dimensions should reinforce one another rather than compete. A checklist is useful when it reminds a qualified reviewer to examine the right issues; it becomes counterproductive when it is treated as evidence that no judgment is required. A review signature is useful when it represents a meaningful technical review; it becomes misleading when the reviewer lacked adequate time, competence, independence, or access to the design basis. A quality procedure is useful when it creates reliable habits and traceability; it is weak when staff learn to satisfy the form without satisfying the purpose.

The central quality question is not simply, “Was the procedure followed?” It is, “Did the combination of process, technical review, and professional accountability provide reasonable confidence that the engineering work was suitable for release?” That question is especially important when the work has unusual consequence, novel technology, complex interfaces, incomplete information, compressed schedule, or significant downstream dependency.

1.3 Roles must be clear enough to preserve independence and accountability

Most engineering work is collaborative, but collaboration should not blur technical roles. The originator develops the work and remains responsible for its technical basis within the assigned role. A checker independently examines the work for technical correctness. A discipline reviewer looks beyond arithmetic to criteria, assumptions, consistency, interfaces, and appropriateness. An interdisciplinary reviewer evaluates boundaries between technical areas. An approver determines whether the work is ready for its stated use and whether required review evidence exists. A project or engineering manager ensures that the required quality process is planned and resourced.

These roles can be combined on low-risk work when organizational procedures allow it, but the organization should understand what independence is lost. The person who created a calculation is naturally familiar with the assumptions and logic and may read what was intended rather than what is actually written. Independent review helps challenge that familiarity. Independence does not require hostility or organizational separation in every case. It requires enough technical distance that the reviewer can reconstruct the work rather than merely confirm the originator’s path.

Approval should also be distinguished from authorship and checking. A manager may approve a document for release but may not be the technical originator. A professional engineer may accept responsible charge for a technical segment only when the applicable jurisdictional requirements are met. A project administrator may control document status without having authority to resolve a technical comment. Quality systems are stronger when the evidence clearly identifies which role each person performed.

1.4 Graded controls prevent both under-review and unnecessary bureaucracy

A useful quality system is risk based. A minor calculation with limited consequence and easy reversibility may be adequately controlled through a competent peer check and a standard release checklist. A safety-related structural calculation, a new process hazard control, or a design that affects several disciplines may require formal independent verification, interdisciplinary review, documented hold points, and approval by the engineer of record or other designated technical authority. The objective is to match rigor to consequence.

Risk-based control should consider more than direct life safety. Cost of rework, service interruption, environmental consequence, regulatory exposure, public visibility, novelty, complexity, one-of-a-kind fabrication, long procurement lead time, and difficulty of correction can justify deeper review. The organization should also consider how many downstream decisions depend on the item. A seemingly modest design input can become high consequence when it is consumed by many other analyses.

Grading does not mean that high-pressure schedules justify skipping controls. Schedule pressure is itself a risk factor because it increases the chance of incomplete coordination, stale inputs, and premature release. When a required review cannot be completed, the response should be an explicit risk decision with defined limitations, temporary status, and follow-up - not an undocumented assumption that the review will somehow be completed later.

SIGNIFICANCE	PROPORTIONATE REVIEW
Lower consequence / routine	Competent peer check and standard release controls
Moderate interfaces or uncertainty	Documented checking, coordination and tracked comments
High consequence or difficult correction	Independent verification and explicit hold points
Very high consequence / unusual work	Specialist challenge, deeper verification and designated technical authority

Figure 2. Review rigor should scale with technical consequence, interface complexity, and the difficulty of correcting an error after release.

1.5 Common quality breakdowns reveal where the system needs control

Engineering organizations repeatedly encounter the same families of quality problems. Requirements are incomplete or contradictory. Design criteria live in email instead of a controlled basis document. Assumptions are not recorded. Interface information is exchanged informally and changes without notification. Calculations are updated while drawings remain unchanged. Reviewer comments are closed because a response was entered, even though the affected design was not revised. Field deviations are accepted verbally. Final records are assembled after the project and do not match what was actually installed.

The presence of one problem does not automatically mean the entire quality system failed. The important question is whether the system detects the issue, controls its consequences, preserves the technical basis for disposition, and learns from recurrence. A healthy system expects that people can make mistakes and designs barriers accordingly. It does not rely on individual memory as the only protection against error.

Error-resistant design processes use multiple complementary defenses: clear criteria, checklists, independent review, automated checks where appropriate, controlled templates, standard calculations, peer discussion, interdisciplinary coordination, configuration status, release gates, and feedback from construction or operations. No single defense is perfect. Reliability comes from the way the defenses overlap.

1.6 Quality planning should occur before design production begins

A design-quality plan converts general expectations into project-specific controls. The plan does not need to be a separate large document for every project; on smaller work it may be incorporated into the project execution plan, scope memorandum, design criteria package, or discipline checklist. The important point is to decide early how quality will be achieved rather than discovering at the milestone that reviewers, criteria, and document-control needs were never planned.

A useful quality plan identifies the design deliverables, applicable procedures, technical leads, originators, independent reviewers, approval roles, major requirements sources, design-basis records, software or calculation tools, review milestones, interdisciplinary interfaces, document-control system, change process, nonconformance process, and required quality records. It should also identify special controls for high-risk or unusual work. If an external checker, specialty consultant, peer review, or owner review is needed, the schedule and scope should be defined while the work can still benefit from the review.

Planning should consider the difference between project complexity and project size. A small project can have high technical consequence or difficult interfaces. A large project can contain many routine elements that do not require the same depth of review. The plan should therefore identify critical items rather than apply the most burdensome control uniformly to every document. Graded quality planning makes the process more credible because staff understand why certain work receives deeper review.

Quality planning also defines what evidence must exist before release. Teams should know whether a calculation check sheet, design-review log, interdisciplinary coordination sign-off, model-verification record, or change-impact form is required. When evidence expectations are defined after the work is complete, the organization risks recreating records rather than documenting actual quality activity.

1.7 Human factors, workload, and communication affect engineering quality

Engineering errors are not caused only by lack of technical knowledge. Fatigue, interruption, schedule compression, unclear roles, confirmation bias, authority gradients, excessive multitasking, and fragmented communication can undermine competent work. Quality systems should recognize these human factors because procedures alone do not eliminate them.

An engineer who develops a complex analysis over several weeks may stop noticing an assumption that would be obvious to a fresh reviewer. A project manager who repeatedly hears that a milestone “must not move” may unconsciously discourage reviewers from raising issues. A junior engineer may notice a mismatch but hesitate to challenge a senior specialist. A team working across multiple software systems may assume that another discipline received a changed input. These are predictable organizational conditions, not unusual personal failures.

Practical controls include independent review, protected review time, explicit escalation paths, design pauses before major release, standardized handoffs, visible open-item lists, and a culture in which technical questions are not treated as disloyalty. Managers should be cautious about using heroic overtime as a quality strategy. Extended overtime can temporarily increase output while degrading checking, communication, and judgment.

Quality culture is demonstrated by what happens when someone raises a concern close to a deadline. If the default response is to suppress, defer, or administratively close the issue, the organization teaches staff that schedule is the real priority. If the response is to assess significance, control the risk, and document the decision, the organization reinforces professional responsibility while still allowing proportionate schedule decisions.

1.8 Engineering tools, templates, and automation require controlled use

Spreadsheets, analysis software, design applications, scripts, templates, and automated checks can improve consistency and reduce manual error, but they can also propagate mistakes rapidly. A spreadsheet with an incorrect formula can affect every project that reuses it. A template based on an obsolete code can create a standardized error. A model can produce precise results from incorrect boundary conditions. Automation increases the importance of controlling the tool and understanding its limitations.

Organizations should determine which tools require validation, qualification, or controlled templates. The rigor depends on consequence. A simple calculation template may be checked against known examples and locked to prevent accidental formula changes. A safety-critical analysis tool may require formal verification, version control, benchmark cases, user qualification, and documented model review. The user should know which software version was used and whether the project procedure requires independent verification of important results.

Automated rule checking is most effective when it supplements rather than replaces technical review. A BIM clash detector can identify geometric collisions but may not understand required maintenance clearance, construction sequence, fire rating, or professional responsibility. A code-checking routine can identify numerical violations but may not know whether the chosen code provision applies. A large-language-model assistant may summarize requirements or draft text but can invent criteria or omit context. Qualified review remains necessary.

The engineering record should preserve enough information to reproduce important results. This may include software version, key input files, model assumptions, scripts, calculation templates, and configuration settings. Reproducibility is especially important when the output becomes a design basis for later modifications.

1.9 Quality audits and technical reviews serve different purposes

A quality audit evaluates whether the organization’s defined process is being followed and whether the management system is effective. A technical design review evaluates the engineering work itself. The two activities overlap but should not be confused. An audit may confirm that every required check form was completed without determining whether the calculation was correct. A technical reviewer may identify an engineering error without evaluating whether the organization’s broader procedure is consistently implemented.

Internal audits can identify systemic weaknesses: repeated missing approvals, inconsistent revision status, weak change documentation, incomplete records, or variation between project teams. Audit findings can lead to process improvement, training, or management action. Technical reviews should remain focused on engineering adequacy, interfaces, assumptions, criteria, and consequences.

On smaller organizations, the same people may participate in both roles at different times. Role clarity matters. When performing an audit, the reviewer asks whether the process was implemented and effective. When performing design review, the reviewer asks whether the technical work is suitable. Both can contribute to quality improvement, but neither should be used as evidence that the other occurred.

1.10 Deep dive - building a project-specific design quality plan

A useful design quality plan can be built from a sequence of practical questions. First, define the engineering products that will be released: calculations, drawings, specifications, models, reports, data sheets, control narratives, design memoranda, or other technical outputs. Second, identify what each product depends on. Third, identify which products or interfaces have the greatest consequence if they are wrong. This establishes the basis for review planning rather than applying a generic process without considering the work.

For each high-significance item, decide who will originate, check, review, and approve the work. Identify whether the checker must be independent of the originator and whether outside expertise is required. Define the review stage at which the issue is best challenged. For example, a complex retaining-wall concept should receive geotechnical and structural challenge before detailed reinforcement drawings

are complete. A complicated control-system architecture should receive operational and cybersecurity input before hardware is procured.

Next, define the controlled information environment. Identify the project design criteria, governing document register, shared interface data, model locations, file naming and revision rules, and official transmittal path. Decide how the team will identify preliminary information and how changes to important inputs will be communicated. Many design errors occur because the quality plan addresses checking but not information status.

The plan should identify technical hold points. A hold point is a decision that should not be passed until defined evidence exists. Examples include approval of geotechnical criteria before foundation finalization, interdisciplinary review before a coordinated package is issued, independent verification before a safety-critical calculation is released, or design reconciliation before record drawings are finalized. Hold points protect the project from irreversible downstream commitments.

Finally, define the records that demonstrate completion. The objective is not to create evidence after the fact. The quality plan should make the normal workflow generate the evidence automatically: check sheets, reviewer markups, closed comment logs, approved design-basis revisions, change-impact records, transmittals, and final configuration indexes. If staff must reconstruct the quality record at closeout, the process is too detached from the engineering work.

1.11 Quality failure taxonomy - classify the problem before selecting the control

Quality problems can be grouped by where they enter the engineering process. Requirement failures occur when the design uses the wrong, missing, obsolete, or ambiguous criterion. Analysis failures occur when the method, model, calculation, or interpretation is incorrect. Interface failures occur when individually correct disciplines do not coordinate. Documentation failures occur when the approved engineering decision is not communicated accurately. Change failures occur when revisions are not fully evaluated or synchronized. Review failures occur when a required independent challenge is weak, incomplete, or administratively closed. Configuration failures occur when users cannot determine the approved technical state.

This classification helps organizations select the right corrective action. If a project repeatedly finds drawing mistakes caused by late mechanical data, adding another drafting check may detect more mistakes but will not control the source. The stronger action may be to establish a mechanical-to-structural interface milestone and status-controlled equipment schedule. If calculations are correct but released drawings are stale, the problem may be change reconciliation rather than engineering competence.

The same event can involve several categories. A wrong equipment load may begin as an interface failure, escape because the checker used the same stale input, and reach the field because document status was unclear. Good analysis identifies the sequence. The objective is to place controls at multiple points: prevent the wrong input, independently challenge it, and prevent obsolete output from being used.

1.12 Implementing a practical engineering quality system in a small or mid-size organization

Engineering quality does not require a large corporate bureaucracy. A small firm can establish strong controls with a concise set of practices: a project design criteria sheet, clear file and revision conventions, independent calculation checking, milestone design reviews, an interface/open-item log, a controlled change form, a final release checklist, and retention of the important technical record.

Start with the organization’s actual failure modes. If recurring problems involve stale drawings, strengthen document status and distribution. If calculations and drawings diverge, add affected-document reconciliation. If final reviews find many interface conflicts, create earlier interdisciplinary reviews. If reviewer independence is weak, define when another engineer or outside specialist is required. The system should solve real problems rather than imitate another industry’s paperwork.

Assign ownership. A quality process fails when everyone assumes someone else is maintaining it. Project managers can plan milestones and resources; discipline leads can define technical review depth;

document control can manage status; responsible engineers can approve technical decisions; management can review recurring issues and corrective actions.

Train engineers on the purpose of controls, not only the forms. Staff are more likely to use a process well when they understand the failure it is designed to prevent. Periodically audit completed projects to see whether the record can reconstruct the design basis, reviews, changes, and final configuration. Improve the process based on what the audit reveals.

The objective is a scalable system that makes good engineering easier and unsafe shortcuts harder. Quality controls should create reliable habits around the moments where engineering information changes hands, decisions are reviewed, and technical authority is exercised.

1.13 Example project quality plan - concise but technically meaningful

For a medium-complexity multidisciplinary project, a practical quality plan can fit on several pages if it addresses the controls that matter. The plan begins by listing the principal deliverables and identifying their technical leads. It names the controlled design-basis document and the official project document repository. It establishes the governing code/standard register and requires discipline leads to confirm criteria before the 30 percent milestone.

The plan identifies milestone reviews at 30, 60, 90, and final design, but it does not treat the percentages as the purpose. At 30 percent the review focuses on design basis, alternatives, major interfaces, and unresolved assumptions. At 60 percent it focuses on technical development and interdisciplinary coordination. At 90 percent it focuses on detailed calculations, drawings, specifications, constructability, and remaining interfaces. Final review focuses on closure, consistency, approvals, and release readiness. High-risk calculations are independently checked before the milestone rather than waiting for the broader review meeting.

An interface register identifies shared data such as equipment loads, utility locations, structural penetrations, hydraulic elevations, electrical demand, controls points, access requirements, and temporary conditions. Each item has an owner, receiving discipline, status, required date, and change-notification path. The register is reviewed at coordination meetings, but the controlled status is maintained in the project system so decisions do not depend on meeting memory.

The plan defines document status as working, review, approved, issued for stated purpose, superseded, and record. Technical changes after approval require an impact screen. Major changes require affected-discipline review and a documented affected-record list. Field changes are initiated through an RFI or field-change form, but any response that modifies design intent is linked to a formal design change and reconciled into final records.

Nonconformances that affect design requirements or installed work are documented with the applicable criterion, actual condition, technical disposition, approval authority, and closure evidence. Recurring issues are trended quarterly. Corrective action is required when significance or recurrence indicates a process weakness rather than an isolated error.

Before release, each discipline completes a concise checklist confirming current design basis, calculation/model checks, comment closure, interface resolution, change incorporation, document status, and required professional approvals. The project engineer performs an interdisciplinary release check on material interfaces and verifies that the revision being issued is the revision that received the required review.

At closeout, the project maintains a change index covering formal design revisions, field changes, approved substitutions, nonconformance dispositions, and other configuration-significant decisions. The record-drawing team reconciles each item and records whether it was incorporated, superseded, or found not applicable. A final audit samples major design decisions to confirm the project record can reconstruct requirement, analysis, review, change, authorization, and final configuration.

This example demonstrates that a quality plan does not need excessive forms. Its strength comes from deciding in advance where technical information is controlled, where independent challenge occurs, how changes propagate, and what evidence defines completion. Organizations can scale the formality up or down while preserving those core functions.

Module 1 key point. Engineering quality is strongest when prevention, independent detection, professional judgment, and traceable release controls work together. A quality signature should represent real technical activity, not administrative completion.

MODULE 2

Requirements, Design Basis & Traceability

Planned instructional time: 45 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

2.1 Requirements must be specific enough to design against and verify

A design cannot be demonstrated correct unless the team can identify what it was required to do. Engineering requirements come from many sources: statutes and regulations, adopted codes and standards, owner criteria, contract documents, permit conditions, environmental commitments, manufacturer limitations, system requirements, operating envelopes, site conditions, project-specific performance targets, and decisions made during design development. A quality process turns those sources into controlled design inputs.

Good requirements are clear enough that a qualified reviewer can determine whether the design satisfies them. “Provide adequate drainage” is an objective, not a complete requirement. The engineering team may need defined rainfall criteria, allowable spread, hydraulic grade limits, outfall constraints, water-quality requirements, material criteria, freeboard, and maintenance assumptions. Likewise, “design for code loads” is incomplete until the code edition, occupancy or risk category, load combinations, importance factors, site parameters, and project-specific criteria are identified.

A requirement can remain open while the design progresses, but its status should be visible. Teams frequently create hidden risk when preliminary assumptions silently become permanent criteria. If a required input is not yet known, the design basis should identify the placeholder assumption, its owner, the date it must be resolved, the analyses that depend on it, and the consequence if it changes.

2.2 The design basis is the controlled explanation of why the design is what it is

The design basis connects requirements to engineering decisions. It may be a formal design criteria manual, basis-of-design report, calculation package, project design memorandum, requirements database, model metadata, or a coordinated set of controlled records. The format matters less than the ability to reconstruct the governing criteria, assumptions, methods, interfaces, and significant decisions.

A useful design basis identifies the applicable codes and editions, source documents, environmental and site conditions, operating states, design loads, material criteria, system boundaries, performance requirements, design life, reliability expectations, modeling assumptions, accepted deviations, and major interface conditions. It should also identify items intentionally outside the scope so downstream users do not infer that they were evaluated.

Design-basis control is especially important on long projects. Codes may change, owner standards may be revised, personnel may rotate, and early decisions may be challenged years later. Without a controlled basis, engineers can unknowingly mix criteria from different project stages. A calculation may be technically correct for one set of inputs while the released drawings reflect another. The design basis provides the common reference that allows the team to detect that drift.

2.3 Assumptions are temporary requirements and should be controlled accordingly

Engineering work often proceeds before every input is available. Assumptions are necessary, but undocumented assumptions are dangerous because they can become invisible. A good assumption record states what was assumed, why the assumption was reasonable at the time, what information will replace it, who owns resolution, what work depends on it, and whether the design is conservative if the assumption proves wrong.

Assumptions should be proportionate to consequence. A temporary estimate used for early quantity planning may need little formality. An assumed geotechnical parameter supporting foundation design, an assumed utility clearance controlling alignment, or an assumed operating pressure controlling equipment selection should be formally tracked until verified. The design team should know whether an assumption is bounding, expected, or merely convenient.

When an assumption changes, the team must perform impact analysis. The new value may affect more than the calculation in which it originally appeared. It may alter clearances, member sizes, equipment ratings, electrical loads, support reactions, procurement specifications, permit conditions, or construction sequencing. The assumption register therefore becomes an early change-control tool.

2.4 Traceability connects every important requirement to implementation and verification

Requirements traceability is the ability to follow a criterion from its source through interpretation, design implementation, and verification. Traceability can be simple on a small project: a calculation cover sheet may cite the governing drawing, code section, owner criterion, and design load. On a complex system, a requirements database may link thousands of requirements to analysis documents, drawings, tests, and acceptance evidence.

The principle is the same at every scale. A requirement should have an identifiable source. The engineering team should know where it is implemented. A reviewer should be able to see how compliance was demonstrated. If the requirement changes, the team should be able to identify the affected work products. Traceability reduces the chance that criteria are lost between disciplines or project phases.

Traceability is particularly valuable for non-obvious requirements. A note in an environmental permit may control a drainage feature. A fire-life-safety requirement may affect mechanical, electrical, structural, controls, and architectural work. A utility-owner criterion may control a transportation detail that otherwise appears acceptable under a general code. A cybersecurity or reliability requirement may affect equipment architecture long before a final system test. The more distributed the implementation, the more valuable traceability becomes.

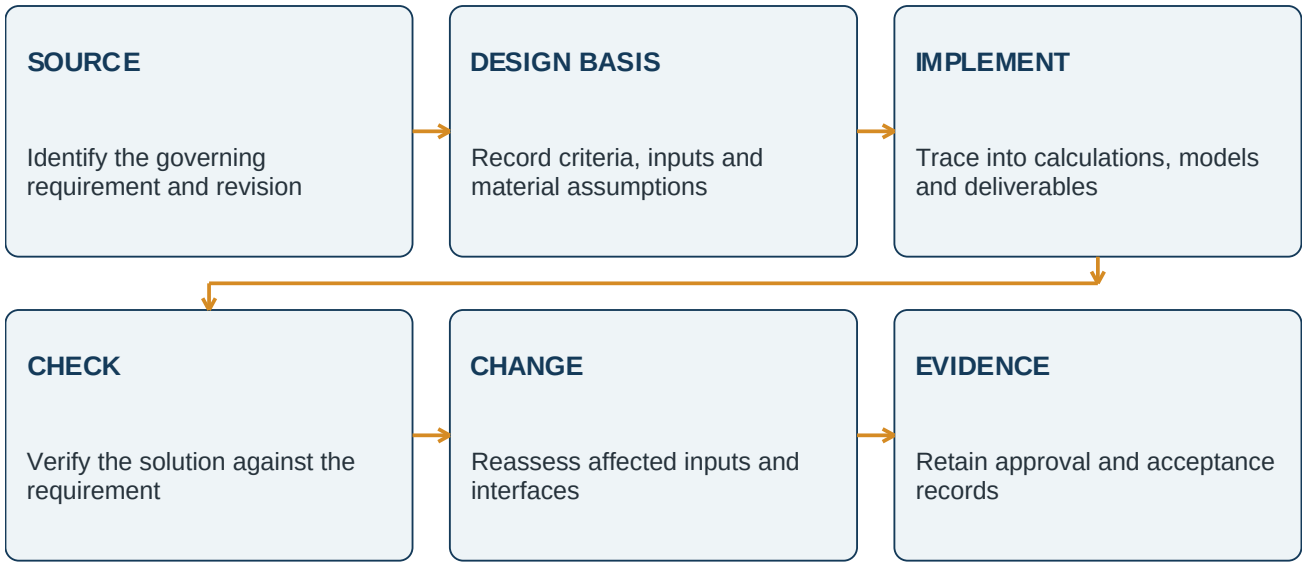

Figure 3. Traceability links the source requirement to the design basis, analysis, deliverable, and evidence of verification.

2.5 Interfaces should be treated as controlled technical inputs

Interdisciplinary interfaces are a common source of engineering error because each discipline can be internally correct while the combined design is incompatible. A structural opening may not match the mechanical equipment envelope. Electrical loads may reflect an earlier mechanical selection. A civil grading change may alter retaining-wall exposure or accessibility. A controls sequence may assume

instrumentation that the equipment package does not provide. A transportation plan may relocate a feature without updating drainage or utility coordination.

Interface control begins by identifying boundaries. The team should define what information crosses the boundary, who owns it, what status it has, when it is required, and how changes are communicated. Formal interface-control documents are appropriate for complex systems, but even a simple interface register or coordination matrix can substantially improve quality.

The key is to avoid the phrase, “I thought they knew.” If a discipline depends on another discipline’s output, that dependency should be visible. If the input is preliminary, its status should be visible. If it changes, the affected parties should receive controlled notification. Interface meetings are helpful, but meeting attendance is not a substitute for a record showing what interface decision was made and what documents were updated.

INTERFACE RECORD	INFORMATION TO CONTROL
Source and owner	Who provides and receives the technical input
Requirement and revision	The governing criterion and approved source
Technical value and units	Loads, dimensions, capacity, clearance or other shared data
Status and due date	Confirmed, provisional or unresolved; required milestone
Affected outputs	Calculations, drawings, models and procurement dependencies
Change and closure	Notification, approval, verification and retained evidence

Figure 4. Interface control identifies cross-discipline dependencies and creates a record for ownership, status, communication, and closure.

2.6 Evolving and conflicting requirements require explicit disposition

Real projects rarely have perfectly stable criteria. Owners refine needs, regulators issue comments, codes are interpreted, field conditions reveal constraints, budgets change, equipment vendors provide new data, and design alternatives mature. Quality does not require that requirements never change. It requires that change be recognized and controlled.

When two requirements conflict, the engineer should not silently choose the one that seems most convenient. The conflict should be documented, the governing hierarchy or authority identified, and the resolution approved by the appropriate party. A contract requirement may exceed a code minimum. An owner standard may conflict with a manufacturer limitation. A project criterion may become infeasible under actual site conditions. Each situation requires technical evaluation and clear authority.

The resolution should update the controlled design basis rather than live only in correspondence. Email can document the discussion, but the final criterion should appear in the source that downstream designers and reviewers actually use. Otherwise the project accumulates “ghost requirements” that are known to a few individuals but invisible to the broader team.

2.7 Verification confirms the deliverable against the approved requirement set

Verification asks whether the design output satisfies the specified requirement. It may involve independent calculation, model comparison, drawing review, checklist confirmation, simulation, test, inspection, or other evidence appropriate to the requirement. Verification should use the approved criteria, not the reviewer’s memory of what the project intended.

A strong verification record identifies the requirement or design input, the method used, the result, any exception, and the reviewer or authority accepting the conclusion. It also identifies the revision being

verified. A check performed on Revision B does not automatically verify Revision D. If later changes affect the verified condition, the verification should be revisited.

Verification should not be confused with validation. In systems engineering, verification generally asks whether the product was built or designed right relative to specified requirements, while validation asks whether the resulting system meets the intended need in its operational context. Many projects use these terms differently, so the organization should define them. The quality principle is to make the acceptance question explicit and preserve evidence that it was answered.

Applied scenario 1 - The hidden owner criterion

A civil design team completes a drainage analysis using the governing code and local criteria. During final review, an owner representative notes that the owner has an internal freeboard requirement that is more conservative than the public standard. The requirement had been sent months earlier in an email to one project manager but was never entered into the design criteria package.

The calculation may be technically correct against the criteria the engineer used, but the quality system failed to control a project requirement. Treating the issue as a simple “designer mistake” misses the systemic cause. The team should identify how owner criteria are received, approved, distributed, and incorporated into the controlled design basis. It should also perform impact review on every work product affected by the requirement.

Revise the design criteria package, evaluate affected calculations and drawings, document the resolution of any schedule or cost impact, and improve the requirements intake process so future criteria cannot remain trapped in private correspondence.

PRACTICE PRINCIPLE Quality failures often begin as information-control failures. A technically correct calculation cannot satisfy a requirement that the design team never controlled.

2.8 Codes, standards, owner criteria, and source versions must be controlled

A requirement is not fully identified by the title of a code or standard. The edition, amendment, adopted jurisdictional modification, project-specific supplement, and effective date can change the technical requirement. Projects that extend over several years can be especially vulnerable to version mixing. One discipline may use the edition identified in the permit basis while another uses a newer internal standard. A vendor may submit data based on a different edition. A model template may contain default values from an earlier project.

The design criteria should state which editions govern and who has authority to approve changes. If a later standard is voluntarily adopted for a specific issue, the decision should be documented so the team does not inadvertently mix incompatible provisions. Where the governing authority requires a transition or grandfathering decision, that basis should be retained with the project record.

Owner criteria can be equally important. Some owners maintain standards for redundancy, maintainability, preferred equipment, cybersecurity, flood resilience, future capacity, or documentation that exceed minimum codes. These requirements may not be visible to a discipline designer unless the project has a controlled intake process. A requirements index should identify the authoritative source and revision, not simply rely on a shared folder containing multiple versions.

External references should be rechecked when a project passes through major phases. A standard may be revised during design, but the existence of a new edition does not automatically mean the project should change. The engineering team should determine what the contract, jurisdiction, owner, and technical risk require. If the project remains on the earlier edition, that decision should be deliberate rather than accidental.

2.9 Changes to the design basis should trigger structured impact review

The design basis is not static. A geotechnical parameter may change after additional borings. A utility owner may revise clearance criteria. A process engineer may provide a new operating case. A code interpretation may change. A client may increase future capacity. When the basis changes, the team should treat the change as a configuration event and identify every dependent analysis and deliverable.

A design-basis change record can be simple: identify the changed criterion, source, previous value, new value, reason, approval, and affected work. For complex projects, traceability tools can identify downstream dependencies automatically. What matters is that the team does not rely on memory to discover which calculations used the old criterion.

Impact analysis should include already-approved work. Engineers sometimes assume that only current design activities need review, but an earlier released package may be invalidated by a later basis change. The team should determine whether issued documents require revision, whether permits or procurement are affected, whether construction must be controlled, and whether prior verification remains valid.

The quality process should also reconcile abandoned changes. If a proposed criterion change is evaluated and rejected, that outcome may be worth recording when the issue is likely to reappear. Otherwise a future team member may reopen the same question without knowing that it was previously analyzed.

2.10 A requirements verification matrix makes acceptance criteria visible

For projects with multiple requirements, a requirements verification matrix can provide a compact quality record. Each row identifies the requirement, source, responsible discipline, implementation location, verification method, evidence, status, and any exception. The matrix can be integrated with a requirements database or maintained as a controlled spreadsheet on smaller projects.

Verification methods should match the requirement. Dimensional requirements may be verified by drawing review or inspection. Structural capacity may be verified by independent calculation. System performance may require simulation or test. Documentation requirements may be verified by record review. Some requirements need multiple methods at different phases, such as analysis before construction and test after installation.

The matrix is especially useful for requirements that cross disciplines or phases. A flood-resilience criterion may influence civil grading, structural elevations, electrical equipment location, mechanical openings, emergency power, and operational planning. A single row can link these implementations and make missing evidence visible.

The matrix should not become a second uncontrolled requirements list. Its requirement text or identifier should trace to the authoritative source. If the requirement changes, both the source and matrix should be updated through the change process. At closeout, the matrix can become a concise index to the evidence demonstrating that the project met its important criteria.

2.11 Detailed example - controlling an evolving equipment interface

Consider a pump station where the mechanical team selects preliminary pumps before final electrical service information is available. The preliminary pump data establishes horsepower, starting method, heat load, foundation reactions, piping geometry, controls points, and maintenance clearance. Structural, electrical, civil, architectural, and controls disciplines begin design using those data.

If the vendor later changes the motor size or equipment footprint, the change is not merely a mechanical update. Electrical demand and short-circuit assumptions may change. The foundation may require different reactions. Access doors or lifting provisions may be inadequate. Controls programming and instrumentation can change. Piping supports, ventilation, and noise criteria may also be affected. The quality system should treat the equipment data as a controlled interface input with a status such as preliminary, approved, or released for fabrication.

A disciplined interface register identifies which disciplines consume each data element. When the vendor revision arrives, the mechanical lead does not need to remember every consequence; the register prompts affected-discipline review. Each discipline either revises its work or documents no impact. The approved equipment data then becomes part of the project baseline.

This example illustrates why requirements traceability and interface control are practical tools rather than abstract systems-engineering concepts. They reduce dependence on personal memory and provide a method for coordinating change across specialized engineering work.

2.12 Discipline examples - how traceability looks in different engineering work

In transportation engineering, a roadway geometry criterion may originate in a state design manual, owner decision, design exception, traffic study, and accessibility requirement. Traceability connects the controlling values to plan sheets, profiles, cross sections, sight-distance calculations, drainage, signing, and construction details. When one criterion changes - for example design speed or lane configuration - the affected-document review should identify more than the roadway geometry sheet.

In structural engineering, the requirement chain may begin with code edition, risk category, loads, owner performance requirements, material criteria, and geotechnical conditions. The design basis connects those inputs to load combinations, analysis models, member design, connection design, foundations, and details. Verification links calculations to the drawing revision that implements the result. A later change in equipment load or opening geometry can then be traced to affected members and drawings.

In mechanical or process engineering, requirements may include operating cases, flow, pressure, temperature, redundancy, maintainability, energy performance, environmental conditions, and equipment standards. A change in one operating case can affect pipe sizing, pump selection, pressure relief, controls, supports, electrical loads, ventilation, and commissioning. Traceability allows the team to determine whether the design still satisfies the full operating envelope.

In electrical engineering, criteria such as available fault current, service voltage, load demand, reliability class, grounding, protection philosophy, and owner standards flow into one-line diagrams, equipment ratings, conductor sizing, protection settings, studies, and specifications. Vendor or utility changes may invalidate several downstream studies. A controlled requirements and interface process helps the team identify what must be repeated.

In controls and software-intensive systems, the design basis includes functional requirements, sequences, input/output points, alarms, communication protocols, fail states, cybersecurity controls, and interface definitions. These systems are especially vulnerable to hidden configuration changes because a small software or firmware revision can alter behavior without a visible drawing change. Requirements traceability and configuration records become central to verification.

2.13 Managing uncertainty without hiding it

Engineering criteria are sometimes uncertain rather than simply unknown. Geotechnical properties vary spatially, hydrologic events are probabilistic, equipment performance has tolerances, future loads are forecasts, traffic volumes vary, and operating conditions can be difficult to predict. Quality practice does not require false precision. It requires the uncertainty to be understood and incorporated into the design basis.

The engineer should identify whether uncertainty is addressed through conservative assumptions, statistical treatment, factors of safety, reliability targets, sensitivity analysis, testing, monitoring, phased decision-making, or another appropriate method. The record should distinguish uncertainty inherent in the physical system from uncertainty created by missing project information.

Reviewers should challenge assumptions that appear precise without adequate basis. A value copied from a prior project may be reasonable for early planning but inappropriate for final design unless the current project confirms applicability. The quality system should make provisional values visible so they receive attention before release.

When uncertainty cannot be resolved before release, limitations should be communicated to downstream users. A design may be approved for a defined range of conditions with monitoring or future verification required. This is stronger than presenting an uncertain assumption as a fixed fact and leaving later users unaware of the boundary.

2.14 Practical design-basis readiness review

Before detailed design begins, the team should be able to explain the technical basis without relying on one person’s memory. Confirm the governing codes and editions, owner criteria, contract requirements, permit commitments, site data, operating cases, design life, performance targets, reliability expectations,

material criteria, environmental conditions, and major exclusions. Identify which inputs are final, preliminary, assumed, or awaiting third-party information.

For each material assumption, ask whether the assumption is conservative, whether it can be verified later, what work depends on it, and who owns resolution. Confirm that conflicting criteria have been resolved by the appropriate authority. Identify requirements that cross disciplines so they are not owned implicitly by everyone and explicitly by no one.

The readiness review should end with a controlled design-basis record and an open-item list. The design can proceed with unresolved inputs when the risk is understood, but unresolved criteria should remain visible until closure. A project that cannot state its basis clearly is not ready for a meaningful final design review.

- Are all governing criteria and source revisions identified?
- Are preliminary assumptions labeled and assigned for resolution?
- Are owner and regulatory requirements incorporated into the same controlled basis used by designers?
- Are interfaces and discipline dependencies identified?
- Are design exceptions, waivers, or deviations documented with authority?
- Can a new qualified team member determine why the design uses its current criteria without reconstructing private correspondence?

2.15 Applied workshop case - transportation geometry changes after drainage design is nearly complete

A roadway project reaches 90 percent design with horizontal alignment, profile, drainage, signing, utilities, and maintenance-of-traffic plans substantially complete. A late right-of-way negotiation forces the roadway centerline to shift approximately four feet over a several-hundred-foot segment. The roadway designer updates alignment geometry and plan sheets and believes the change is minor because design speed, lane widths, and profile remain unchanged.

The change nevertheless affects drainage inlet offsets, pipe lengths, ditch geometry, utility clearances, sign foundations, pavement quantities, temporary traffic-control transitions, and potentially sight distance. The project quality process should classify the alignment shift as an interdisciplinary design change rather than a drafting revision. The roadway discipline should identify the technical delta and route the revised geometry to affected disciplines with status and required response date.

Each affected discipline performs impact review. Drainage confirms whether spread calculations, pipe slopes, structures, and outfalls change. Utilities confirm horizontal clearance and relocation impacts. Traffic and signing review device locations. The geotechnical and pavement teams determine whether the shifted footprint changes subsurface assumptions. Maintenance of traffic reviews constructability. The project manager evaluates schedule and cost only after the technical consequences are understood.

The change record should identify the previous and new alignment basis, affected document list, discipline responses, revised calculations, approvals, and final issued revisions. A final interdisciplinary back-check confirms that no plan sheet or calculation remains on the old geometry. The lesson is that physical magnitude does not determine change significance; interface density does.

2.16 Applied workshop case - electrical protection study based on stale equipment data

An electrical engineer completes a coordination and arc-flash study using equipment ratings from an approved design package. Several months later, procurement changes transformer impedance and motor sizes. The one-line diagram is revised, but the study input file is not updated because the design team assumes the study will be rerun during commissioning. Construction proceeds using protection settings from the earlier study.

The calculations may have been correct when performed, but the study is no longer verified against the approved configuration. This is a traceability and change-control failure. The change process should

recognize that transformer and motor data are controlled inputs to protection, short-circuit, and arc-flash analyses. Affected-document reconciliation should identify the study automatically.

The team should assess whether interim settings create unacceptable protection or safety risk, update the study using final equipment data, revise settings and labels as necessary, and document implementation. The final record should identify the model/input revision and equipment configuration used for the study.

Corrective action may include linking major equipment data changes to an electrical study dependency register, defining a commissioning hold point before energization, or preventing final equipment approval without confirmation of study impact. The lesson is that a technically sound analysis becomes invalid when its inputs no longer represent the controlled system.

Module 2 key point. Requirements, assumptions, and interfaces are design inputs. Control them with enough traceability that a reviewer can determine what governed the design, where it was implemented, and what must be revisited when an input changes.

MODULE 3

Design Review, Checking & Independent Verification

Planned instructional time: 45 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

3.1 Review should be planned around the maturity and consequence of the design

Design review is most effective when the purpose is defined before the review begins. A concept review should challenge the selected approach, major criteria, hazards, interfaces, constructability, and unresolved assumptions. A preliminary review should examine architecture, calculations, discipline coordination, and whether the design basis is stable enough for detailed development. A detailed review should test calculations, specifications, models, drawings, and interfaces. A final review should confirm technical completeness, comment closure, revision consistency, approvals, and release readiness.

A single final review cannot efficiently perform all of these functions. If the fundamental concept is wrong, the cost of change increases as detailed work, procurement, and construction commitments accumulate. Early reviews are therefore valuable even when they are less formal. They create deliberate decision points before the project becomes locked into an approach.

Review planning should identify the review objective, documents and data required, applicable criteria, reviewer qualifications, independence expectations, interdisciplinary participation, schedule, comment process, and closure authority. Reviewers should receive enough time to reconstruct important technical decisions. A rushed review at the end of a milestone often becomes a formatting check because the schedule no longer allows meaningful technical challenge.



Figure 5. Review depth and objectives should evolve as the design matures from concept through post-issue change.

3.2 Discipline review and interdisciplinary review answer different questions

A discipline review asks whether work within a technical specialty is correct, complete, and consistent with the discipline's criteria. A structural reviewer may examine loads, analysis assumptions, member capacity, connection design, detailing, and code provisions. An electrical reviewer may examine demand, fault current, protection, conductor sizing, grounding, and equipment ratings. A transportation reviewer may examine geometry, drainage, traffic control, signing, pavement, and applicable standards.

Interdisciplinary review asks whether the technical specialties work together. It focuses on boundaries, shared geometry, loads, clearances, equipment requirements, construction sequencing, operations, controls, utilities, and other interactions. This review often detects errors that a discipline review cannot because each individual discipline sees only its own internally consistent package.

The two reviews should not be collapsed into an undefined "project review." Review evidence should make clear what was actually examined. A project meeting where discipline leads briefly scan drawings is not equivalent to a structured interdisciplinary review with identified interfaces and comment closure. Conversely, not every project needs a formal multi-day review. The level of formality should match complexity and consequence.

3.3 Independent checking should reconstruct the logic of the work

Independent checking is stronger than proofreading and stronger than rerunning the originator's spreadsheet without questioning inputs. The checker should understand the technical objective, identify the governing criteria, verify important inputs, test units and conversions, examine equations or model settings, confirm boundary conditions, review assumptions, assess reasonableness, and compare the conclusion to the associated deliverable.

For calculations, a useful check may include selected hand calculations, alternate methods, order-of-magnitude estimates, independent software runs, or targeted reconstruction of critical load paths. The checker does not necessarily need to duplicate every line. The check should concentrate on the decisions and failure modes that matter. For models, the reviewer should examine geometry, element properties, loads, constraints, combinations, convergence, mesh sensitivity where relevant, and whether output is interpreted correctly.

Independence also includes independence from the originator's expectations. If the reviewer begins with the answer and merely looks for evidence supporting it, confirmation bias can defeat the control. A disciplined check starts from criteria and inputs. The reviewer should be willing to reach a different conclusion and should document unresolved differences until they are technically reconciled.

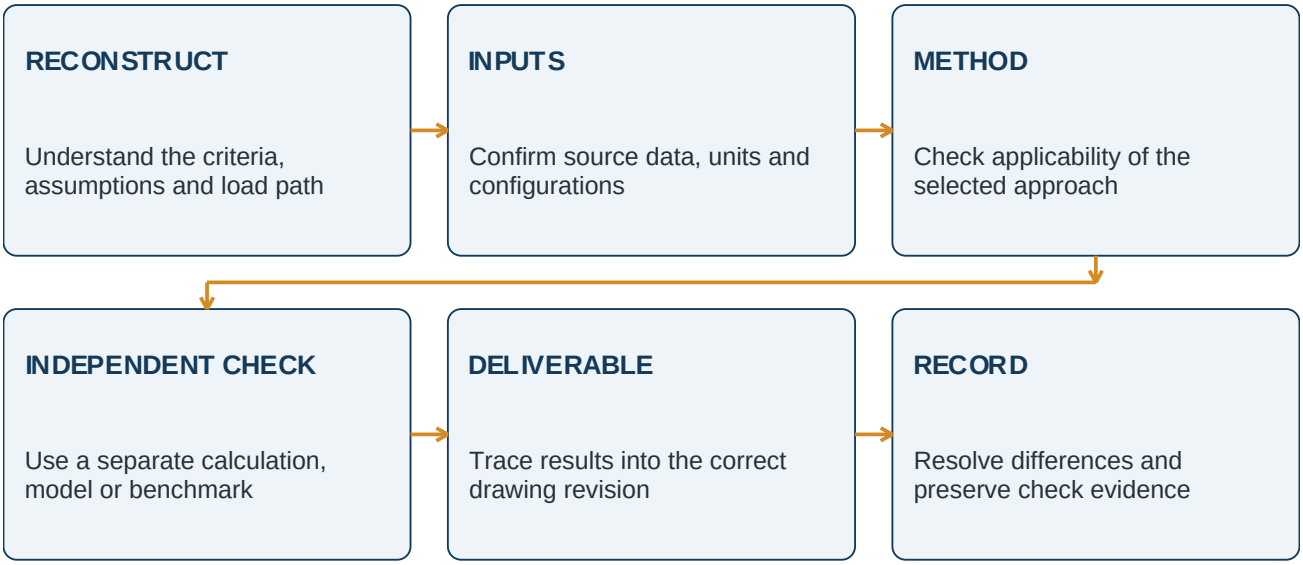

Figure 6. Independent checking reconstructs inputs, method, execution, interfaces, results, and the review record.

3.4 Review drawings, calculations, specifications, models, and reports as a coordinated system

Engineering deliverables are connected. A calculation determines a required capacity; a drawing specifies geometry; a specification controls materials or workmanship; a model establishes operating behavior; a report documents assumptions and conclusions. Quality review should confirm not only that each document is reasonable but that the documents agree.

Common mismatches include a drawing showing a member size different from the calculation, a specification referencing an obsolete standard, a model using an equipment rating that changed during procurement, a report describing a design assumption that the final drawing no longer uses, or details that conflict with general notes. Automated comparison can help, but many inconsistencies require technical understanding.

Reviewers should also examine completeness. The absence of an error is not the same as a complete design. Missing details, unassigned interfaces, unspecified acceptance criteria, unclear limits of work, missing references, and ambiguous notes can create construction or operational risk even when every shown element is technically correct. Completeness review asks whether a qualified user can implement the design without inventing critical technical decisions in the field.

3.5 Constructability, operability, maintainability, and safety are review lenses, not separate professions

Where relevant to the engineering scope, design review should consider how the design will be constructed, operated, maintained, inspected, and safely accessed. This does not mean the design engineer assumes the contractor's means-and-methods responsibility or the operator's full operational responsibility. It means the engineer considers foreseeable conditions that affect whether the design can achieve its intended function.

Constructability review may identify impossible access, conflicting sequences, inadequate tolerances, temporary-state loads, equipment installation envelopes, or dependence on unsupported field assumptions. Operability review may identify controls, isolation, startup, shutdown, or abnormal operating conditions that the design did not address. Maintainability review may reveal that equipment cannot be removed, inspected, calibrated, or safely serviced. Safety review may identify hazards created by layout, access, stored energy, traffic exposure, lifting, confined space, or other foreseeable conditions.

These lenses are most valuable when the right stakeholders participate early. A constructability issue found after bid may become a change order. The same issue found during preliminary review may

require only a design adjustment. The quality system should create opportunities for relevant experience to influence the design before the cost of change increases.



A multidisciplinary review is strongest when the team can relate controlled design information to real project interfaces and field conditions. This is a contextual illustration, not a documented incident.

3.6 Comment logs are technical records and should preserve the disposition

A review comment should be specific enough that the originator understands the issue and a later reader can reconstruct the resolution. Useful comments identify the location, technical concern, governing criterion or rationale, and requested action when appropriate. Comments such as “check this” or “not sure” may be useful as markup during collaboration but are weak as final review records.

Disposition should also be technical. “Accepted” should lead to a visible revision. “No change” should include enough basis to show why the concern did not require revision. “Noted” is not a disposition when the comment identifies a substantive conflict. If a comment is transferred to another discipline, the record should show ownership and closure rather than simply moving it out of the original reviewer’s queue.

Closure should confirm both the response and the implementation. A reviewer may agree with the proposed response, but the final drawing can still fail to incorporate it. For important comments, the reviewer should verify the affected deliverable before closure. Automated workflow status can support this process, but the status should reflect technical reality rather than replace it.

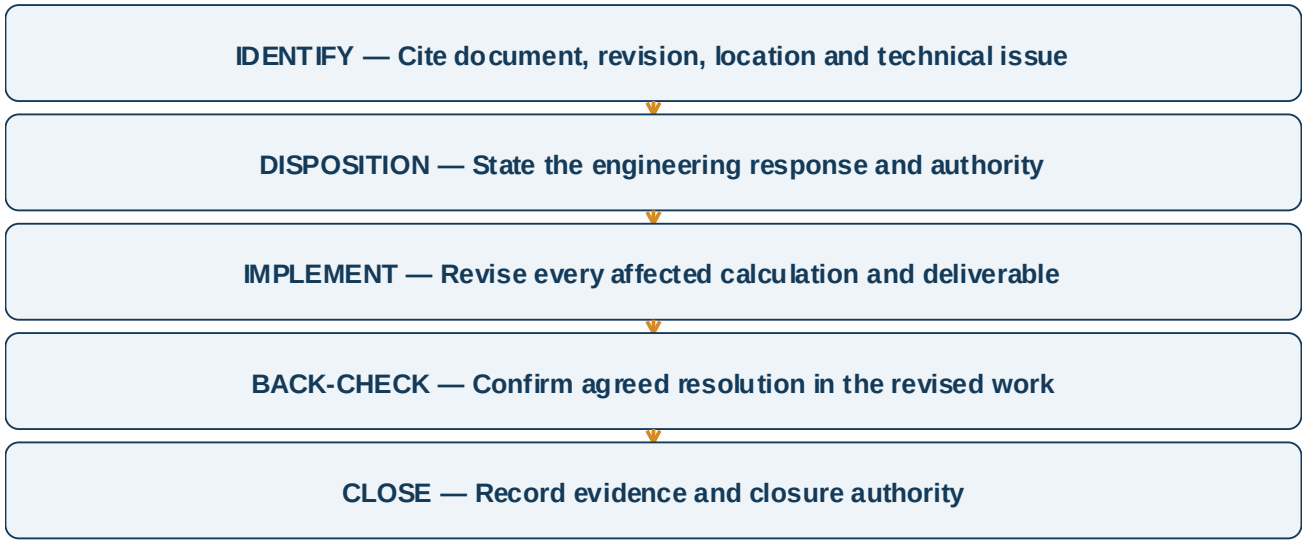

Figure 7. Technical comment closure requires a documented issue, technical response, affected revision, reviewer verification, and retained evidence.

3.7 Avoid rubber-stamp review and review by exhaustion

Two opposite review failures are common. Rubber-stamp review occurs when signatures are collected without meaningful technical challenge. Review by exhaustion occurs when reviewers generate large numbers of low-value comments, focus on formatting, or repeatedly revisit preference-based issues while material technical questions receive less attention. Both waste the independence that review is supposed to provide.

Review quality should be judged by the significance of issues identified, the quality of technical dialogue, the evidence of resolution, and the confidence gained - not by the number of comments. A mature reviewer distinguishes errors, risks, questions, preferences, and editorial improvements. The review process may classify comments by significance so that critical issues cannot be lost among minor edits.

Management should protect reviewer independence. A reviewer should not be pressured to close a comment simply because the milestone is due. If a disagreement cannot be resolved at the working level, the process should define technical escalation. Escalation is not failure; it is a control for situations where qualified professionals reach different conclusions on an issue that matters.

Applied scenario 2 - The administratively closed comment

During final design review, an independent checker comments that a retaining-wall calculation uses a groundwater elevation inconsistent with the geotechnical report. The originator responds, “water table addressed elsewhere,” and the project coordinator marks the comment closed to meet the submittal deadline. No revised calculation or cross-reference is provided.

The comment status is administrative, but the technical issue remains unresolved. A later reviewer cannot determine whether the groundwater condition was evaluated, whether a different criterion was approved, or whether the calculation is simply wrong. Closing the comment without evidence defeats the review control.

Reopen the comment, identify the governing groundwater criterion, reconcile the calculation with the geotechnical basis, revise affected documents or provide a defensible no-change basis, and have the checker verify the resolution before release.

PRACTICE PRINCIPLE A review log is not a schedule tracker. Closure should represent technical resolution and implementation in the controlled design.

3.8 Calculation checking should target both arithmetic and engineering logic

A calculation check should begin with the problem definition. What question is the calculation intended to answer? What criteria govern? What inputs and assumptions are used? What is the expected order of magnitude? A reviewer who starts with the originator's final number may overlook a conceptual error because every intermediate step appears mathematically consistent.

Arithmetic and units remain important. Many engineering errors arise from unit conversions, sign conventions, incorrect load combinations, transposed dimensions, or formulas copied from another condition. Checkers should verify enough of the arithmetic to develop confidence in execution, but the depth should reflect risk. Critical calculations may require a largely independent reconstruction; routine work may be adequately checked through selected calculations and reasonableness tests.

Reasonableness is one of the strongest quality tools. Compare the result to simple bounds, previous experience, alternate methods, known physical behavior, or limiting cases. If a finite-element model predicts deflection an order of magnitude smaller than a simple beam estimate, the discrepancy deserves explanation even if the software converged. If a hydraulic model indicates that doubling flow has almost no effect on headloss, the reviewer should challenge the inputs or interpretation.

Calculation checking should also follow the output into the deliverable. A correct calculation that is not reflected on the drawing or specification does not create a correct design. The check record should identify the associated document revisions and significant interfaces.

3.9 Model-based engineering needs review of configuration, assumptions, and interpretation

Complex engineering increasingly relies on digital models: structural finite-element models, hydraulic networks, power-system studies, process simulations, traffic models, computational fluid dynamics, geospatial models, BIM coordination models, reliability models, and control-system simulations. Model review requires more than confirming that the program runs without errors.

The reviewer should understand model purpose, software version, geometry, properties, loads or boundary conditions, solver settings, simplifications, calibration, and how results are extracted. Inputs should trace to controlled sources. Model naming and storage should make the approved version identifiable. When a model is revised, the team should determine whether prior reports, plots, design tables, or drawings need regeneration.

Sensitivity analysis can reveal hidden dependence on uncertain inputs. A model may produce a precise answer that changes dramatically with a small adjustment to one assumption. Where uncertainty matters, the engineering record should identify the range considered and the basis for the selected design value. Model complexity should not be used to hide uncertainty.

Independent model review can use alternate methods. A simple hand check, reduced-order model, historical data, or separate software may be sufficient to challenge important results. The objective is not to reproduce every detail; it is to verify that the model behaves consistently with physical expectations and project criteria.

3.10 Specifications and notes require the same quality discipline as calculations

Specifications often contain critical engineering requirements that drawings do not show: materials, testing, tolerances, installation criteria, performance requirements, submittal requirements, commissioning, workmanship, and acceptance criteria. Because specification sections are frequently reused, they are vulnerable to stale requirements and project-inapplicable language.

A quality review should confirm that referenced standards are current for the project basis, defined options are resolved, units and values match the drawings, product requirements are technically feasible, and responsibilities are clear. Requirements should not conflict across sections. A specification should not require a test method different from the drawing notes or call for material properties inconsistent with the calculation basis.

General notes and standard details deserve similar scrutiny. Standardization improves quality only when the standard remains applicable. Teams should avoid inserting broad boilerplate that contradicts project-

specific conditions. The design review should distinguish requirements that are intentionally standard from those that need project tailoring.

Coordination between drawings and specifications is particularly important when responsibility is split between different authors. The review plan should identify who owns cross-document consistency rather than assuming each author will notice the other document's changes.

3.11 Reviewer competence and independence should match the question being reviewed

A reviewer does not need to be identical to the originator, but the reviewer should be qualified to evaluate the technical issue. A senior manager with broad experience may be an excellent project reviewer but may not be the appropriate independent checker for a specialized dynamic analysis. A specialist may be technically qualified but too closely involved in developing the underlying method to provide the desired independence.

The organization should define the degree of independence needed. Options include a peer who did not originate the work, a separate discipline lead, an engineer from another office, an external specialist, or a formal third-party peer reviewer. Higher consequence, novelty, or conflict may justify greater independence.

Independence also involves incentives. A reviewer whose performance depends entirely on meeting the originator's schedule may face pressure to accept unresolved issues. Contractual structures can create similar pressure. A mature quality program protects the reviewer's ability to raise a material concern and provides escalation when project leadership disagrees.

Reviewer competence should be documented where procedures or contracts require it, but titles alone are not enough. The assigned reviewer should have relevant technical experience and adequate access to the design basis. If a review crosses multiple specialties, use multiple reviewers rather than relying on one person to provide superficial coverage outside their expertise.

3.12 Design review meetings should convert discussion into controlled decisions

Meetings are useful for resolving complex design issues because participants can exchange context quickly. However, verbal agreement is fragile. The quality process should convert important meeting decisions into controlled records: review comments, decision logs, design-basis updates, interface records, or approved action items.

A good review meeting identifies the issue, evidence, alternatives, technical decision, responsible person, affected documents, and closure date. The record should distinguish discussion from approval. Phrases such as "team generally agreed" are weak when the decision changes design criteria. The responsible technical authority should be identifiable.

Meeting minutes should not attempt to transcribe every conversation. They should preserve material decisions and open items. Participants should have an opportunity to correct significant misunderstandings. When a decision changes a controlled requirement or design basis, the source document should be updated rather than relying on the minutes as the only record.

This discipline is especially valuable when the project later faces a claim or technical question. The record shows not only what decision was made but that it was considered by the appropriate participants before implementation.

3.13 Worked review pattern - from design basis to final back-check

Assume a bridge design calculation determines a bearing reaction that controls a pier cap. A meaningful review begins by confirming the governing design criteria: code edition, load combinations, live-load model, impact factors, temperature cases, construction stages, and any project-specific requirements. The reviewer then confirms that the structural model uses the correct geometry, support conditions, material properties, and load paths.

The checker performs selected independent calculations or alternate-model comparisons for critical reactions. Results are compared for reasonableness, and differences are resolved. The reviewer then

traces the approved reaction into the pier-cap calculation, bearing schedule, drawings, and specifications. If the reaction changed after a prior review, the checker identifies whether downstream work was re-evaluated.

During drawing review, the reviewer confirms dimensions, elevations, reinforcement, notes, and details are consistent with the calculation. Interdisciplinary review checks clearances, drainage, utilities, construction sequence, and other interfaces. Review comments are logged with specific references. The originator revises the work and responds with technical dispositions.

Back-check verifies that the final revised calculation and drawings contain the agreed changes. Approval then confirms that required review evidence exists for the correct revision. This sequence illustrates why “calculation checked” is not enough. Quality review connects criteria, analysis, interface, document, and final release.

3.14 Reviewing software outputs and automated design reports

Automated design software can produce long reports containing thousands of checks. Reviewers should not equate report volume with confidence. The first task is to confirm that the modeled problem matches the physical and regulatory problem. Inputs, units, assumptions, code options, load combinations, geometry, and boundary conditions should be reviewed before detailed output.

Next, identify the controlling conditions. Which members, scenarios, nodes, devices, or constraints govern? Are the results stable under reasonable input variation? Do warnings, convergence messages, or default assumptions exist? If the software selects design parameters automatically, verify that the selections match project requirements and constructability.

Spot checks should target important outputs using independent methods. A structural design program can be compared to hand calculations for representative members. A hydraulic model can be checked against continuity and simplified headloss estimates. A power study can be checked against basic system calculations. The reviewer should understand why the software result is plausible.

Finally, confirm that the report revision corresponds to the model and final deliverables. A common quality error occurs when a model is updated after the report is exported or when a design table is manually transferred incorrectly. The record should preserve the relationship among source model, output report, reviewer evidence, and released design.

3.15 Review depth for specifications, narratives, and design reports

Narrative technical documents can create different review risks than drawings. A basis-of-design report may state criteria that later become obsolete. A control narrative may describe system behavior not fully represented in drawings. A specification may include performance language that transfers design responsibility ambiguously. A technical memorandum may reach a conclusion without preserving key assumptions.

Reviewers should identify the intended audience and use. A report used only for internal analysis may tolerate different formatting than a document supporting permitting, public communication, or contractual requirements. However, technical accuracy, completeness, and truthful limitation statements remain important.

Cross-reference review is valuable. If a report cites a calculation, drawing, model, or standard, verify that the referenced revision is appropriate. If the narrative states that a condition is “code compliant,” confirm that the underlying analysis actually evaluates the relevant code requirement. Avoid broad claims that exceed the evidence.

The final back-check should include significant narrative changes. Editing technical language can alter meaning. A seemingly minor wording change from “may” to “shall,” or from “typical” to “maximum,” can change responsibility or acceptance criteria and may warrant technical review.

3.16 Extended scenario - incomplete peer review on a novel design

A firm designs a novel lightweight canopy using a material system with which the project engineer has limited experience. The engineer is competent in structural analysis but has not previously designed the proprietary connection system. The project schedule allows a standard internal peer review but not the external specialty review originally proposed in the quality plan. A vendor engineer provides design tables and assures the team that the connection is routinely used.

The quality issue combines competence, novelty, independence, and schedule pressure. Internal review may be adequate for the overall structural model while still being insufficient for the specialized connection behavior. Vendor information is useful but not independent of the product supplier. The quality plan already identified external specialty review because the technical risk was recognized early.

A defensible decision would separate the portions of work the internal team can verify from the specialized issue requiring additional competence. Options may include obtaining the planned external review, limiting the internal engineer's approval scope, requiring sealed delegated design where permitted and appropriate, or revising the design to use a system within established competence. Simply documenting that schedule prevented the specialty review would not reduce the technical uncertainty.

This scenario shows why quality planning, graded controls, and professional competence intersect. The purpose of the quality plan is to identify needed independent expertise before the deadline. If circumstances change, the organization should make a deliberate technical-risk decision rather than silently degrading the planned control.

3.17 Review sign-off, sealing, and professional communication

Quality-system approvals and professional engineering approvals can look similar in an electronic workflow, but they represent different functions. A checker may sign to document that specified portions of a calculation were independently examined. A discipline lead may sign to indicate that internal review is complete. A project manager may approve release for contractual or schedule purposes. A licensed professional may sign and seal engineering documents when required by law and when the work satisfies responsible-charge requirements. The workflow should not allow these meanings to blur.

A useful approval record identifies the role, document revision, date, and scope of review. If a professional seal is required, the sealed document should be the same controlled revision that completed the required review. Late technical edits after sealing should enter the revision process and receive the level of re-review appropriate to the change. Copying a prior approval block onto a changed document without renewed review undermines both configuration control and professional accountability.

Professional communication around approval should also be precise. Statements such as "approved," "reviewed," "accepted," "no exceptions taken," or "conforms to design intent" can carry different meanings under contracts and organizational procedures. Engineers should use terminology consistent with their actual role and avoid language that implies a broader review than was performed. If review was limited to specified technical aspects, the record should make the limitation clear.

When disagreements remain, the final record should preserve the technical resolution. A reviewer who raises a material concern should not be asked to sign a generic approval that implies agreement unless the concern is resolved or the approval role is explicitly limited. If another authorized engineer makes the final technical decision, the record should identify that authority and basis. Clear role language protects the project from the false appearance that everyone performed the same review.

These principles apply to digital workflows as strongly as paper. Electronic signatures, approval buttons, automated stamps, and workflow states should be configured so users understand what action they are taking. Training should emphasize that an electronic click can represent a professional decision and should not be treated as an administrative convenience. The quality system is credible when approvals accurately reflect the technical work performed and the authority exercised.

ROLE	TECHNICAL FUNCTION
Originator	Develop the work and record its design basis
Checker	Independently reconstruct important technical work
Discipline reviewer	Evaluate criteria, method, assumptions and adequacy
Interdisciplinary reviewer	Evaluate shared inputs, boundaries and consequences
Approver / release authority	Confirm readiness for intended use and required evidence

Figure 8. Different workflow signatures represent different technical roles; approval scope, independence, and authority should remain explicit.

3.18 Practical independent calculation and model check

A calculation check should begin before the arithmetic. Confirm the problem statement, expected behavior, governing criteria, and input sources. Identify what parts of the result actually control the design. Use independent reasoning to challenge the originator's path. Check units and signs, but also check whether the correct physical problem is being solved.

For software-based work, identify software version, model revision, settings, boundary conditions, load cases, combinations, convergence or warning messages, and post-processing assumptions. Compare selected results to hand estimates, alternate models, historical benchmarks, or limiting cases. Examine sensitivity to uncertain inputs where it could affect the design decision.

Trace the checked result into the deliverable. Confirm the drawing, schedule, specification, or report uses the same value and revision. A completed check is not only a mark on a calculation; it is evidence that the engineering conclusion and released communication are aligned.

- What requirement or decision does this calculation support?
- Are all important inputs traceable to controlled sources?
- Do units, signs, load combinations, boundary conditions, and assumptions make physical sense?
- Has the checker used at least one independent reasonableness or alternate-method test for important results?
- Are software defaults and settings appropriate and documented?
- Does the associated drawing or specification reflect the checked result?
- If the calculation was revised, were prior check results invalidated or repeated as necessary?

3.19 Practical interdisciplinary design review

Interdisciplinary review is most effective when the team reviews interfaces rather than simply asking every discipline whether its own work is complete. Identify the physical and functional boundaries: loads, clearances, penetrations, elevations, utilities, equipment, control points, access, drainage, power, communications, temporary conditions, construction sequence, and maintenance needs. Use coordinated drawings, models, interface registers, and open-item lists to make the boundaries visible.

Reviewers should ask whether one discipline is relying on preliminary information from another and whether changed inputs have propagated. Confirm that shared geometry and data are consistent. Identify requirements that have multiple implementations, such as flood resilience, fire/life safety, emergency power, accessibility, environmental controls, or reliability.

The review record should capture material conflicts, technical decisions, responsible parties, and required document updates. Avoid closing interface issues through meeting discussion alone when the design documents remain inconsistent.

- Which shared inputs are still preliminary?
- What changed since the previous interdisciplinary review?

- Where do two disciplines independently define the same geometry, load, or requirement?
- Are constructability, access, operability, maintainability, and temporary conditions adequately considered within the engineering scope?
- Are open interface decisions assigned to a specific technical owner?
- Does closure include verification that affected documents were updated?

3.20 Applied workshop case - structural calculation software update changes default behavior

A structural group upgrades analysis software between preliminary and final design. The new version changes a default setting for member stiffness modifiers. Most projects explicitly define the modifier, but one legacy template leaves the setting at software default. An engineer opens the existing model in the new version, makes several geometry updates, and reruns analysis. The results change modestly but remain plausible, so the engineer assumes the difference reflects the geometry revisions.

The quality issue involves tool configuration and reproducibility. A controlled software-change process should identify material changes between versions, benchmark important templates, and require users to confirm project-critical defaults. The model record should identify software version and relevant settings so a reviewer can reproduce the analysis.

An independent checker reviewing only final member unity ratios may not detect the source of the change. A stronger check compares important reactions, drifts, modal behavior, and member demands to prior results and asks whether differences are explained by intended design changes. Unexpected changes should trigger model-configuration review.

If the new default is appropriate, the team should document the basis and determine whether previously reviewed results need re-verification. If the old project basis should remain, the model setting should be restored explicitly rather than relying on default behavior. The organization should update the legacy template so future projects cannot repeat the ambiguity. This case shows how software configuration is part of engineering configuration.

Module 3 key point. A meaningful design review is planned, technically independent where appropriate, proportionate to risk, and closed only when the affected engineering record reflects the resolved decision.

Design review evidence from the Hyatt Regency walkways

The NBS investigation drawing makes the walkway load path visible. In the revised separate-rod arrangement, the fourth-floor connection carried the load transferred from the lower walkway as well as its own walkway load. Checking that the revised rods fit was not an adequate check of the affected connection.

The original arrangement also lacked the required capacity. The lesson therefore extends beyond rejecting a shop-drawing change: review the governing demand, capacity and complete load path. An investigation drawing explains a failure; it is not an approved construction detail.

Demonstrate the fourth-floor box-beam connection capacity for all loads introduced by the hanger arrangement. Closure requires the supporting calculation, the corresponding revised detail and documented qualified review. A response marked “noted” does not establish that the detail and calculation agree.

Load-path interpretation. For a simplified comparison with equal walkway contributions P at a hanger location, the revised upper box-beam connection carries P + P = 2P. This relative-demand comparison explains the change in load path; it is not a capacity calculation or a reconstruction of the actual crowd loading.

REVIEW CONTROL	EVIDENCE NEEDED FOR CLOSURE
Demand	All loads reaching the affected connection, including transferred lower-walkway load
Capacity	Calculation for the actual connection geometry and material
Implementation	Revised detail synchronized with the checked calculation
Authority	Qualified technical review and a traceable release decision

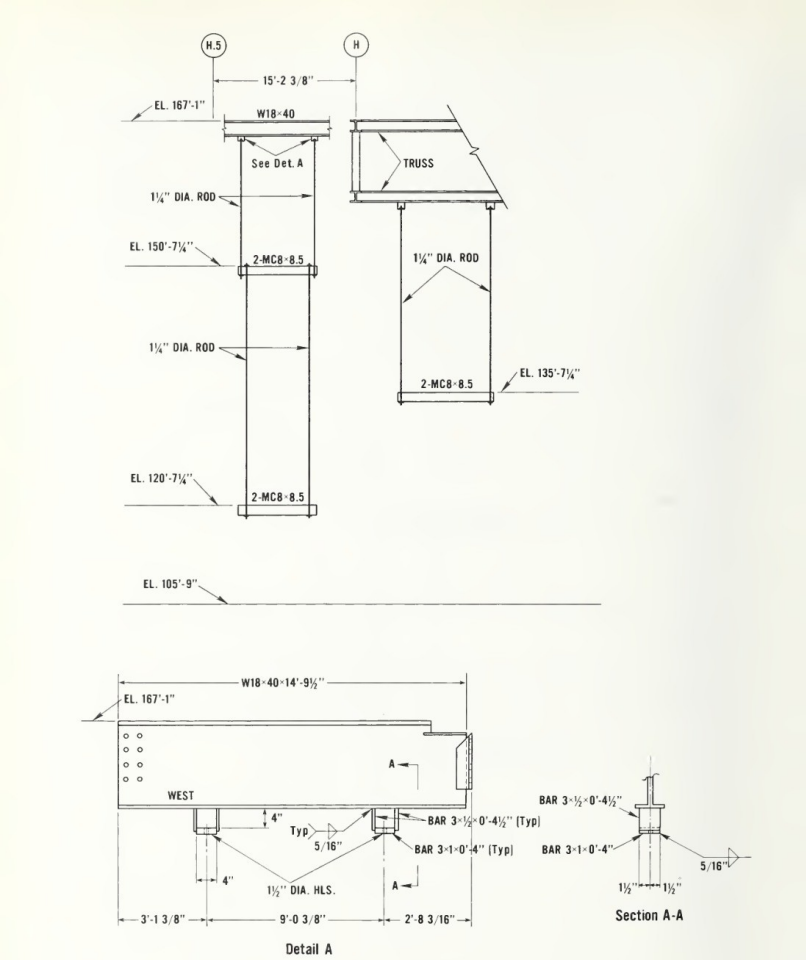

Figure 3.11 Elevation of typical box beam-hanger rod assembly.

28

Documented investigation evidence. Box-beam and hanger-rod geometry. NBS Building Science Series 143, Figure 3.11, printed p. 28. The original detail also lacked the required capacity. Open the primary source

Document Control, Configuration Management & Engineering Change Control

Planned instructional time: 45 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

4.1 Document control answers a simple field question: What information is approved for use right now?

Engineering teams generate working files, review files, approved files, issued files, superseded files, and final records. If users cannot tell these states apart, technically correct documents can become quality hazards. A drawing may be superseded but remain in a shared folder. A contractor may print an early revision. A calculation may be updated without a corresponding drawing revision. A report may cite a model that has since changed.

Document control establishes identification, revision, status, approval, distribution, retrieval, supersession, and retention. The control system can be sophisticated software or a disciplined manual process; what matters is that users can determine the current approved information and that superseded information is retained without being mistaken for current work.

File naming alone is not enough. “Final_v7_reallyfinal.pdf” communicates neither technical status nor approval. Controlled metadata should identify the document number, revision, title, status, approval, date, and other project-specific information needed to establish authenticity. The system should preserve previous revisions when necessary for traceability while preventing accidental use.



Figure 9. Document control distinguishes working, review, approved, issued, superseded, and retained record states.

4.2 Configuration management controls the technical state of the system, not merely files

Configuration management is broader than document control. It identifies the items that define the technical configuration, establishes baselines, controls changes, records status, and verifies that the documented configuration matches the actual approved state. Depending on the project, configuration items may include drawings, specifications, software, equipment data, calculations, models, requirements, settings, firmware, design parameters, interfaces, or physical assets.

A baseline is a controlled reference state. Once a design baseline is established, proposed changes should be evaluated against it rather than introduced informally. Baselines are useful because they allow

the team to answer, “What did we believe the approved configuration was at this point in the project?” Without baselines, a project can become a stream of revisions with no stable reference.

Configuration status accounting tracks what revisions and changes are approved, implemented, pending, or superseded. Configuration audit confirms that the technical record is internally consistent and corresponds to the intended physical or functional state. These concepts scale from aerospace and process systems to ordinary civil and building projects. The formal terminology may differ, but every engineered asset benefits from knowing its approved technical state.

4.3 A revision should identify what changed and why

Revision control is most useful when it preserves technical history. The record should identify the affected document or configuration item, revision identifier, change description, reason, originator, required review, approval, date, and distribution status. Major design changes may also need impact analysis, references to decision records, and links to affected calculations or models.

Revision descriptions such as “updated per comments” are often too vague to support later traceability. A better description identifies the technical change: “Revised beam size from W18x35 to W21x44 based on updated mechanical support load; coordinated Details 4/S502 and 7/M301.” The purpose is not bureaucratic detail. It is to let reviewers and downstream users understand the technical delta.

Revision clouds and change marks help users identify visible drawing changes, but they are not the entire change record. Some changes affect calculations, specifications, models, schedules, procurement data, or hidden parameters that are not obvious on a drawing. The change process should identify every affected record, not just the most visible document.

4.4 Engineering change control starts with impact analysis

A design change can be initiated by a client request, field condition, code update, value-engineering proposal, procurement substitution, RFI response, design error, constructability issue, test result, regulatory comment, or operational need. The source of the change does not determine whether technical control is required. If the change affects the approved design basis, configuration, or engineering conclusion, it should enter a controlled technical process.

Impact analysis asks what the change touches. Does it affect safety margins, code compliance, system performance, interfaces, loads, hydraulics, controls, utilities, clearances, procurement, permits, temporary conditions, commissioning, operations, maintenance, cost, or schedule? Which calculations, drawings, specifications, models, and reports depend on the changed condition? Does the proposed change create a new hazard or invalidate previous verification?

The most common change-control failure is local optimization. A field team solves the immediate issue, but the solution changes a dimension, load, material, sequence, or interface that another discipline depends on. The local change may be technically reasonable in isolation and still be wrong for the overall system. Controlled impact analysis forces the team to look beyond the local problem.

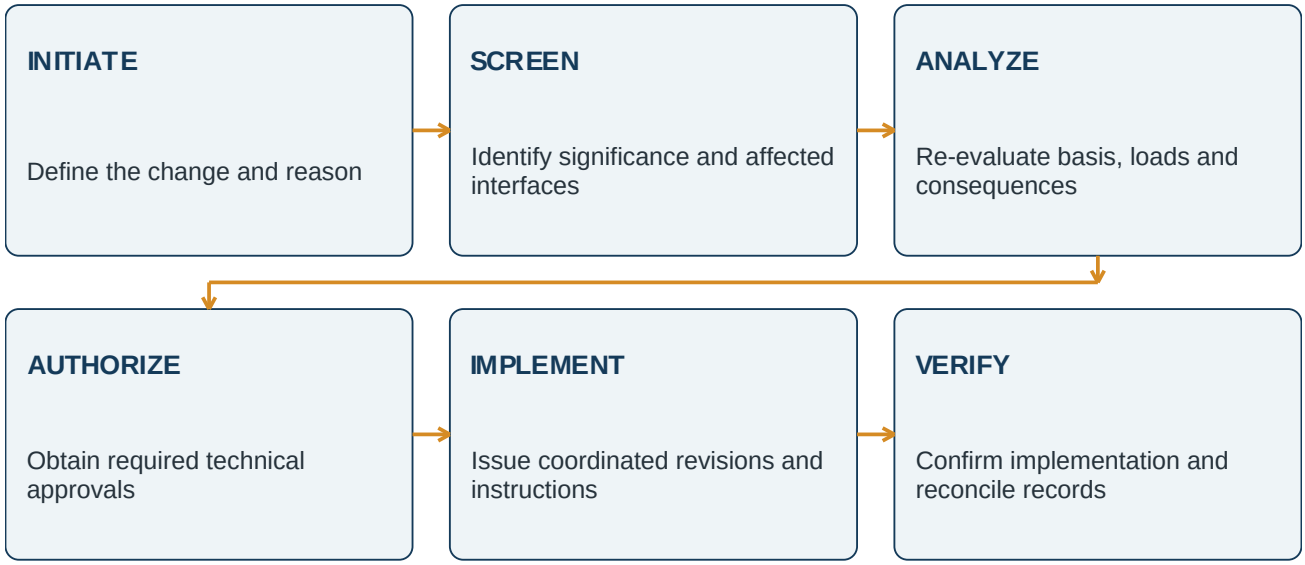

Figure 10. Engineering change control evaluates need, significance, impact, approval, implementation, verification, and record reconciliation.

4.5 Authority to propose, review, approve, and implement a change should be explicit

Many project participants can identify the need for change. Fewer have authority to approve a technical design change. A contractor may propose an alternate. An inspector may document a field condition. A vendor may recommend a substitution. A project manager may authorize cost or schedule. The responsible engineer or other designated technical authority determines whether the changed design is technically acceptable within the applicable scope and jurisdictional requirements.

The quality process should distinguish commercial authorization from technical authorization. A signed change order may authorize cost but not automatically demonstrate that the engineering change was reviewed. Conversely, an engineer's technical approval may not authorize contractual cost. Complex projects often need both. Mixing the two can create ambiguity about whether a technical issue was actually evaluated.

Emergency or field changes require special discipline because pressure encourages verbal decisions. When immediate action is necessary, the organization should define who can authorize temporary measures, how limits are communicated, what information must be captured, and how the decision is later reconciled into the permanent design record. Emergency status should shorten the path, not eliminate technical accountability.

Field change. Field conditions often create legitimate change needs. Quality depends on capturing the condition, evaluating technical impact, obtaining proper authority, and reconciling the decision into the controlled design record.

4.6 Approved changes must be synchronized across affected records

After approval, the change process is not complete until affected records are synchronized. A revised calculation may require drawing changes. A new equipment selection may change electrical demand, support loads, controls points, clearances, foundation design, specifications, and commissioning procedures. A drainage change may affect grading, utilities, quantities, permits, traffic control, and environmental commitments.

Organizations often experience "split configuration," where different parts of the record represent different technical states. One drawing reflects the approved change while another remains old. A model is updated but the calculation package is not. A field sketch is implemented but never incorporated into record drawings. This condition creates future risk because later designers, operators, and maintainers cannot know which source is authoritative.

Change reconciliation uses an affected-document list or dependency review to confirm that each related item was updated or deliberately found unaffected. The responsible party should verify implementation, not simply assume that distribution occurred. On long projects, periodic configuration audits can identify mismatches before closeout.

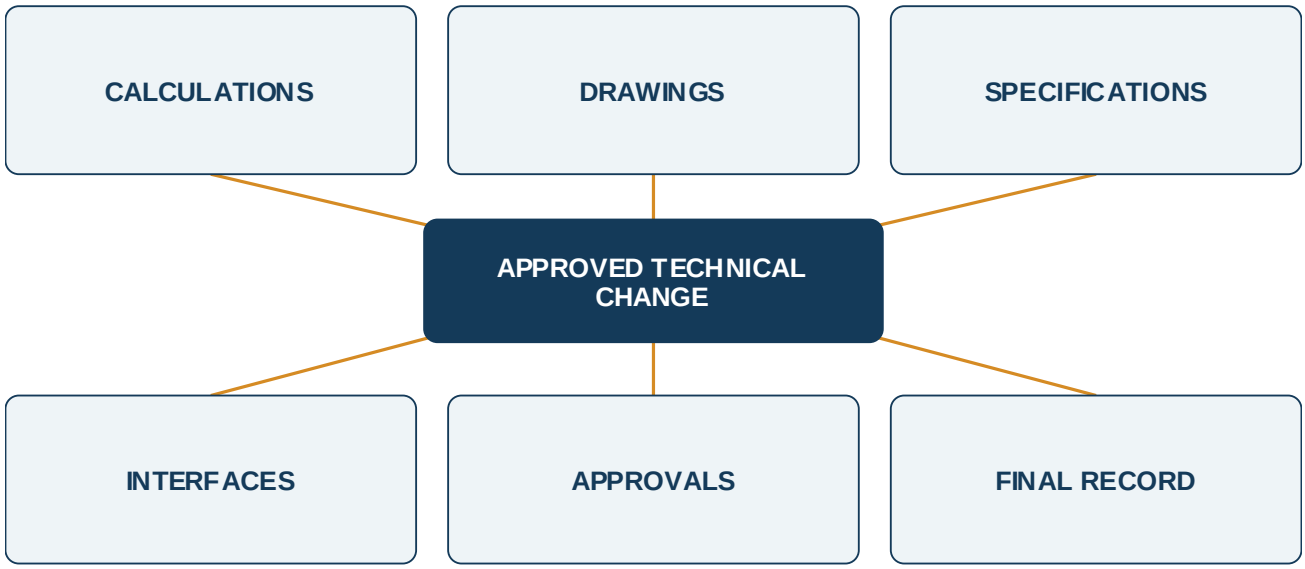

Figure 11. One approved technical change can affect calculations, drawings, specifications, models, schedules, field instructions, and final records.

4.7 Preventing use of superseded information is a field-quality control

Document control becomes most visible where engineering information is used: construction, fabrication, procurement, testing, commissioning, operations, and maintenance. A perfect revision history in the design office does not protect the project if the field cannot identify the current approved drawing. Distribution lists, electronic document-management systems, controlled print practices, transmittals, and superseded-document removal are practical quality controls.

The risk increases when multiple organizations maintain separate systems. The engineer may issue a revision through one platform while the contractor distributes through another. A vendor may work from an earlier approved submittal. An inspector may retain a marked-up print that no longer reflects the design. Quality planning should define the official distribution path and how receipt is confirmed for important changes.

Field markups are valuable records but should have a defined lifecycle. They can document actual conditions and support final record preparation, but they should not become an uncontrolled parallel design system. When a markup represents an engineering decision, the change should be captured in the formal design record with appropriate review and approval.

Applied scenario 3 - The calculation and drawing diverge

A steel support is redesigned after a vendor increases equipment weight. The structural calculation is revised and approved, but the drafting team is working from an earlier markup and the issued drawing retains the original member size. The calculation package and drawing now represent different approved states.

This is a configuration-control failure, not merely a drafting typo. The calculation may be correct, yet the released deliverable is not synchronized with the engineering decision. If construction proceeds from the drawing, the technical review of the calculation provides no protection.

Stop or control use of the affected drawing as appropriate, issue the corrected revision through the controlled process, evaluate whether related details or disciplines are affected, verify field status if work has begun, and improve the change checklist so every approved change has an affected-record reconciliation step.

PRACTICE PRINCIPLE An engineering decision is not fully implemented until the controlled work products that communicate it are synchronized.

4.8 Classify changes so the process applies the right level of review

Not every revision is a design change. Editorial corrections, formatting, and administrative metadata can often be processed with limited technical review if they do not alter meaning. Technical changes require evaluation because they can affect design intent, performance, interfaces, or compliance. Configuration-significant changes may require broader interdisciplinary review and re-verification.

A practical classification system might distinguish editorial, minor technical, major technical, and emergency changes. The definitions should be based on consequence rather than the number of drawing marks. A one-word change to a material grade can be more significant than dozens of drafting edits. The classification should determine required reviewers, approval authority, documentation, and whether previously completed verification must be repeated.

Organizations should be careful not to create incentives to classify changes downward merely to save schedule. The person initiating the change may not understand all downstream consequences. A screening reviewer or discipline lead can help determine significance before the change proceeds.

Change classification also supports management visibility. A project with a high volume of major technical changes late in design may have unstable requirements or weak early coordination. Tracking the pattern can support corrective action and schedule realism.

4.9 Vendor submittals and substitutions are part of configuration control

Supplier and contractor submittals often introduce detailed information after the design is substantially complete. Product dimensions, loads, electrical characteristics, controls, materials, fabrication details, and installation requirements can differ from design assumptions. Submittal review should therefore confirm conformity with the design requirements and identify information that affects the engineering configuration.

An approval stamp should not imply that every detail became an engineering design change. The scope of submittal review should be defined by contract and professional role. However, when vendor information changes a design input - such as equipment weight, anchor pattern, heat rejection, electrical demand, or control interface - the engineering team should process the change through the appropriate impact path.

Value-engineering proposals and substitutions deserve particular attention because they are often framed as equivalent. Equivalence should be evaluated against all relevant requirements, not only the primary function or first cost. Reliability, maintenance, warranty, code listing, redundancy, performance margin, environmental conditions, interface geometry, cybersecurity, and lifecycle support can affect technical acceptability.

The final record should identify the approved product and any resulting design changes. Otherwise record drawings may reflect the original basis while the installed configuration reflects the supplier's accepted alternate.

4.10 Field changes need a rapid but complete technical pathway

Construction and implementation inevitably reveal conditions that differ from assumptions: utility conflicts, dimensional discrepancies, inaccessible components, unforeseen subsurface conditions, fabrication tolerances, equipment substitutions, sequencing constraints, or damaged work. A field-change process should allow fast evaluation without bypassing technical control.

The initiating record should describe the observed condition, location, current approved requirement, urgency, and any proposed solution. Photographs, measurements, sketches, survey data, or test results can help establish the facts. The responsible engineer then evaluates the technical impact and determines whether the proposed action is acceptable, requires revision, or needs additional information.

Temporary measures should be clearly identified as temporary. A verbal direction to proceed may be necessary in an emergency, but the record should capture who authorized it, what limits applied, and what follow-up was required. The final permanent disposition should be incorporated into controlled documents.

Field staff should be trained to distinguish observation from design authority. Inspectors, resident engineers, and construction managers often have extensive practical knowledge, but their delegated authority varies. A clear escalation path lets them solve problems efficiently without inadvertently assuming responsibility outside their role.

4.11 BIM, common data environments, and digital twins increase the need for configuration discipline

Shared digital models improve coordination because multiple disciplines can work from integrated information, but they also create new quality questions. Which model is contractual? Which model revision was used for fabrication? Who can modify shared parameters? How are linked models updated? What is the status of clash-detection issues? How are approved field changes transferred into the record model?

A common data environment should make status and permissions visible. Work-in-progress information should not be confused with shared or published information. Model elements that drive downstream fabrication, quantity, or analysis should have controlled ownership. Model federation does not remove discipline responsibility; it creates a technical environment in which interface changes can propagate rapidly.

Digital twins and asset-information models extend configuration control into operations. If operating staff rely on model data for maintenance or future modifications, record accuracy becomes a lifecycle safety and reliability issue. Closeout should therefore reconcile approved changes and verify critical asset data rather than treating the model as an incidental visualization.

Organizations should define the authoritative source when information exists in multiple forms. If a drawing and model disagree, the contract or project procedure should state which governs and how discrepancies are resolved. Ambiguity between digital sources can be as dangerous as using an obsolete paper drawing.

4.12 Record drawings and closeout should reconcile actual configuration, not merely archive design intent

Record drawings, as-built information, and closeout models have different contractual meanings across projects. The engineering team should use terminology consistent with its scope and available evidence. Regardless of label, the quality objective is to preserve an accurate representation of the final approved condition within the limits of the engineer’s knowledge.

Closeout should collect approved design changes, field dispositions, submittal-driven changes, survey information, test results, commissioning findings, and other configuration records. The team should identify unresolved temporary conditions and confirm that superseded design information is not accidentally carried into the final set.

The process should also preserve uncertainty. If the engineer did not observe or verify a hidden condition, the final record should not imply certainty beyond the available information. Accurate limitation statements are part of defensible professional documentation.

A well-reconciled final record reduces future risk. Maintenance, renovation, emergency response, expansion, and forensic evaluation depend on knowing what was actually approved and implemented. Configuration control therefore continues through closeout rather than ending when the last design drawing is issued.

4.13 Detailed field-change example - from RFI to record configuration

A contractor discovers that a proposed footing conflicts with an undocumented utility. The first controlled record is the field condition: surveyed utility location, photographs, excavation limits, current footing geometry, and the affected drawing revision. The contractor proposes shifting the footing and rotating a column base. The request enters the engineering change process rather than being treated as a routine RFI response.

The structural engineer evaluates eccentricity, soil bearing, overturning, reinforcing, column forces, and adjacent foundations. Civil reviews grading and drainage. Utility coordination confirms clearance and access. Architectural or equipment layout may need review if the column moves. The geotechnical engineer may need to confirm whether the revised location remains within the investigation assumptions.

If the change is acceptable, the responsible engineer issues a controlled sketch or revised drawing with defined status. The contractor implements the approved change. Inspection or survey verifies the installed location. The change record identifies the affected calculation and drawings. At closeout, the final structural and site records are reconciled to the installed footing location.

If the project instead relies on a verbal direction, several risks appear: the technical basis may be lost, another discipline may continue using the original geometry, the record drawing may be wrong, and future modifications may rely on incorrect foundation information. The controlled pathway is not bureaucracy; it is the mechanism that turns a necessary field adaptation into a reliable engineering configuration.

4.14 Change control for design errors discovered after issue

When an engineer discovers an error after issue, the quality response should prioritize consequence control. Determine what documents are affected, who may be relying on them, whether fabrication or construction has begun, and whether the error creates an immediate safety or compliance concern. The engineer may need to notify project leadership and downstream users before the full corrective design is complete.

Preserve the original record. Do not overwrite or obscure the issued calculation or drawing in a way that makes the history unrecoverable. Develop the correction, perform appropriate independent review, identify all affected interfaces, and issue a controlled revision. If work has already been installed, evaluate the actual condition rather than assuming the correction can simply be applied prospectively.

The organization should distinguish the technical correction from cause analysis. The project needs the design corrected quickly; the quality system may later investigate why the error occurred and escaped review. Combining the two can delay urgent technical action. Conversely, closing the matter after correction misses the opportunity to prevent recurrence.

Communication should be factual and appropriately bounded. The engineer should state the affected condition, immediate controls, and revised technical direction without unnecessary speculation about blame. A transparent controlled response protects the project and creates a credible professional record.

4.15 Submittal and RFI workflows should not become uncontrolled design channels

RFIs and submittals are essential project communication tools, but they can unintentionally become parallel design systems. An RFI response may change a dimension, material, sequence, or technical criterion without triggering revision of the design documents. A submittal markup may approve an alternate condition that never reaches the calculation package. Years later, the official drawings still show the original design.

The quality process should screen responses for design impact. Many RFIs simply clarify existing intent and require no design change. Others reveal an omission or request a new condition. When the response changes the controlled design, the organization should create or link a formal change record and identify the documents that must be revised.

Similarly, submittal review comments should distinguish conformance review from new design direction. If the engineer directs a change through a submittal, the impact should be reconciled into the project documents. This preserves a single coherent configuration instead of forcing future users to reconstruct design from multiple platforms.

4.16 Practical engineering change impact review

When a change is proposed, resist the temptation to begin by editing the drawing. First define the current approved condition and the proposed delta. Identify why the change is needed and whether it alters a requirement, design assumption, configuration item, interface, or verified result. The more clearly the delta is described, the easier it is to perform an efficient impact review.

Review impact across technical, regulatory, operational, construction, and record dimensions. Ask whether the change affects safety margins, code compliance, permits, loads, clearances, performance, testing, commissioning, procurement, temporary states, owner criteria, or other disciplines. Identify which prior analyses or approvals become invalid.

After approval, use an affected-record list. The change should not close until the team has either updated each affected calculation, drawing, specification, model, schedule, submittal, or final record, or documented why the item is unaffected. Verify implementation when the physical configuration is involved.

- What is the current approved baseline and exact proposed change?
- Who has technical authority to approve the change?
- Which disciplines and downstream users depend on the changed information?
- What prior verification, permit, submittal, or approval must be revisited?
- What documents and models must be synchronized?
- How will the final implemented condition be verified and captured in the record?

4.17 Applied workshop case - mechanical equipment substitution during procurement

A hospital project specifies a particular air-handling unit as the basis of design. After award, the contractor proposes a substitute unit with comparable airflow and efficiency at lower cost. The mechanical reviewer confirms performance and initially marks the submittal acceptable. The substitute is slightly heavier, has a different electrical arrangement, different condensate connection, altered service-clearance needs, and a revised controls interface.

A quality-controlled submittal review should separate primary performance equivalence from interface impact. The submittal is screened as configuration-significant because the equipment data are inputs to structural, electrical, plumbing, controls, architectural, and maintenance design. The mechanical engineer identifies the changed data and routes them through the project interface process.

Structural checks support loads and vibration. Electrical reviews voltage, starting characteristics, disconnect requirements, fault-current contribution where relevant, and feeder sizing. Controls verifies point lists and sequence compatibility. Architectural reviews access and service clearances. Plumbing reviews condensate routing. The owner or facilities group may need to confirm maintainability and spare-parts implications.

If the substitute remains acceptable, affected drawings, schedules, specifications, calculations, and controls documents are revised. The approved submittal is linked to the change record, and final configuration records identify the installed unit. The project also confirms that the technical approval and commercial change authorization are both complete. The case demonstrates why “equal” equipment should be evaluated across the full requirement set rather than one headline performance value.

4.18 Applied workshop case - civil field change creates an environmental permit conflict

During site construction, the contractor encounters unsuitable soil in a proposed swale and requests a field adjustment that raises the swale bottom and shifts the alignment. The civil engineer confirms that local drainage still functions and verbally agrees to the change so grading can continue. Weeks later, an

environmental reviewer notes that the approved permit drawings include a specific treatment geometry and discharge condition tied to the original swale.

The field decision considered immediate hydraulic function but not the full requirements basis. The quality system should have identified environmental permit criteria as controlled design requirements and screened the field change for regulatory impact. A field change process that routes only to the originating discipline is insufficient when requirements cross organizational boundaries.

The engineer should document the actual constructed condition, evaluate hydraulic and treatment performance, determine whether permit modification or agency coordination is required, and establish the technically and legally acceptable final configuration. If rework is necessary, the disposition should be controlled. If the changed condition can be accepted, the permit record and final drawings should be reconciled.

Cause analysis should examine why the permit requirement was invisible during field review. Possible improvements include a requirements matrix identifying permit-linked features, a field-change checklist asking whether regulatory commitments are affected, and interdisciplinary review for changes to environmentally controlled components. The case illustrates why traceability is valuable precisely when the governing requirement is outside the immediate design calculation.

4.19 Applied workshop case - record drawings conflict with verified field conditions

At project closeout, survey data show that several underground utility lines were installed with approved field adjustments. Individual RFIs document the changes, and each was technically acceptable. The record-drawing team updates most locations but misses two because the RFIs are stored in a separate construction-management platform. The final record set is issued with a professional note describing it as based on available project information.

The limitation statement is appropriate, but the organization still has a process gap. Closeout should reconcile known approved field changes into the final controlled record. The issue is not that record drawings can guarantee every hidden condition; they cannot. The issue is that known, documented changes were not systematically captured.

A quality-controlled closeout process can use a change index listing every design revision, field sketch, RFI with design impact, nonconformance disposition, survey update, and approved substitution. Each item is marked incorporated, superseded, or not applicable. The final reviewer back-checks high-significance items against the record set.

If the missing utilities are discovered before final release, revise the record drawings and document the correction. If discovered later, issue a controlled record revision and notify the appropriate owner records system. Corrective action should integrate construction-platform change data with the design closeout process. The lesson is that record accuracy depends on configuration reconciliation, not merely a broad disclaimer.

Module 4 key point. Change control is a closed technical loop: identify, evaluate, approve, implement, verify, and reconcile. A field solution or approved calculation that never reaches the controlled design record is not a completed change.

MODULE 5

Nonconformance, Corrective Action & Continuous Improvement

Planned instructional time: 35 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

5.1 Nonconformance means a requirement was not met

A nonconformance is a condition in which a specified requirement has not been satisfied. In engineering design, the nonconformance may be a calculation, drawing, specification, model, review process, or

configuration record that does not meet the defined criteria. In implementation, it may be installed work, fabricated equipment, test results, material, or system performance that differs from the approved requirements.

The first quality action is to preserve control. Identify the affected item, requirement, status, and extent. Prevent inadvertent use or further implementation when necessary. Preserve evidence. Determine who has authority to evaluate the condition. The objective is to avoid a situation where the project informally adapts around the deviation before the engineering consequence is understood.

Not every discrepancy needs the same formal process. A minor drafting typo corrected before issue is different from a field-installed deviation affecting a safety-critical design. The organization should define thresholds for formal nonconformance documentation while preserving enough records to support learning and traceability.

5.2 Disposition is a technical decision and authority must match the consequence

Common disposition pathways include correction to conform, rework, repair, replacement, rejection, use-as-is acceptance, or an approved design change. The terminology varies across industries, but the important distinction is authority. A field representative can often identify and document a deviation. Acceptance of a deviation that changes the engineering basis may require evaluation by the responsible engineer or another appropriately qualified authority.

Use-as-is acceptance should not be shorthand for convenience. It should document why the nonconforming condition still satisfies the relevant requirements or why a controlled change to the requirement is appropriate. The analysis may need revised calculations, testing, evaluation of margins, interface review, or owner/regulatory approval. If acceptance changes the design basis, the corresponding documents should be updated.

A repair disposition should define what work is required and how conformity will be verified afterward. A replacement decision may be technically simple but still require configuration updates. Quality is preserved when the disposition record connects the observed condition to the engineering evaluation, authority, final action, and closure evidence.

5.3 Correction and corrective action solve different problems

Correction fixes the immediate nonconforming condition. Corrective action addresses the cause so the problem is less likely to recur. If a drawing contains the wrong dimension, the correction may be to revise and reissue the drawing. If the wrong dimension occurred because two disciplines exchange uncontrolled spreadsheets and the same type of mismatch has happened several times, corrective action may require a controlled interface-data process.

Organizations sometimes overreact by requiring formal root-cause analysis for every small error. This creates paperwork that staff learn to treat mechanically. Others underreact by repeatedly correcting the same symptoms without examining why they recur. The appropriate response is proportional to significance, recurrence, detectability, and systemic potential.

Corrective action should target a controllable cause. “Engineer failed to pay attention” is rarely sufficient because it does not identify how the system will reduce recurrence. Stronger analysis asks what information, process, workload, review control, interface, training, tool, or management condition allowed the error to escape. Human error can be part of the cause, but the quality system should look for the conditions that made the error likely or hard to detect.

TERM	MEANING	ILLUSTRATIVE RESPONSE
Nonconformance	A specified requirement was not met	Identify an undersized member in the issued drawing
Correction	Fix the affected item	Revise the drawing and evaluate affected installed work
Corrective action	Address the cause or control weakness	Require calculation-to-drawing back-check and verify its effectiveness

Figure 12. Nonconformance identifies the unmet requirement, correction fixes the affected item, and corrective action reduces recurrence.

5.4 Cause analysis should be proportional and evidence based

Simple recurring-error analysis can be highly effective when it moves beyond labels. Ask what happened, what requirement was not met, where the error entered the process, why the existing controls did not prevent it, and why the existing checks did not detect it before release. The answers may lead to deeper questions about unclear criteria, poor interfaces, overloaded reviewers, inadequate independence, tool configuration, training, or incentive structures.

Methods such as five-whys analysis, cause-and-effect diagrams, fault trees, barrier analysis, event-and-causal-factor analysis, or formal root-cause methods can support the investigation. The method should match the consequence. The important feature is evidence. Do not force a predetermined cause to fit the method.

Cause analysis should distinguish occurrence controls from detection controls. A design-basis checklist may prevent omission of a required criterion. An independent review may detect the omission before release. If both fail, the organization should understand why. Improving only the final check may leave an inefficient process that continues producing defects; improving only the originating process may overlook weaknesses in independent detection.

5.5 Trend quality data to find recurring system weaknesses

Individual nonconformances provide local information. Trends can reveal systemic problems. Repeated interface comments, late design changes, document-status errors, calculation revisions after issue, reopened review comments, field RFIs caused by ambiguity, or recurring use of superseded documents can indicate where the quality process needs improvement.

Metrics should support learning rather than create scorekeeping behavior that discourages reporting. A team judged only by the number of nonconformances may hide issues or reclassify them. A reviewer measured only by comment count may generate low-value comments. Useful metrics combine context: issue type, significance, phase discovered, recurrence, root cause, rework, closure time, and whether the issue escaped to downstream users.

The strongest improvement actions feed lessons back into the engineering process. Update standards, checklists, templates, training, automation, review criteria, interface registers, design-basis requirements, or resource planning. The goal is not a perfect historical record of mistakes. It is a better future design process.

5.6 Management of deviations and exceptions should preserve the design basis

Engineering organizations sometimes intentionally deviate from a standard practice because the project context justifies a different approach. A deviation is not necessarily a nonconformance if it is approved through the process defined for exceptions. The quality concern is that the exception remain visible and technically justified.

A strong deviation record identifies the requirement or standard practice, proposed exception, technical basis, risks, compensating controls, approval authority, duration or applicability, and affected documents. Temporary deviations should include an expiration or closure condition. Permanent deviations should be incorporated into the controlled design basis so future users do not unknowingly return to the superseded requirement.

Accepted-as-is conditions should receive similar discipline. If the organization accepts a nonconforming condition because evaluation shows adequate performance, that basis should be retained with the final record. Years later, a maintenance engineer or modification designer should be able to understand that the condition was knowingly evaluated rather than simply overlooked.

Applied scenario 4 - Repeated late interface conflicts

Three consecutive projects discover major electrical-mechanical coordination conflicts during final review. Each project team fixes the drawings before issue, and the issue is closed as a successful quality-control catch. No formal corrective action is opened because the defects did not reach construction.

The detection control is working, but the process is repeatedly creating preventable defects late in design. The organization is paying rework cost and depending on final review as the primary defense. The recurrence suggests a systemic interface-control weakness.

Trend the issues, identify where load and equipment data should be exchanged, define controlled interface milestones, establish ownership and status for shared inputs, and verify that the next project uses the improved process. Continue final review as a detection barrier, but do not treat repeated late catches as an acceptable steady state.

PRACTICE PRINCIPLE A quality system should learn from near misses and late catches, not only from failures that escape to the field.

5.7 Root-cause tools are useful only when they improve the explanation

Formal root-cause tools can help teams organize evidence, but the method should not become the objective. Five-whys questioning is useful when the causal chain is relatively simple. Cause-and-effect diagrams help explore categories such as people, process, tools, environment, information, and management. Barrier analysis asks which preventive and detective controls should have stopped the event and why they failed. Fault trees and event trees can support more complex technical systems.

The investigation should avoid hindsight bias. After an error is known, the correct answer can seem obvious. The team should reconstruct what information was reasonably available at the time, what cues were present, what assumptions were normal, and how workload or workflow affected the decision. This produces more useful corrective actions than simply identifying the person closest to the error.

Causal statements should be specific enough to support action. “Procedure not followed” prompts the next question: why? Was the procedure unclear, unavailable, impractical, unfamiliar, bypassed by schedule pressure, or poorly enforced? “Training issue” should identify what knowledge was missing and why the existing competence system did not detect the gap.

A strong cause analysis often identifies multiple contributing factors. Corrective action can then target the combination of conditions rather than searching for a single root cause when the evidence does not support one.

5.8 Corrective action should include verification of effectiveness

Completing an action item is not the same as demonstrating that recurrence risk was reduced. A corrective action should define how effectiveness will be verified. If the action updates a checklist, review several future projects to determine whether the checklist is used and whether the targeted omission declines. If the action changes an interface process, audit whether required data are exchanged at the planned milestone. If training is provided, verify competence through work review rather than attendance alone.

Effectiveness measures should be practical and proportionate. Some actions can be verified immediately by testing the revised template or workflow. Others require time to observe recurrence. The closure record should identify the evidence supporting the conclusion that the action worked.

If the problem recurs, the organization should not automatically add another layer of paperwork. Reassess whether the original cause was correct, whether the action was implemented, and whether new conditions changed the process. Repeated ineffective corrective actions can indicate that the organization is treating symptoms or selecting actions that are easy to document rather than effective.

5.9 Lessons learned should be converted into reusable engineering knowledge

Project teams often hold lessons-learned meetings at closeout, but the value is limited if the lessons remain in a report that future teams never read. High-value lessons should be transferred into the tools and processes engineers actually use: design standards, checklists, calculation templates, training modules, interface registers, model libraries, example details, review criteria, or risk registers.

Lessons should distinguish context. A solution that worked on one project may not be universally appropriate. Record the condition, consequence, and applicability so future users understand when the lesson matters. Technical lessons are especially valuable when they explain a failure mode that standard procedures do not make obvious.

Organizations can also use design reviews to bring forward relevant historical lessons. A reviewer should ask whether similar systems or details have experienced problems. This creates institutional memory without requiring every engineer to remember every past project.

Positive lessons matter too. A coordination method, automated check, constructability review, or early stakeholder workshop that prevented rework should be captured so successful controls become repeatable.

5.10 Quality metrics should support management decisions, not distort behavior

Quality metrics can help leaders allocate resources and identify process weaknesses. Useful measures may include review comments by type and phase, late design changes, reopened comments, nonconformances, field RFIs caused by design ambiguity, revision frequency, percentage of changes with complete impact analysis, repeated interface issues, or time required to close significant technical comments.

Metrics need interpretation. A team with more documented issues may have a stronger reporting culture than a team with artificially low numbers. A rise in review comments can reflect worse design quality, deeper review, or both. Management should combine quantitative trends with qualitative review of significant events.

Avoid targets that encourage underreporting. “Zero design errors” can be a long-term aspiration but a poor performance measure if staff fear that documenting a mistake will harm evaluations. Better measures examine when issues are detected and whether recurrence declines. Catching an error in concept review is generally better than discovering it in the field, even though both count as an issue.

Management review should convert data into decisions about staffing, training, tools, procedures, and review depth. Quality metrics are useful when they change resource allocation or process design, not when they exist only for dashboards.

5.11 Latent conditions deserve attention even when no failure occurred

A project may finish successfully despite weak controls because experienced people compensated informally. The absence of failure does not prove that the process was robust. Near misses, late catches, heroic rework, and undocumented workarounds reveal latent conditions that can cause future problems under different staffing or schedule conditions.

For example, if a senior engineer repeatedly discovers missing criteria through personal memory, the project may succeed while the requirements process remains weak. If field staff routinely call the designer directly to clarify ambiguous details, problems may be solved without creating records, leaving

future teams unable to learn. If one reviewer works nights before every milestone to reconcile discipline conflicts, the process depends on exceptional effort rather than planned control.

Quality improvement should identify these compensating behaviors and decide which need to become formal controls. The objective is not to eliminate professional initiative. It is to avoid building a system that works only when the same highly experienced individuals are present.

5.12 Mini-case analysis - three different causes that can produce the same visible error

Visible problem: a released drawing contains an incorrect equipment load. Case A begins with a vendor data sheet that changed after design. The mechanical team received the new data, but the structural team did not. The root issue is interface and change communication. Corrective action should improve controlled data exchange and affected-discipline notification.

Case B begins with the correct load reaching the structural engineer. The calculation was updated, but the drafting file was not. The reviewer checked the calculation but did not back-check the drawing. The root issue is change reconciliation and review closure. Corrective action should link approved calculation changes to affected deliverables and require back-check before release.

Case C begins with both calculation and drawing using the same wrong load because the project criteria package contained an outdated value. Every discipline and reviewer worked from the same controlled but incorrect source. The root issue is requirements source control. Adding more drawing checking would not solve the problem. Corrective action should strengthen how authoritative vendor data are established and revised.

The three cases demonstrate why visible defects do not identify the correct corrective action. A quality investigation should trace where the wrong condition entered the process and which controls failed to prevent or detect it. Otherwise organizations accumulate additional final checks while upstream causes remain unchanged.

5.13 Management review should connect quality trends to resources and project planning

Quality problems often reflect resource decisions. If independent reviews are repeatedly compressed into the final day of a milestone, the solution may require schedule planning rather than another checklist. If the same specialist is overloaded across several projects, reviewer capacity becomes a quality risk. If document-control errors increase after adopting a new platform, training and workflow configuration may need attention.

Management review should therefore examine quality indicators alongside staffing, schedule, project complexity, and organizational change. Significant nonconformances, recurring late changes, high rework, audit findings, and unresolved corrective actions can inform where management needs to invest.

Leaders should also review whether quality controls remain proportionate. Processes can become bloated over time as every past issue adds a new form. Periodic simplification can improve compliance and focus attention on meaningful controls. Retire steps that no longer reduce risk, automate administrative tasks, and preserve the technical controls that protect engineering decisions.

The objective of management review is an adaptive quality system. Engineering practice, software, codes, project delivery methods, and organizational structure change. Quality procedures should evolve with them while maintaining the core principles of clear requirements, competent review, controlled change, and defensible records.

5.14 Practical nonconformance and corrective-action review

For a nonconformance, begin with control of the affected condition. Identify the unmet requirement, extent, status, and immediate consequence. Determine whether further use or work should be stopped, segregated, limited, or otherwise controlled while the technical disposition is developed. Preserve evidence before the condition is altered.

The disposition should identify the qualified technical authority, evaluation, final action, and verification of closure. If the condition is accepted as-is or repaired, retain the basis demonstrating why the result is

acceptable. If the disposition changes the design basis, reconcile the change into controlled project records.

Then decide whether corrective action is warranted. Consider significance, recurrence, systemic potential, and detectability. Trace where the problem entered the process and which barriers failed. Select actions that address controllable causes, and define how effectiveness will be verified on future work. A corrective-action process is successful when recurrence risk changes, not merely when the action list is complete.

- What specific requirement was not met?
- What immediate control is needed to prevent unintended use or additional consequence?
- Who has authority to disposition the condition?
- Does the disposition require design-basis or configuration updates?
- Is the issue isolated or part of a recurring pattern?
- Which preventive and detective controls failed?
- What evidence will demonstrate that corrective action reduced recurrence?

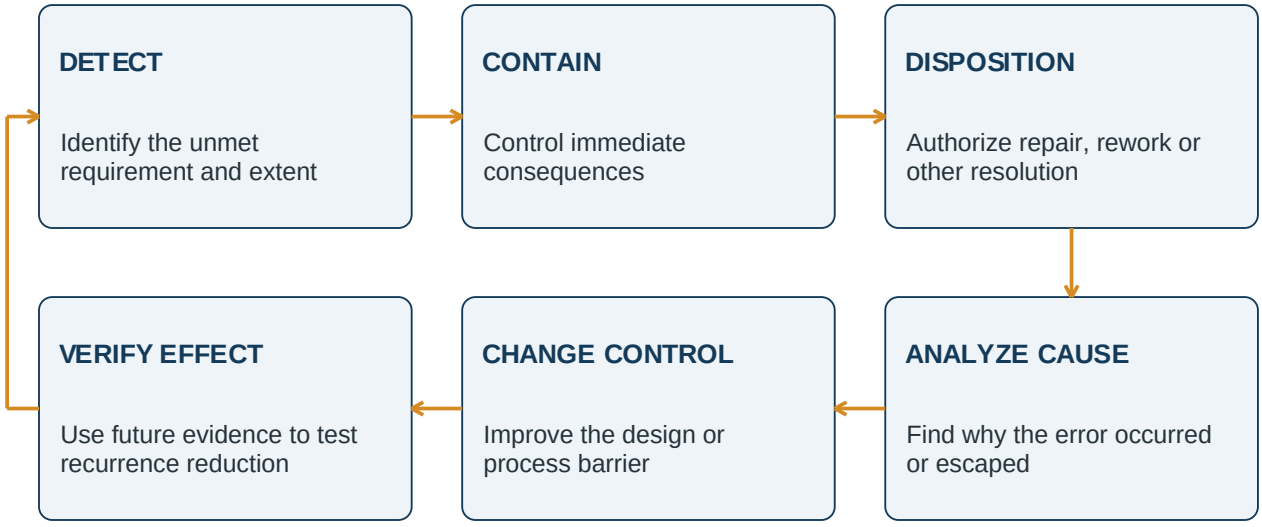

Figure 13. Corrective action is complete only when effectiveness is checked and the learning is fed back into future engineering practice.

Module 5 key point. Correct the immediate problem, but use recurrence and significance to decide when the engineering process itself must change. Quality improvement is strongest when lessons become controlled practices rather than informal reminders.

MODULE 6

Applied Quality Scenarios, Records & Professional Accountability

Planned instructional time: 35 minutes. Study the examples, decision points, diagrams, and scenarios as part of the instructional content.

6.1 Quality records should reconstruct the decision, not merely prove that forms were completed

A defensible engineering record allows a qualified person to reconstruct the technical decision after the people who made it are unavailable. It identifies the governing requirement, design basis, work product, reviewer, significant comments, revisions, approved changes, nonconformance dispositions, and final configuration. The record should demonstrate both technical reasoning and control of the document state.

Records can include calculations, drawings, specifications, model files, review checklists, comment logs, meeting decisions, design criteria, interface registers, approved submittals, change records, nonconformance reports, field instructions, test results, correspondence, and final record documents. The organization should define which are controlled records and how long they are retained.

Quantity is not the objective. Thousands of emails do not automatically create a good engineering record. The evidence should be organized enough that the important decision can be found and understood. A concise approved design memorandum can be more useful than a long chain of informal messages. The final record should also preserve status: which document was approved, which was superseded, and which configuration was actually implemented.

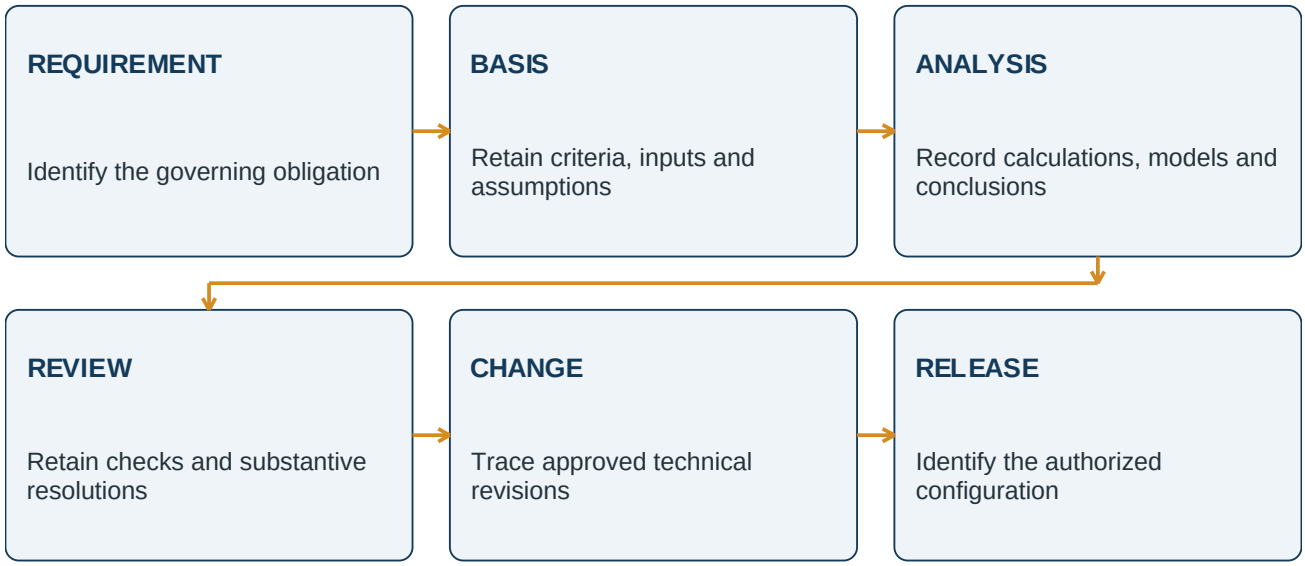

Figure 14. A defensible engineering record links the requirement, work product, review, issue, decision, technical authority, and final state.

6.2 Professional responsibility remains with qualified people even when the workflow is automated

Electronic quality-management systems can route reviews, enforce signatures, compare revisions, prevent release without required approvals, and preserve audit trails. These controls are valuable, but they do not make the technical decision. A workflow can confirm that a reviewer clicked “approve”; it cannot determine whether the reviewer was competent, whether the review was adequate, or whether the technical basis was sound.

Automation should therefore support professional judgment rather than obscure it. Required fields should capture meaningful information. Checklists should be tailored to actual failure modes. Digital approvals should identify the role being performed. Review dashboards should distinguish overdue technical comments from minor administrative items. Automated rule checks and model validation can detect many issues, but qualified engineers should understand the limits of the tools.

The same principle applies to artificial intelligence and automated design assistance. These tools can accelerate drafting, search, calculation, code checking, and document comparison, but they can introduce errors, hallucinated criteria, hidden assumptions, or false confidence. The engineer responsible for the work should verify inputs, methods, outputs, and applicability before relying on automated results.

6.3 Release is both a technical decision and a configuration-control event

Before engineering work is released, the organization should determine whether the design basis is current, technical work is complete, required checks are documented, interfaces are resolved, comments are closed, changes are incorporated, documents have correct revision and status, responsible-charge requirements are met, and the record is complete enough for the intended use.

Release criteria should match the purpose. A preliminary package issued for pricing should not be mistaken for construction-ready design. A drawing issued for permit may not be authorized for procurement. A design-development model may contain unresolved assumptions. Clear status allows work to proceed for a limited purpose without creating false confidence about maturity.

Once released, the design becomes a configuration baseline for downstream users. That is why late edits outside the controlled process are dangerous. A seemingly minor edit made after approval can invalidate review evidence. Any post-approval technical change should be evaluated to determine whether re-review or re-approval is required.

RELEASE CHECK	EVIDENCE
Design basis	Applicable requirements, criteria and important assumptions identified
Technical work	Analyses, models and deliverables complete for intended use
Interfaces	Affected disciplines coordinated and shared inputs confirmed
Review closure	Substantive comments implemented and back-checked
Changes	Approved changes incorporated in every affected record
Document status	Correct revision and distribution path identified
Professional responsibility	Required authority, signatures and seals appropriate to role
Record	Decision trail and implemented configuration retained

Figure 15. Release readiness confirms design basis, technical work, interfaces, review closure, change incorporation, document status, professional responsibility, and retained evidence.

6.4 Applied scenario - field revision not incorporated into final records

During construction, an underground conflict requires relocation of a conduit bank. The engineer reviews a field sketch, confirms that clearances and electrical design remain acceptable, and authorizes the change by email. Construction proceeds successfully. At closeout, the original issued drawing is included in the record set because the field sketch was stored in the contractor's RFI system and never entered into the design document-control process.

The immediate technical decision may have been sound, but the final record is wrong. Future excavation, maintenance, or expansion work could rely on the original conduit location. The quality issue is not that the engineer used email; it is that the approved change was never reconciled into the controlled record.

The appropriate response is to identify the approved field decision, verify the implemented condition, revise the final record drawing, preserve the authorization basis, and improve the closeout/change process so field-authorized technical changes are captured automatically. The case demonstrates why verification of final configuration is part of quality assurance rather than clerical closeout.

6.5 Applied scenario - design change without downstream interface review

A mechanical team changes a large air-handling unit to resolve procurement availability. The new unit has similar airflow and power but a different operating weight and access envelope. Mechanical documents are updated and approved. Structural support calculations, electrical demand schedules, architectural access panels, and controls point lists are not formally reviewed because the change is considered "equal performance."

The error is the assumption that similar primary function means limited system impact. The new equipment creates interfaces beyond airflow. The change-control process should identify affected

disciplines and require affirmative impact review. Some disciplines may conclude that no change is needed, but that conclusion should be documented.

A disciplined response would screen the change for configuration significance, route the revised equipment data to affected disciplines, update supporting calculations and documents where necessary, verify constructability and maintenance access, and reconcile the final record. The scenario illustrates the value of an affected-document and interface checklist.

6.6 Applied scenario - incomplete design-basis documentation

A senior engineer develops a successful foundation design using experience with local soil conditions, prior projects, and conservative assumptions. The calculations are correct and the design performs well. Years later, another engineer must modify the structure, but the original calculation does not identify the assumed groundwater elevation, settlement tolerance, or why a particular soil parameter was selected. The geotechnical report contains several possible values.

The original design may have been technically adequate, but the record does not preserve enough design basis for efficient future engineering. The modifying engineer must reconstruct assumptions, increasing uncertainty and cost. If the original engineer is unavailable, the organization loses institutional knowledge.

Quality assurance should require important assumptions and criteria to be documented contemporaneously. The lesson is not to create unnecessary narrative around every routine choice. It is to record the decisions that materially affect future verification, modification, or interpretation.

6.7 Applied scenario - reviewer comments closed without implementation

A final review identifies six material drawing comments. The originator agrees with all six and enters responses describing the required changes. The reviewer checks the response log and approves closure. During issue, only four drawing changes are incorporated because two comments were missed in drafting. The review log is complete, but the released design does not reflect the approved disposition.

This failure separates response approval from implementation verification. For material comments, closure should include confirmation that the affected work product was actually revised. Some systems require the reviewer to inspect the final changed document; others use a controlled markup and back-check process. The specific method can vary, but the technical loop must close.

The organization should evaluate whether the error was isolated or whether the workflow routinely closes comments before revised documents exist. If systemic, the corrective action may be to change status sequencing so a comment cannot reach “closed” until the revised deliverable is available for back-check.

6.8 Applied scenario - field acceptance exceeds delegated authority

An inspector observes that installed anchor spacing differs from the approved drawing. The contractor states that the alternate spacing has worked on other projects and asks whether the inspector can “accept it in the field” to avoid delay. The inspector believes the difference is minor and documents the condition as acceptable in the daily report.

The quality problem is not whether the alternate spacing is ultimately safe. The issue is that the disposition requires engineering evaluation and authority beyond routine inspection. An observation role does not automatically include authority to revise the design basis. Informal acceptance can convert a technical deviation into an undocumented configuration change.

The correct response is to document the condition, preserve status as required, route the deviation for engineering evaluation, obtain an authorized disposition, communicate the decision, verify the completed work, and update final records if the approved configuration changes. Clear role boundaries protect both the inspector and the project.

6.9 Applied scenario - urgent release under schedule pressure

A permit deadline is approaching and an owner asks the engineer to issue a package before the planned interdisciplinary review is complete. The discipline designs are individually checked, but several shared interfaces remain open. The owner accepts the schedule risk and says the team can “clean it up after permit.”

Schedule authority does not erase technical responsibility. The engineer should determine the actual intended use of the package, the significance of unresolved interfaces, and whether issuing the documents could create reliance beyond their maturity. If limited release is appropriate, the status should be explicit, unresolved items identified, and downstream use restricted. If the open issues could affect public safety, code compliance, or fundamental design adequacy, release may be inappropriate regardless of schedule pressure.

The quality system should define release gates and exception authority so the decision is not improvised during a deadline. A controlled exception can be legitimate; an ambiguous “issue now, fix later” culture undermines both review evidence and professional accountability.

6.10 Practical release checklist for engineering work

- Confirm the governing requirements, codes, owner criteria, and design basis are current and traceable.
- Confirm material assumptions, constraints, and open items are resolved or explicitly controlled.
- Confirm calculations, models, drawings, specifications, and reports are internally consistent.
- Confirm discipline and interdisciplinary interfaces have been reviewed at the appropriate depth.
- Confirm independent checking or verification required by project risk and procedure is complete.
- Confirm review comments are technically dispositioned, implemented, back-checked where necessary, and closed.
- Confirm all approved changes are incorporated into affected work products and configuration records.
- Confirm document numbers, revisions, status, approvals, and distribution information are correct.
- Confirm responsible-charge, sealing, or technical-approval requirements are satisfied where applicable.
- Confirm the retained record is sufficient to reconstruct the important design decisions and final configuration.

6.11 Integrated case study - requirements through final change reconciliation

Consider a municipal operations building with emergency power, fuel storage, drainage, communications, and mechanical systems. Early in design, the owner requires 72-hour continuity of operations during a major storm. The requirement affects generator capacity, fuel volume, flood elevation, ventilation, electrical distribution, structural support, controls, site access, and maintenance planning. The project team records the continuity requirement in the design basis and assigns cross-discipline interfaces.

During preliminary review, the electrical engineer identifies that the proposed generator capacity is based on preliminary mechanical loads. The mechanical team marks several loads as assumptions and establishes a resolution date before equipment procurement. The structural team identifies generator and fuel-tank reactions as controlled inputs. Civil establishes the required flood elevation and site access criteria. The design review log captures these open interfaces rather than treating them as informal coordination.

At detailed design, the generator vendor proposes an alternate model with improved fuel efficiency but different footprint, weight, ventilation, and controls. Because the vendor data are configuration-significant, the change enters the change-control process. Mechanical reviews ventilation and heat rejection, structural reviews foundation reactions, electrical confirms capacity and fault contribution, controls reviews communication points, architectural confirms access, and civil confirms the revised tank arrangement does not conflict with drainage. Each discipline records impact or no impact.

Construction later discovers an underground utility that conflicts with the fuel line. The field team documents the location and proposed reroute. The responsible engineers evaluate clearance, pressure drop, structural penetrations, access, and drainage implications. The approved field change is incorporated into the controlled drawings and final record set. Closeout verification confirms the installed generator and fuel system support the 72-hour requirement.

This integrated example shows how quality controls connect rather than operate as isolated forms. Requirements traceability establishes the objective. Interface control makes dependencies visible. Design review challenges preliminary assumptions. Vendor change control coordinates new information. Field change control preserves technical authority. Configuration reconciliation makes the final record match the implemented system. At every phase, the quality evidence answers what requirement governed, what changed, who reviewed it, and whether the final configuration still satisfies the original need.

6.12 Professional responsibility and quality-system responsibility are complementary

An organization can define excellent procedures, but licensed professionals remain responsible for the engineering judgments assigned to them. Conversely, individual professional responsibility does not mean the engineer should work without organizational controls. Quality systems support competent professional practice by improving information, independence, traceability, and review.

The engineer should understand the meaning of each signature or digital approval. Signing as originator is different from signing as checker, discipline reviewer, engineer of record, or project approver. Organizations should not use professional seals or approval signatures as generic administrative milestones. The role represented by the signature should match the actual technical responsibility and jurisdictional requirements.

Responsible charge is particularly important when work is prepared by teams. A professional engineer may rely on qualified staff, specialists, and independent reviewers, but should maintain the level of direction, control, knowledge, and review required by applicable law and professional practice. A quality system can document the process, but it cannot manufacture responsible charge after the fact.

When a quality procedure and professional judgment appear to conflict, the engineer should escalate the issue rather than quietly bypass either. Procedures can be wrong or inapplicable; professional judgment can also be mistaken. A documented technical exception reviewed by appropriate authority is stronger than an undocumented work-around.

6.13 Scenario set - rapid decisions under competing project pressures

Scenario A: A fabricator identifies that an approved detail is difficult to construct and proposes a simpler connection. The proposed detail appears structurally adequate but changes load transfer and erection sequence. The project is behind schedule. Quality response: treat the proposal as a technical change, obtain the responsible engineer’s evaluation, review erection and interface consequences, revise controlled documents if approved, and do not let schedule transform a proposal into implied acceptance.

Scenario B: A reviewer discovers that a model uses a slightly different material property than the calculation narrative. The difference does not control the final design margin. Quality response: determine whether the discrepancy is material, correct the inconsistent record, and decide whether re-analysis is needed. Avoid both extremes: ignoring the mismatch because it seems small, or performing disproportionate rework with no technical benefit.

Scenario C: An owner asks to issue a drawing for procurement before final interdisciplinary coordination. Quality response: define the limited release purpose, identify unresolved interfaces, restrict downstream use, and determine whether the procurement item can be safely fixed while other design details continue. A controlled preliminary release can be legitimate when status and limits are explicit.

Scenario D: A field inspector reports repeated anchor-location deviations that have each been individually accepted after engineering evaluation. Quality response: continue evaluating the current condition, but also trend the recurrence. The pattern may indicate layout, fabrication, template, survey, or communication problems that warrant corrective action.

Scenario E: A senior engineer informally checks a junior engineer's calculation during a screen-sharing session and says it looks good. The project procedure requires an independent check record. Quality response: the technical interaction is valuable, but the required evidence still needs to be completed. The reviewer should document the scope of the check, important points verified, comments, and approval for the correct revision rather than recreating a meaningless form later.

6.14 What a reviewer should see in the engineering record

For a quality-controlled engineering package, a qualified technical reviewer should be able to identify the governing design basis without searching through unstructured correspondence. The important calculations and models should identify their inputs and revisions. Drawings and specifications should be internally coordinated. Review records should show who checked the work, what significant issues were raised, and how they were resolved. Changes after review should be visible and re-reviewed to the extent appropriate.

The reviewer should also be able to follow important changes. If a major equipment load changed, the record should show the new source data, affected disciplines, revised analyses, approval, and updated final documents. If a nonconforming field condition was accepted, the record should show the engineering basis and final configuration. If a requirement was intentionally waived or changed, the authority and rationale should be visible.

This standard is not perfection. Engineering records vary by project size and delivery method. The test is whether the evidence is sufficient to understand the technical decisions that materially affected safety, performance, compliance, and final configuration. A well-controlled record lets a future professional continue engineering work without first reconstructing the project from personal memory.

6.15 Personal practice checklist for the individual engineer

- Before starting work, identify the governing criteria and verify that important inputs are current.
- Record assumptions that materially affect the design and identify how they will be resolved.
- Ask what other disciplines or downstream users depend on the information you are creating.
- Use independent checking appropriate to consequence; do not confuse familiarity with verification.
- When reviewing, focus first on technical significance, interfaces, and design basis before formatting.
- Do not close a comment until the technical resolution is reflected in the affected work product.
- When information changes, identify every calculation, drawing, model, specification, and interface that may be affected.
- Distinguish technical approval from contractual, administrative, or schedule authorization.
- Preserve the engineering basis for accepted deviations and field changes.
- Before release, confirm the work product, review evidence, revision status, and professional responsibility all refer to the same technical configuration.
- After an error or near miss, ask what process change would make recurrence less likely rather than relying only on personal vigilance.

The applied cases integrate requirements control, independent review, document status, change evaluation, nonconformance and final-record principles. Follow the governing requirement, the control that should have acted, the role with decision authority, the affected records and the evidence of closure.

6.16 Workshop synthesis - a repeatable decision sequence

Across these cases, the technical disciplines differ but the quality logic is consistent. First identify the governing requirement and current approved configuration. Second determine whether the issue is a defect, a new input, a proposed change, or a documentation mismatch. Third control status so affected parties do not proceed from information whose validity is uncertain. Fourth route the issue to qualified technical authority and all affected interfaces. Fifth perform impact analysis at a depth proportionate to consequence. Sixth document the decision, implementation, verification, and final configuration.

This sequence helps engineers avoid two common mistakes. The first is treating every issue as a local technical problem even when it changes shared configuration. The second is treating quality as

paperwork after the engineering decision has already been made. A strong quality process operates at the moment the decision occurs and makes the technical evidence part of normal work.

For personal practice, ask five questions whenever important engineering information changes: What requirement does this affect? Who else depends on this information? What prior review or verification becomes invalid? Who has authority to approve the changed condition? What record must be updated so the next engineer sees the same technical state? Those questions capture the essence of design quality, review, and change control in a form that can be applied across engineering disciplines.

6.17 Final release and record audit

Before final release, select a sample of significant design decisions and attempt to reconstruct them using the project record. Can the reviewer find the requirement, calculation or analysis, review evidence, design change, approval, and final deliverable without relying on the memory of the original team? This simple exercise is an effective test of record quality.

Confirm that the release status is appropriate for the intended use and that required signatures or seals represent the actual professional role. Review the revision index and important changes since the previous release. Confirm that superseded files cannot be mistaken for current work and that final distribution reaches the parties who rely on the design.

For closeout, reconcile approved field changes, substitutions, nonconformance dispositions, surveys, tests, and commissioning information. The final record should represent the best controlled understanding of the implemented configuration while stating any appropriate limitations. A strong closeout record reduces future engineering uncertainty and preserves the value of the design work long after the original project team has moved on.

6.18 Why quality records matter after the project is over

Engineering quality records often become more valuable with time. Future engineers may need to modify the asset, investigate a failure, respond to a regulatory question, replace equipment, assess remaining life, or determine whether a proposed change is compatible with the original design. If the record preserves only final drawings but not the assumptions and significant design decisions, future work begins with unnecessary uncertainty.

A useful lifecycle record does not require permanent retention of every working note. It preserves the items that explain the approved technical state: governing criteria, key calculations and models, significant review evidence, approved changes, accepted deviations, final configuration, and important limitations. The retention period and exact record set should follow applicable law, contract, owner requirements, and organizational policy.

Record quality also supports organizational learning. When a later project encounters a similar problem, engineers can review not only the final solution but why alternatives were rejected, what assumptions controlled, and what field experience followed. This turns prior engineering work into reusable knowledge rather than a collection of static files. Defensible records therefore serve both accountability and future technical efficiency.

6.19 Final reflection - quality is a technical habit

The strongest engineering quality systems eventually become habits rather than separate administrative events. Engineers ask for the current criteria before beginning analysis, identify interfaces while developing the design, seek independent challenge before committing to a solution, document important assumptions while they are fresh, and treat post-approval revisions as controlled technical decisions. These habits reduce the amount of corrective paperwork needed later because the engineering process itself generates reliable evidence.

Quality is therefore both organizational and personal. The organization supplies procedures, tools, independent review, document control, and escalation paths. The individual engineer supplies competence, attention, professional skepticism, truthful communication, and willingness to stop when

the technical basis is unclear. When both levels work together, quality assurance supports efficient project delivery rather than competing with it.

A final practical test is simple: hand the approved package to a qualified engineer who was not on the project and ask whether that person can identify the governing criteria, understand the major assumptions, trace important calculations to deliverables, determine what changed after review, and identify the final approved configuration. Where that reconstruction is difficult, the quality record may need improvement even if the project itself was successful. The exercise turns abstract recordkeeping into a concrete engineering-usability test.

That independent reconstruction test also reveals where project knowledge still depends on individual memory rather than controlled engineering information and durable review evidence.

Course summary

Module 6 key point. The final evidence of engineering quality is not the absence of paperwork errors. It is a controlled technical record showing what requirement governed, what was designed, what was reviewed, what changed, who had authority, and what was ultimately released or implemented.

Engineering quality assurance is the structured discipline that keeps professional judgment connected to reliable information and controlled technical decisions. Quality begins with requirements and design basis, continues through checking and interdisciplinary review, and remains active after issue through document control, configuration management, change evaluation, nonconformance disposition, and final record reconciliation.

The most important controls are mutually reinforcing. Requirements traceability protects against designing to the wrong criteria. Independent checking challenges hidden assumptions and execution errors. Interdisciplinary review addresses boundaries that no single discipline owns. Comment closure preserves the technical resolution. Configuration control prevents superseded information from being mistaken for current design. Change control ensures local decisions do not create unexamined downstream consequences. Corrective action turns recurring defects into process improvement. Quality records preserve professional accountability over the life of the project or asset.

A strong system does not eliminate engineering judgment or attempt to replace competence with forms. It creates the conditions in which qualified judgment is more likely to be based on current information, independently reviewed, correctly communicated, and defensible after the project team has moved on.

Principal references

These public source records support the quality-management, configuration-management, systems-engineering and professional-practice principles used in this course. The course does not reproduce proprietary standards or replace the governing project requirements. Verify the editions incorporated into the applicable contract, owner procedure or jurisdictional requirement.

- ISO 9001:2026, Quality management systems — Requirements. Current published edition; replaces ISO 9001:2015 and its 2024 amendment. Referenced for quality-management principles. Open source
- ISO 10007:2017, Quality management — Guidelines for configuration management. Published edition; referenced for configuration identification, change control, status accounting and audit principles. Open source
- NASA Systems Engineering Handbook, NASA/SP-2016-6105 Rev. 2. Requirements, technical reviews, verification/validation, configuration management and technical-data management. Open source
- NCEES Model Rules, revised August 2025, Section 240.15, Rules of Professional Conduct. A model framework; the applicable jurisdiction's adopted requirements govern professional work. Open source
- NSPE Code of Ethics for Engineers. Competence, truthful communication, responsible practice and protection of public health, safety and welfare. Open source
- NBS Building Science Series 143, Investigation of the Kansas City Hyatt Regency Walkways Collapse (1982). Investigation findings and Figure 3.11, printed p. 28. Open source

Applicable discipline-specific codes, standards, owner criteria, contracts, regulatory requirements and organizational engineering procedures govern actual project practice when they impose more specific requirements.

Reference records checked November 15, 2026. ISO 9001:2026 was published September 16, 2026; project adoption and certification transition requirements must be checked separately.

Instructor

James Barber, P.E. — Florida Professional Engineer No. 75518. B.S. in Civil Engineering, University of Florida. His BuilderCE subject areas include civil and transportation engineering, engineering management, professional practice, design review and technical quality.

Study guide complete. Return to the BuilderCE lesson page and select Continue to begin the final examination. A score of at least 70% is required to pass. Retakes are permitted.