GALLANT PROJECT SOLUTIONS LLC
Service-Disabled Veteran-Owned Small Business (SDVOSB) · Veteran-Owned Small Business (VOSB)

Running Requirements

A Practical Primer for Design and Construction Projects

Greg Tuite, PE PMP
Principal · Gallant Project Solutions LLC
25+ Years · Healthcare · Hospitality · Theme Park · Defense · Commercial

May 2026 · First Edition · First in the Gallant PM Primer Series

Running Requirements: A Practical Primer for Design and Construction Projects

First Edition · May 2026 · First in the Gallant PM Primer Series

For permissions, licensing, or engagement inquiries:

gregory.tuite@gallantprojectsolutions.com

www.gallantprojectsolutions.com

Gallant Project Solutions LLC is an SBA-certified Service-Disabled Veteran-Owned Small Business (SDVOSB) and Veteran-Owned Small Business (VOSB).

The field examples in this primer are drawn from real project experience. Identifying details have been changed or omitted to protect confidentiality. All dollar figures, dates, and project names used in examples are illustrative unless otherwise stated.

About the Author

Greg Tuite, PE PMP is the founder and principal of Gallant Project Solutions LLC, an SBA-certified Service-Disabled Veteran-Owned Small Business (SDVOSB).

Greg brings more than 25 years of experience leading complex capital programs across award-winning theme park and immersive entertainment projects, hospitality and resort developments, complex mixed-use and institutional capital programs, and military facility construction. He has served as owner's representative, project manager, and requirements lead on projects ranging from special operations training facilities to large-scale guest experience venues.

His work spans the full project lifecycle — from early requirements definition and DRD development through commissioning and closeout — giving him a practitioner's view of where requirements management succeeds, where it fails, and why the discipline matters more on complex projects than anywhere else.

Greg holds a Professional Engineer (PE) license and Project Management Professional (PMP) certification. He served in the United States military and founded Gallant Project Solutions to bring enterprise-grade requirements intelligence to owners, designers, and constructors who need it most.

Gallant Project Solutions LLC provides requirements management services, owner's representation, and project delivery advisory to owners, A/E firms, and contractors on capital projects. The firm is headquartered in the United States and serves clients nationally.

Contact: gregory.tuite@gallantprojectsolutions.com · www.gallantprojectsolutions.com

Introduction

Most capital projects — buildings, facilities, infrastructure, attractions — start with excitement and optimism. Stakeholders know what they want. Designers are eager to create. Contractors are ready to build. But somewhere between kickoff and ribbon-cutting, the project gets harder than anyone expected.

Budgets creep. Schedules slip. Scope arguments emerge. Late in construction, someone discovers that a system does not meet an unstated expectation, or that a feature everyone assumed would be included was never actually designed. Fixing these problems after the fact is expensive, slow, and frustrating.

The root cause is usually not incompetence or bad intent. It is that requirements were never clearly defined, agreed upon, and tracked. Requirements are the explicit, testable descriptions of what the project must achieve — what it must do, how well it must perform, and under what constraints. When requirements are vague, scattered, or missing, every downstream decision becomes a guess. When they are clear and managed deliberately, the project has a fighting chance to deliver what was actually needed.

This guide is for project managers, designers, construction managers, and owner's representatives who work on capital projects in any industry: theme parks and entertainment, military and government facilities, healthcare, hospitality, commercial and industrial construction. It is a practical introduction to requirements management — the discipline of defining, documenting, tracking, and verifying what a project must achieve from concept through commissioning.

You will learn:

- What requirements are, why they matter, and what happens when they are ignored
- How to capture and document requirements in a Design Requirements Document (DRD)
- How to track and trace requirements through design and construction using a requirements log and Requirements Traceability Matrix (RTM)
- How to manage requirements as a living system — through change control, RFIs, submittals, and commissioning
- How to capture lessons learned and feed them into the next project

The methods in this guide are not proprietary or exotic. They come from aerospace, defense, and systems engineering disciplines, adapted for the reality of design and construction projects where teams are large, schedules are tight, and documentation is imperfect.

Tracking requirements has always been the right answer and the hard answer. On many past projects it felt too time-consuming to do well, because drawings, specifications, emails, and meeting notes all captured different slices of intent. Today, better collaboration tools — and increasingly, AI — mean you can capture and trace requirements with far less manual effort. This guide shows you the process; those tools simply make it easier and more productive to run.

This primer does not promise to eliminate risk or prevent all problems. It will not fix poor leadership, unrealistic budgets, or impossible schedules. What it will do is give you a structured way to define success, communicate expectations, and verify delivery. Used consistently, these practices reduce rework, protect budgets and schedules, and improve the quality of what you hand over to the owner or end user.

Let us start with the simplest question: What is a requirement?

Requirements, Constraints, and Assumptions

Project teams often confuse requirements with constraints or assumptions. Separating these concepts improves clarity and decision-making.

Term	Definition	Example
Requirement	A condition the project must satisfy and that can be verified.	"The facility shall open to the public no later than October 1, 2027."
Constraint	A limitation imposed on the project — regulatory, site, or contractual.	"The building height shall not exceed the zoning limit of 75 feet."
Assumption	A condition believed to be true for planning purposes, but that may change.	"Ride manufacturer delivery will occur within 24 weeks of purchase order."

When assumptions change, they often generate new requirements or schedule impacts. Explicitly documenting them helps the team react early rather than discovering problems late.

1. What Is a Requirement?

A requirement is a statement that describes something the project must achieve, perform, or comply with. It defines success in a way that can be tested, measured, or verified.

Requirements answer questions like:

- **What must this facility do?** (Functional requirements)
- **How well must it perform?** (Performance requirements)
- **What constraints must it satisfy?** (Regulatory, budget, schedule, physical)
- **What conditions must it operate under?** (Environmental, operational, maintainability)

What Makes a Good Requirement?

A good requirement is:

- **Clear** — No ambiguity about what is being asked.
- **Testable** — You can verify whether it was met.
- **Necessary** — It addresses a real need or constraint.
- **Feasible** — It can be achieved within project constraints.
- **Traceable** — You can link it to design decisions and verification activities.

Good and Bad Requirements

The difference between a vague requirement and a useful one is specificity. Vague requirements leave room for interpretation and disagreement. Specific requirements can be verified.

Vague or Untestable	Clear and Testable
"The facility should be user-friendly."	"90% of guests shall complete self-check-in in under 2 minutes during usability testing."
"Provide adequate lighting."	"Task lighting at work surfaces shall provide a minimum of 50 footcandles, measured at desk height."
"The system must be reliable."	"The ride system shall achieve 98% uptime during scheduled operating hours, measured monthly."
"Comply with all applicable codes."	"The facility shall comply with IBC 2021, NFPA 101 (2021 edition), and all local amendments adopted by [jurisdiction] as of [date]."
"Design for future expansion."	"The electrical service and distribution system shall be sized to support a 25% increase in connected load without replacing the main switchgear."

Categories of Requirements

Requirements come in several types. Most projects need at least some of each. The category prefix becomes the first part of the requirement ID in your log (FR-001, PR-042, etc.).

Functional Requirements (FR)

What the facility or system must do.

- "The training range shall accommodate live-fire exercises for up to 30 personnel simultaneously."
- "The attraction shall process a minimum of 1,800 guests per hour at theoretical capacity."
- "The facility shall provide secure storage for 500 weapons with individual access logging."

Performance Requirements (PR)

How well the system must perform.

- "HVAC systems shall maintain interior temperature between 68°F and 74°F under full occupancy."
- "Queue shade structures shall reduce direct solar exposure to 90% of queue area during peak sun angles (10 AM–4 PM, June–August)."
- "The facility shall achieve LEED Gold certification."

Schedule and Milestone Requirements (SMR)

When specific conditions must be achieved to support downstream activities. These requirements establish time-based constraints that directly drive the project schedule.

- "Ride system anchor bolts and embedded steel plates shall be installed and surveyed prior to foundation concrete placement to maintain ride alignment tolerances."
- "MRI imaging equipment shall be delivered and set in place before final wall enclosure of the imaging suite."
- "Primary electrical service shall be energized no later than 90 days prior to facility commissioning to support system testing."

Capturing these requirements early allows planners to build realistic schedules rather than discovering sequencing constraints during construction.

Budget and Cost Requirements (BCR)

Financial limits, cost targets, or value priorities for the project or specific systems. Treating budget expectations as explicit requirements helps ensure design decisions remain aligned with the owner's financial objectives.

- "Total project construction cost shall not exceed $65 million (2026 dollars), excluding owner-furnished equipment."
- "The exhibit lighting system shall be designed within a maximum installed cost of $1.2 million."
- "Operational staffing shall not exceed 12 FTEs per shift at steady-state operation."

Physical and Interface Requirements (PIR)

Constraints related to physical characteristics or interfaces with other systems, sites, or infrastructure.

- "The loading dock shall accommodate a 53-foot trailer with 14-foot vertical clearance."
- "Electrical service shall interface with existing 12.47 kV campus distribution at pole [ID], approximately 300 feet north of the building."
- "The building footprint shall not exceed 45,000 square feet due to site setback and easement constraints."

Regulatory and Compliance Requirements (RCR)

Requirements imposed by codes, standards, regulations, and Authority Having Jurisdiction (AHJ) conditions.

- "The facility shall comply with IBC 2021, NFPA 101, UFC 3-600-01, and all local amendments."
- "Accessibility shall meet or exceed ADA Standards for Accessible Design (2010, as amended)."
- "Fire alarm system shall be designed, installed, and commissioned per NFPA 72 and approved by the local fire marshal prior to occupancy permit issuance."

> **AHJ Conditions Are Requirements**
>
> Permit conditions, plan-check comments, and pre-application meeting notes often create binding requirements that affect cost, schedule, and design. Treat them as Regulatory and Compliance Requirements (RCR-XXX), with clear source documentation and owner assignments. Do not let AHJ conditions live only in email or permit files.

Sustainability and Environmental Requirements (SER)

Expectations around energy, water, waste, and environmental impact.

- "The facility shall achieve an Energy Use Intensity (EUI) of no more than 45 kBtu/sf/year."
- "Stormwater runoff shall not exceed pre-development rates for the 10-year and 100-year storm events."
- "Construction waste diversion shall achieve a minimum of 75% by weight."

Security and Life-Safety Requirements (SLR)

Requirements for protecting people and assets.

- "The facility shall support full evacuation of all occupants in under 4 minutes, verified by drill."
- "Access control shall restrict entry to secure areas using multi-factor authentication (card + biometric)."

Operational and Maintenance Requirements (OMR)

Expectations around how the facility will be used and maintained after commissioning.

- "HVAC filter access shall not require ladders or scaffolding; all filter racks shall be accessible from walkable surfaces."
- "The facility shall be operable with a steady-state staff of no more than 12 FTEs per shift."

- "Spare parts for all theatrical lighting fixtures shall be available from at least two suppliers with lead times under 30 days."

Why Requirements Matter

When requirements are clear and managed, projects are more predictable. When they are vague, missing, or ignored, projects suffer in predictable ways.

What Goes Wrong	The Cost
Rework and Late Changes	Design and construction proceed on assumptions. When assumptions prove wrong — often late in the process — work that was already “complete” must be redone. Rework is expensive, slow, and compounds schedule pressure.
Cost Growth and Waste	Unclear requirements lead to scope creep, gold-plating, and reactive problem-solving in the field. Each costs money. Requirements discipline enables deliberate tradeoffs: Must Have, Should Have, Nice to Have.
Schedule Delays	Late-breaking requirements, unanticipated permit conditions, or performance criteria that were never defined all create delays. Requirements-based practice makes constraints visible early.
Quality Shortfalls	If performance targets are never stated, there is no basis for verification. The result is often a facility that does not actually meet the owner’s operational needs.
Stakeholder Conflict	When expectations are not written down, people remember them differently. Disputes become arguments about memory and intent. Requirements create a shared record of what was agreed, by whom, and when.

> **The Earlier, The Cheaper**
>
> Requirements are the foundation of project control. The earlier you define and manage them, the less pain you experience later. A requirement change at concept design costs a fraction of the same change discovered during construction.

2. The Design Requirements Document (DRD)

The Design Requirements Document (DRD) is the single-source record of what your project must achieve. It is the baseline against which all design, construction, and commissioning activities are measured.

The DRD is not the same as drawings, specifications, or a scope-of-work narrative. Those documents describe how you will build. The DRD describes what you must deliver and why.

What Goes in a DRD?

A good DRD includes:

- **Project Overview** — Purpose and mission, high-level goals and success criteria, key stakeholders and their priorities.
- **Functional Requirements** — What the facility must do; operational scenarios and use cases.
- **Performance Requirements** — Quantitative targets (capacity, speed, efficiency, comfort, etc.); standards and benchmarks.
- **Physical and Interface Requirements** — Site constraints, footprint, adjacencies; utility and infrastructure interfaces.
- **Regulatory and Compliance Requirements** — Codes, standards, permits, AHJ conditions; accessibility, life-safety, and environmental regulations.
- **Sustainability and Environmental Requirements** — Energy, water, waste, and carbon targets; green building certifications or commitments.
- **Security and Life-Safety Requirements** — Threat considerations, access control, egress, fire and life-safety systems.
- **Operational and Maintenance Requirements** — Staffing models, maintainability, spare parts, training.
- **Constraints** — Budget ceiling, schedule milestones, and site or operational restrictions.
- **Assumptions and Risks** — What the team is assuming to be true, and known risks that could affect requirements.
- **Verification and Acceptance Criteria** — How each requirement will be tested, inspected, or demonstrated; who approves and what “done” looks like.
- **Governance and Change Control** — Who owns the DRD; how requirements changes are proposed, reviewed, and approved.

DRD Structure Template

The following outline can be adapted to any project type. Start minimal; add sections as the project grows.

Minimal DRD Outline

1. INTRODUCTION

   1.1 Project Overview and Purpose

When to Create the DRD

Start the DRD early — during pre-planning or concept design. Do not wait until design is underway. By then, many implicit requirements will already be baked into drawings, and it becomes much harder to course-correct.

Phase	DRD Activity
Concept / Pre-Design	Draft initial DRD: high-level requirements, assumptions, and constraints.
Schematic Design (SD)	Refine and expand DRD; resolve major TBDs; establish Baseline Version 1.0.
Design Development through Construction	Manage changes via change control; update DRD version as needed.
Commissioning / Closeout	Finalize as-built DRD reflecting what was actually delivered.

Writing Good Requirements

Use “shall” language for mandatory requirements, not “should,” “may,” or “consider,” which imply optionality.

- "The facility shall accommodate 2,500 guests per operating day."
- "HVAC systems shall maintain zone temperatures between 68°F and 74°F under full occupancy."

Be specific about quantities, tolerances, and conditions:

Requirements vs. Design Solutions

Requirement: "Provide daylight to 60% of occupied floor area."
Design Solution: "Use skylights and clerestory windows."

> The requirement states what must be achieved. The design solution is how you achieve it. Keep them separate so you can evaluate alternative designs against the same requirement.

Keeping the DRD Useful

A DRD is only valuable if people actually use it.

- **Make it accessible.** Store it in a shared, version-controlled location. Everyone should know where to find the current version.
- **Keep it concise.** A 30-page DRD that people read is better than a 200-page document no one opens. Aim for clarity, not exhaustive detail.
- **Reference it in design reviews, RFIs, and submittals.** When design decisions are made, ask: which requirements does this address? Reference requirement IDs in RFI responses.
- **Update it deliberately.** Do not let the DRD become static. As requirements are refined or changed, update the document and version it.

Common DRD Mistakes

Mistake	Why It Matters
Starting too late	Writing the DRD after design is half-done means playing catch-up and documenting decisions retroactively.
Treating it as write-only	A DRD written once and never referenced is useless. It must be a live working document.
Mixing requirements and design	The DRD describes what and why, not how. Keep design solutions in the design documents.
Vague language	“User-friendly,” “adequate,” “sufficient,” “as needed” are not requirements. Write to be tested.
No ownership	If no one is responsible for maintaining the DRD, it will not be maintained.
No change control	If requirements change informally and the DRD is not updated, you have two conflicting baselines.

> **Field Example — DRD Preventing Costly Misalignment**
>
> On a large themed entertainment project, early stakeholder workshops captured functional and performance requirements in a draft DRD. One requirement stated:
>
> FR-012: "The attraction shall accommodate a minimum of 1,800 guests per hour at theoretical capacity, based on a 4-minute dispatch cycle and 12-guest vehicles."
>
> During schematic design, the creative team proposed a ride concept with 8-guest vehicles and a 5-minute cycle, which would yield only about 960 guests per hour — roughly half the required capacity.
>
> Because the capacity requirement was explicit and visible in the DRD, the design team caught the conflict early. They adjusted the vehicle count and loading procedure at far lower cost than if the issue had been discovered during construction or commissioning.
>
> Without the DRD, the mismatch might not have surfaced until systems testing, when redesign would have been

> prohibitively expensive.

3. The Requirements Log and RTM

The DRD captures what the project must achieve. The requirements log and Requirements Traceability Matrix (RTM) are how you track, manage, and verify those requirements through the lifecycle of the project.

The Requirements Log

The requirements log is a structured list — usually a spreadsheet or database — that tracks every requirement from identification through verification. Each requirement gets a unique ID, a source, an owner, a priority, a status, and traceability links to design elements and verification activities.

The log is a living document. It gets updated as requirements are refined, as design progresses, and as verification occurs. You can manage it in any tool; the value comes from structure and discipline, not the software.

Requirements Log Structure

Field	Description	Example
Req ID	Unique identifier	FR-001, PR-042, RCR-015
Category	Type of requirement	Functional, Performance, Regulatory, Operational, etc.
Requirement	Clear, testable statement	"The facility shall accommodate 2,500 guests per operating day."
Source	Where it came from	Stakeholder workshop 9/12/23; DRD §5.1; IBC 2021; AHJ meeting
Owner	Who is responsible	Operations Director, Chief Engineer, PM
Priority	Importance level	Must Have, Should Have, Nice to Have
Status	Current state	Proposed, Approved, In Design, Implemented, Verified, Closed
Verification Method	How it will be tested	Inspection, Test, Analysis, Demonstration
Traced To (Design)	Design element addressing it	Drawing A-301; Spec 26 05 00; BIM Element ID 45372
Traced To (Verification)	Proof it was met	Test Report CT-HVAC-01; Inspection Photo; As-Built Dwg A-801
Notes	Additional context	Dependencies, risks, assumptions, clarifications
Date Added / Last Updated	Version tracking	2023-09-15 / 2024-04-15

Category Prefixes — Quick Reference

Prefix	Category
FR	Functional Requirement
PR	Performance Requirement
SMR	Schedule and Milestone Requirement
BCR	Budget and Cost Requirement
PIR	Physical and Interface Requirement
RCR	Regulatory and Compliance Requirement (including AHJ conditions)
SER	Sustainability and Environmental Requirement
SLR	Security and Life-Safety Requirement
OMR	Operational and Maintenance Requirement

Priority Levels

Priority	Definition	Examples
Must Have	Non-negotiable. Failure to meet this requirement means the project cannot open, operate, or comply with law or contract.	Life-safety egress capacity; ADA minimums; budget ceiling (hard funding constraint); opening date (contractual commitment)
Should Have	Important, but could be deferred or adjusted under cost/schedule pressure with stakeholder approval.	Enhanced finishes; operational efficiency improvements; desirable guest amenities
Nice to Have	Desirable but not essential. First to cut if needed.	Future expansion provisions; premium FF&E upgrades; enhanced landscaping beyond minimum

Priority should be assigned with stakeholder input, not by the PM alone.

Status Levels

Status	Meaning
Proposed	Identified but not yet reviewed or approved by stakeholders
Approved	Accepted by stakeholders and included in the baseline
In Design	Being addressed in design documentation (drawings, specs, analyses)
Implemented	Built, installed, or configured in the field
Verified	Tested, inspected, and confirmed to meet criteria
Closed	Fully satisfied, with documentation complete
Deferred	Postponed to a future phase with stakeholder approval

Rejected	Determined to be out of scope or invalid, with documented rationale

The Requirements Traceability Matrix (RTM)

The RTM is a view into the requirements log that shows the linkages around each requirement: requirements to design elements, requirements to verification activities, requirements to risks, and requirements to change orders.

The RTM answers questions like: Which requirements are addressed in Drawing Set A-300? Which requirements still lack verification evidence? If we cut this system, which requirements are affected?

Req ID	Requirement (abbreviated)	Traced To (Design)	Traced To (Verification)	Status
FR-001	Accommodate 2,500 guests/day	Site Plan A-101; Capacity Model	Opening-day flow test; 30-day ops report	Verified
PR-001	HVAC 68–74°F, full occupancy	Mech Drawings; Spec 23 05 00	Commissioning Test CT-HVAC-01	Verified
RCR-001	Comply with IBC 2021, NFPA 101	All construction documents	Third-party code review; AHJ inspection and certificate	Verified
SLR-007	Emergency egress under 6 min.	Egress plans; FLS narrative	Evacuation drill: 5 min 20 sec with staff	Verified

Verification Methods

Method	Description	Example
Inspection	Visual or physical check against criteria	Verify 12 ADA parking spaces; measure dimensions and signage placement
Test	Operational or performance test with measured results	Run HVAC at full load; measure zone temperatures; confirm 68–74°F maintained for 2 hours
Analysis	Engineering calculation, simulation, or modeling	Structural load analysis confirms 100 psf floor capacity; model shows deflection within L/360 limit
Demonstration	Functional demonstration with users or operators	Staff demonstrate emergency egress procedure; full facility evacuation completed within target time

Assign Verification Methods Early

Decide how you will prove each requirement was met during planning, not during commissioning. If you wait until the end to decide how you will test, you may discover you cannot.

Building and Maintaining the Log and RTM

Step 1: Populate from the DRD. Copy requirements from the DRD into the log. Each requirement gets a row. If the DRD lists 50 requirements, your log starts with 50 rows, each with status Approved.

Step 2: Add requirements from other sources as the project progresses — stakeholder meetings, BIM coordination issues, RFI responses, code review findings, AHJ permit conditions, constructability reviews with contractors. Every new requirement gets logged, reviewed, and either approved or rejected. Do not let phantom requirements live only in email threads.

Step 3: Link to design documentation. As design progresses, update the Traced To (Design) column with drawing numbers, specification sections, BIM element IDs, and analysis reports.

Step 4: Update verification evidence during commissioning. As tests are completed and inspections done, update the Traced To (Verification) column with test reports, inspection records, and certificates.

Field Example — RTM Preventing Costly Rework

On a large themed entertainment project, a specialty contractor proposed a value-engineering change: substitute a different projection system to save cost and reduce lead time.

The PM used the RTM to identify affected requirements, including projection brightness, color accuracy, and maintainability. The proposed substitute met the brightness requirement but failed the color accuracy requirement and introduced risk around proprietary parts.

Because the requirements and their traceability were visible, the team could see exactly which requirements would be compromised. They framed options clearly: reject, accept with formally relaxed requirements, or seek an alternative. They ultimately negotiated an alternate system that met all requirements while still saving money and protecting schedule.

Without the RTM, this would have been an opinion-heavy debate. With the RTM, it was a data-driven decision based on baselined requirements.

4. Requirements as a Living System

Requirements are not static. They evolve as the project learns, constraints shift, and stakeholders refine their needs. The goal is not to freeze requirements early and hope for the best — it is to manage change deliberately so the project can adapt without losing control.

Requirements Evolution vs. Scope Creep

Requirements evolution is normal and healthy:

- Clarifying ambiguous requirements as design progresses and unknowns resolve
- Refining performance targets based on analysis, modeling, or testing
- Adjusting priorities as funding or schedule constraints become real
- Incorporating AHJ feedback during permitting
- Responding to constructability input from contractors while still satisfying the requirement

Scope creep is undisciplined growth:

- Adding new features without approving a change to the baseline
- Promoting “nice-to-have” requirements to “must-have” without evaluating cost and schedule impact
- Stakeholders freelancing requirements directly to the A/E or contractor without logging them
- Design solutions masquerading as requirements (“add a skylight” vs. “provide daylight to 60% of occupied spaces”)
- Requirements added late in construction when they are most expensive to implement

> **The Key Distinction**
> Evolution is deliberate and documented. Creep is accidental and chaotic. The requirements log and change control process are what separate one from the other.

The Change Control Process

Change control is how you keep evolution from turning into chaos.

1. Someone proposes a change (new requirement, modified requirement, or deletion).
2. The change is documented in a change request form or log entry.
3. Impact is assessed: cost, schedule, risk, traceability, and priority effects.
4. Project leadership and relevant stakeholders review the change.
5. Decision: Approve, Reject, or Defer — with rationale documented.
6. If approved: update requirements log, update DRD if warranted, communicate to design and construction via formal channels.

7. If rejected or deferred: document rationale, communicate clearly to the requestor.

Change Request Template

Field	Description
Change ID	Unique identifier (e.g., CR-001, CR-002)
Date Submitted	When the change was proposed
Submitted By	Name, role, organization
Affected Requirement(s)	Requirement IDs affected, or “New Requirement” if adding
Proposed Change	Clear description of add, modify, or delete
Rationale	Why this change is needed; what problem it solves
Impact Assessment	Cost, schedule, risk, traceability impact
Alternatives Considered	Options evaluated; why this one is recommended
Decision	Approved / Rejected / Deferred
Decision By and Date	Who decided and when
Baseline Version Updated	New DRD or log version, if applicable
Implementation Notes	How it will be implemented (design revision, change order, etc.)

Requirements and RFIs

RFIs are a primary source of requirements clarification and evolution during construction. When an RFI touches a requirement:

- **If the answer changes a requirement:** initiate a change request and route through approval.
- **If the answer clarifies an existing requirement without changing it:** update the requirement’s Notes field with the clarification.
- **Always:** reference requirement IDs in the RFI response so the design intent is traceable.

Requirements and Submittals

Submittals (shop drawings, product data, samples) show how the contractor intends to meet requirements. Best practice: link submittals to requirement IDs in the submittal log so reviewers can see exactly which requirements a submittal must satisfy.

Requirements During Value Engineering

With requirements traceability, VE proposals become explicit negotiations against the baseline rather than opinion-heavy debates:

- Use the RTM to identify requirements affected by the proposed VE change.
- Evaluate: does the proposed change still satisfy the requirement as written?
- If not: can you relax or remove the requirement without jeopardizing project success?
- Obtain approval from requirement owners and stakeholders before proceeding.

Requirements-Driven Gate Reviews

Design and construction gate reviews should be requirements-driven, not just drawing-driven.

Traditional Question	Better Question
Are the drawings 30% complete?	What percentage of requirements have design coverage?
Is the project on schedule?	Which Must Have requirements are still unresolved?
Are we within budget?	What is the plan and timing for closing open requirements gaps?

Gate outcomes: Proceed — design is progressing adequately. Proceed with Conditions — design continues but specified issues must be resolved by a set date. Do Not Proceed — significant requirements gaps; design must be reworked.

Field Example — Change Control in Action

On a military training facility project, midway through construction, the customer requested an additional training structure in the compound.

With a basic change control process: a change request was logged, including cost and schedule impact and the affected requirements. Analysis showed that the proposed location conflicted with surface danger zone requirements and certain safety criteria. Alternatives were developed: relocate the structure, defer it to a future phase, or adjust existing range layouts.

A cross-functional group selected an option that met safety and performance requirements, accepted a defined cost and schedule impact, and added new requirements to the log.

Result: the change was made deliberately, with full understanding of cost, schedule, and requirement impacts — and no surprises at commissioning.

5. Requirements in the Field

Requirements do not stop mattering when drawings are issued. In many projects, the most expensive requirement problems show up during construction and commissioning — when design assumptions collide with field conditions, RFIs, submittals, and schedule pressure.

Using the Log During Construction

The requirements log should be an active reference during construction, not an archived document. Keep it current by:

- Reviewing each RFI against the log and updating the Notes field or initiating a change request when the RFI touches a requirement.
- Linking submittals to requirement IDs so reviewers can confirm compliance.
- Flagging requirements that are at risk due to field conditions or proposed substitutions.
- Updating status as requirements move from In Design to Implemented.

Commissioning and Verification

Commissioning is the structured process of verifying that installed systems perform as required. The requirements log and RTM provide the test plan:

- Every requirement with a Test or Demonstration verification method corresponds to a commissioning activity.
- Every requirement with an Inspection verification method corresponds to a site inspection or punchlist item.
- Every requirement with an Analysis verification method corresponds to a document or calculation to be submitted and reviewed.

Before commissioning begins, review the requirements log and confirm:

- Every requirement has a verification method assigned.
- Every verification method has a named person, a schedule, and a defined acceptance criterion.
- Requirements that depend on owner-furnished equipment or third-party testing are coordinated.

What “Done” Looks Like

A requirement is Closed when:

- **Verification evidence exists —** a test report, inspection record, certificate, or as-built document confirming the requirement was met.
- **The evidence is recorded in the log —** the Traced To (Verification) column is populated with a specific reference.

- **An authorized party has accepted it —** the Verification Authority has signed off or the evidence is in the project record.

At substantial completion, you should be able to generate a report showing: total requirements, verified requirements, unverified requirements, and deferred requirements. That report is the handover record.

Field Example — Commissioning with the RTM

On a healthcare facility project, the commissioning agent used the requirements log as the primary commissioning checklist. Each HVAC requirement had a designated test, an acceptance criterion, and a responsible party documented in the log.

When a test failed for two exam rooms — outside air delivery was below the specified minimum — the RTM made it immediate clear which requirements were affected (PR-008 and PR-012) and which design elements were responsible (AHU-3 serving Wing B).

The team corrected the balancing issue, re-tested, and updated the log with the passing results. The owner received a complete record: every requirement, how it was verified, and when it was closed.

6. Requirements in Meetings and the Field

One of the most common failure modes in requirements management is not bad process — it is decisions made in meetings and conversations that never make it into the system.

Capturing Requirements from Meetings

Every significant project meeting is a potential source of new or changed requirements. End significant meetings with a brief decisions-and-requirements recap:

- "Here are today's decisions."
- "Here are the candidate requirements we will review for the log."
- "Did any decisions today create, modify, or retire requirements?"

At a regular cadence — weekly, for example — review meeting minutes, major email threads, and workshop outputs, and update the requirements log accordingly. This keeps the system synchronized with the real project conversations.

Design Workshops and Requirements Harvesting

In structured design workshops, ideas, constraints, and expectations flow fast. Simple techniques to separate requirements from design ideas:

- Assign a dedicated scribe for requirements, separate from the person capturing design concepts.
- Use a visible list labeled "Candidate Requirements" so the group can see when a statement is being treated as such.
- At the end of the workshop, review the candidate list: clarify wording, confirm ownership, and assign tentative priorities.

> **Requirements Hidden in Design Ideas**
>
> Design idea: "What if we did a two-story queue with a balcony overlook?"
>
> Requirement hidden inside: "The queue shall be fully covered and climate-controlled to support year-round operation."
>
> The design team may explore many ways to meet the requirement. The requirement itself should be captured in the log.

Handling Phantom Requirements

Phantom requirements are requirements that exist in someone's head, in an email chain, or in a side conversation with the A/E — but never in the log. They are the leading cause of late-stage surprises.

Countermeasures:

- Any decision that changes how success is defined belongs in the requirements system, not just in someone's memory.
- If a stakeholder contacts the A/E or contractor directly with a requirement, route it back through the log and change control.
- Treat email decisions as candidates until they are logged — not as approved requirements.

Field Example — Meeting Discipline Preventing Scope Drift

On a multi-phase campus project, early design workshops produced many ideas for future phases: extra amenities, optional training spaces, and technology features that were interesting but not funded.

The team used a meeting-minutes template with a Candidate Requirements section and a separate Parking Lot / Future Ideas section. Items aligned with current budget and mission went into Candidate Requirements. Items belonging in future phases went into the Parking Lot, with a note: "Not in current phase scope."

Later, when stakeholders asked, "What happened to the idea for the additional training bay?" the PM could point to the workshop notes showing it was identified as a future-phase concept, and the DRD showing it was not baselined as a requirement in this phase.

Result: a clearer, less contentious conversation about scope. It was not "forgotten" — it was consciously deferred.

7. Lessons Learned and the Next Project

Most projects start as if they are the first of their kind. In reality, your organization has already paid to learn many of the same lessons — often more than once. A structured requirements system gives you a way to capture that experience and carry it forward.

Capturing Lessons from Requirements

At major milestones and at closeout, review:

- Requirements that were hard to meet — took major effort, created risk, or drove cost and schedule pressure.
- Requirements that turned out to be over-specified or unnecessary — costly to implement, little value in operations.
- Requirements that were missed or discovered late, and why — only surfaced during permitting, construction, or commissioning.
- Requirements that were crucial for success but easy to overlook — especially in operations, maintainability, and AHJ coordination.

Useful questions:

- Which requirements would we word differently next time?
- Which requirements needed clearer verification methods?
- Which categories were underdeveloped (operations, maintainability, AHJ)?
- Where did assumptions masquerade as requirements, or vice versa?

Capture lessons as short, concrete statements tied to requirement IDs:

> **Concrete vs. Generic Lessons**
>
> Generic: "Coordinate earlier." "Involve operations sooner."
>
> Concrete: "PR-014 (HVAC noise criteria) was added late in design. Next time, define acoustical performance requirements in the initial DRD and verify early with modeling."

Lessons Learned Table

Lesson ID	Requirement(s)	What Happened	Impact	Recommendation
L-001	PR-014	Added late after noise complaints in mock-up review	Redesign of ceiling and ductwork, 3-week delay	Include acoustical performance requirements in DRD and verify with early modeling

Building a Starter Requirements Library

As you repeat projects of similar types — training ranges, attractions, hotels, clinics — you will see recurring requirement patterns. Instead of reinventing them, build a starter library with:

- **Common functional requirements by facility type —** minimum capacities, guest counts, operating hours, simultaneous uses.
- **Standard performance targets —** thermal comfort bands, acoustical criteria, uptime targets, throughput ranges.
- **AHJ and regulatory patterns —** typical permits, recurring code issues, standard life-safety expectations by jurisdiction or facility type.
- **Operational and maintenance expectations —** access clearances, maintenance intervals, staffing constraints, spare-parts availability.

The library is a starting point, not a checklist. It reduces blank-page time and the risk of repeating omissions, while still allowing tailoring.

Balancing Standardization and Tailoring

Standardize	Tailor
The structure — DRD sections, requirement categories, log fields, RTM views, gate checklists.	Content — targets, tolerances, priorities, constraints, local AHJ requirements, specific operational models.
Patterns, not rigid values — "define explicit throughput targets and verification methods" is a pattern.	Project-specific values — "1,800 guests per hour" is a value.

Raising Organizational Maturity

As you apply this across a portfolio, several things change: the DRD template gets sharper; teams become more fluent in writing clear, testable requirements; gate reviews shift from percent drawings complete to percent requirements covered, implemented, and verified.

Over a few cycles, this produces less rework across projects, fewer late surprise requirements, more predictable budget and schedule outcomes, and a stronger culture of defining success up front and managing to it.

At that point, when you add new tools or automation, you are accelerating a system that already works — not trying to fix a broken process with technology.

Conclusion

You do not need a perfect system to get value from requirements management. You need a clear, modest structure — and the discipline to use it consistently.

The tools in this primer are intentionally simple:

- A Design Requirements Document (DRD) that defines what success looks like.
- A requirements log and RTM that track how those requirements are addressed and verified.
- Basic change control so evolution stays deliberate instead of chaotic.
- Meeting and field practices that keep the system in sync with real conversations and conditions.

Used together, these elements give you a way to catch misalignments early when they are cheap to fix, have more grounded conversations about tradeoffs in cost, schedule, and scope, and hand over a facility with a clear record of what it was designed and built to do.

You will still have hard projects. Not every risk can be foreseen, and not every constraint can be relaxed. But with a running requirements system, you are no longer managing by memory and good intentions. You are managing against a shared, evolving definition of success.

What Comes Next

This primer is the first in the Gallant PM Primer Series. Forthcoming titles build on the same structure:

- **Running Risk —** a practical risk management primer for design and construction projects, built on the same register-and-traceability structure, with risks linked directly to the requirements they affect.
- **Running Change —** a practical change management primer that leans on the RTM and log to make tradeoffs explicit and auditable across phase gates, RFIs, submittals, and value engineering.
- **Running AI —** specific guidance for integrating AI-assisted requirements extraction, conflict detection, and traceability into the project delivery process described here.

Those future layers can be powerful. This primer is about the foundation underneath them. If you and your teams can make these practices routine — on small projects as well as big ones — you will see the difference in fewer surprises, more predictable outcomes, and projects that actually deliver what they set out to achieve.

Gallant Project Solutions LLC

Advisory Services

Gallant Project Solutions helps owners, designers, and constructors implement practical requirements management systems that improve project delivery. The firm is an SBA-certified Service-Disabled Veteran-Owned Small Business (SDVOSB) and Veteran-Owned Small Business (VOSB).

Services

Gallant offers requirements management as a structured consulting engagement — from a productized Requirements Snapshot on a single project, through baseline engagements at project launch, monthly retainers across design and construction, and audit engagements for projects already underway. Each engagement uses the same methods described in this primer, applied at the scale and cadence the project needs.

For current engagement options, sample deliverables, and project examples, visit **www.gallantprojectsolutions.com**.

Work with Gallant

Gallant Project Solutions is a Service-Disabled Veteran-Owned Small Business — SBA-certified SDVOSB and VOSB.

We work with federal agencies, healthcare systems, developers, theme park owners, and A/E firms who need practical requirements management that actually gets used.

Contact: gregory.tuite@gallantprojectsolutions.com
Web: www.gallantprojectsolutions.com