Project Requirements and Quality Management - Cumulative Study Guide

Liam Smith · Version 0.3 · 30 October 2026

Instructor: **Robert Clines**. Lecture coverage: **October 5, 7, 12, 14 and 19, 2026**. Textbook: *Project Requirements and Quality Management*, version 8.0, 314 PDF pages. Coverage ends with the October 19 recording; this is not a completed-course transcript synthesis. Sections marked **Textbook supplement** extend beyond the supplied lectures and carry no inferred instructor-emphasis label.

Use the Quick Review for recall, the Practice Questions to apply the ideas, and the Source and Review Index for coverage, source differences and remaining checks. All workbook citations use **PDF-viewer page numbers**, counting the cover as page 1. Lecture references use the app's numbered, cue-bounded segments and recording-relative times.

Page key

1. The course's central connection

A project can deliver its planned product and still miss the business need. Requirements work identifies what would actually create value; scope planning defines the deliverables and work; quality planning defines how well those deliverables must work. These activities need a shared chain of reasoning rather than independent documents that disagree.

The organizing sequence is: business vision and goal; project objective; problem or opportunity; stated and unstated needs; requirements; proposed solution and features; deliverables for the scope plan and performance criteria for the quality plan. Work backward along this chain whenever a requested feature has no clear purpose. Work forward whenever a business need has no deliverable or acceptance test. *(Oct 7, 00:01:52-00:03:41; segments 4-6.)* (Workbook, PDF pp. 52, 85 and 120.)

Three recurring safeguards are **clear objectives, sufficiently defined requirements and disciplined change control**. Stable requirements mean changes are deliberate and assessed; they do not mean the team is forbidden to learn or replan. *(Oct 5, 00:23:58-00:25:57; segments 80-83.) (Oct 12, 02:16:07-02:22:21; segments 360-374.)*

PM and BA: different responsibilities, shared outcomes

Role	Main question	Typical work
Project manager	How will we deliver the agreed value within project constraints?	Objectives, integration, scope, schedule, finances, resources, risks, stakeholder expectations and change decisions.

Role	Main question	Typical work
Business analyst	What problem should we solve, and what solution would meet the real need?	Current/future-state analysis, elicitation, requirements, models, solution options, validation and value evaluation.
Both	Do stakeholders share an understanding of success?	Communication, clarification, scope discussions, prioritization and agreement on done.

The PM remains responsible for ensuring requirements work exists and is managed, even when a BA performs it. Business analysis is an activity, not simply a job title. A PM, system analyst or business liaison may perform BA work; combining responsibilities does not make either kind of work disappear. *(Oct 5, 00:37:24-00:50:49; segments 121-152.) (Oct 5, 00:53:01-01:01:58; segments 165-184.)* (Workbook, PDF pp. 23-25 and 35-39.)

Four BA activities recur: synthesize diverse information; elicit actual needs; facilitate communication across organizational boundaries; align business needs with feasible technical capabilities. Underlying competencies include analytical thinking/problem solving, behavioral characteristics, business knowledge, communication, interaction skills, and tools/technology. Listening and confirming understanding matter as much as producing documents. *(Oct 5, 00:55:31-01:09:31; segments 171-201.)* (Workbook, PDF pp. 24-25 and 28.)

Frameworks to distinguish

Framework	Members	What it helps you understand
Competing constraints	Scope, schedule, budget, risk, quality, resources	Changes create tradeoffs; establish which constraints drive the decision.
Five PM focus areas	Initiating, planning, executing, monitoring and controlling, closing	When and how project work is organized; activities overlap and repeat.
Seven domains used in this course	Governance, scope, schedule, finance, stakeholders, resources, risk	What must be managed; requirements influence every domain.
Six BA knowledge areas	Planning/monitoring; elicitation/collaboration; requirements life cycle; strategy analysis; requirements analysis/design definition; solution evaluation	The different purposes of BA work, not six strictly sequential phases.

Instructor emphasis: understand the interacting constraints and the seven domains. The performance-domain/focus-area table was emphasized again on October 12 as a reference map. The workbook's prose says “8” domains but lists seven; this guide follows the seven named domains and the lecture. Table numbers and PMBOK page references remain edition checks, not independently verified PMI citations. *(Oct 5, 01:22:37-01:32:45; segments 208-230.) (Oct 5, 01:40:11-01:49:27; segments 248-267.) (Oct 12, 01:54:16-01:58:22; segments 284-298.)* (Workbook, PDF pp. 26-28 and 40-46.)

For the BA knowledge areas, remember their jobs: plan the analysis; discover and confirm information; maintain requirements and their relationships; align change with strategy; organize/model/verify requirements and compare designs; evaluate the solution in use. Work often occurs concurrently, with feedback between areas. *(Oct 5, 01:56:25-02:14:07; segments 286-326.)*

Charter, objectives and priorities

A purpose explains why the project is worthwhile. An objective states the outcome, deliverable and measurable target. In the loyalty-app discussion, platform, timing and adoption targets make the objective more specific; “build an app” alone does not define success. Include explicit in-scope and out-of-scope items and revisit estimates as understanding improves. The instructor used **plus/minus 30%** for that exercise's rough-order-of-magnitude estimate; it is a local exercise assumption, not a universal ROM range. *(Oct 5, 02:24:22-02:39:47; segments 359-424.)*

Prioritization is a reasoned choice among competing needs, not a promise to deliver every brainstormed item. The core loyalty capability supports the app's purpose; notifications or advanced analytics can be lower priority. An approved commitment must be met or formally changed, but an unapproved candidate list is still being refined. *(Oct 5, 02:35:30-02:39:14; segments 408-423.) (Oct 7, 02:15:07-02:23:28; segments 306-328.)*

---

2. Needs, requirements, features and acceptance

Concept	Meaning	Original loyalty-app example
Need	Underlying problem or opportunity that justifies change.	Customers cannot easily see or redeem their loyalty balance.
Requirement	A usable, specific representation of a need that can guide design and verification.	An authenticated customer can view the current points balance.
Feature	A service or characteristic of the proposed solution.	A loyalty-balance screen.
Acceptance criterion	Evidence that an agreed requirement has been satisfied.	The screen displays the balance returned by the loyalty engine for the signed-in account.

The example is editorial, based on the course methods. Begin with the need and derive requirements and features. Beginning with a favorite feature can bias the requirements toward justifying that feature. Avoid treating “need” and “requirement” as synonyms for priority: requirement candidates can be optional, but items in approved scope are commitments. *(Oct 7, 00:05:46-00:12:46; segments 20-34.)* (Workbook, PDF pp. 53-54 and 86-87.)

The four requirement categories

Category	Question answered	Example
Business	Why change, and what organizational outcome indicates success?	Increase repeat-customer engagement.
Stakeholder	What does a particular person or group need?	A customer needs to see points and available rewards.
Solution	What capability or quality must the solution provide?	Display balance and rewards; meet agreed response-time criteria.
Transition	What temporary capability or condition is needed to move to the future state?	Migrate balances, train support staff, or temporarily run old and new systems together.

Functional solution requirements describe behavior, functions and information processing. **Non-functional** requirements describe quality or performance attributes. Displaying a customer's interaction history is functional; loading it within two seconds under defined peak usage is non-functional. **Instructor emphasis:** know this distinction. *(Oct 7, 00:13:41-00:23:54; segments 37-57.)* (Workbook, PDF pp. 55 and 88-90.)

Ten requirement-quality checks

Characteristic	Review question
Consistent	Does it conflict with another requirement?
Complete	Does it contain the information needed for its intended use?
Unambiguous	Can readers interpret it in one agreed way?
Correct	Does it represent the stakeholder's actual need?
Specific	Is it precise enough to guide design?
Feasible	Can it be achieved within technology, resources, budget and timing constraints?
Verifiable	Can a test, inspection or other agreed check establish whether it is met?
Traceable	Can it be connected to a need and to implementation/verification?
Unique	Is it stated once, with a stable identifier?
Prioritized	Is its relative importance understood and agreed?

Do not rely on adjectives such as “fast,” “secure” or “easy.” State the relevant condition and measurable expectation. The lecturer's “more than two failed checks” refinement rule is a classroom heuristic; one critical failure can still make a requirement unusable. *(Oct 7, 00:09:36-00:12:46; segments 28-34.)* (Workbook, PDF pp.

54 and 87.)

3. Elicitation: uncover the problem before the solution

The four course stages are **elicitation, analysis, documentation and verification**, supported throughout by requirements management. Revisit earlier work when modeling exposes gaps or stakeholder verification changes the understanding. *(Oct 7, 00:23:58-00:27:03; segments 58-63.)* (Workbook, PDF pp. 56, 91 and 103.)

Refine the business problem

First agree what problem or opportunity is being addressed. Describe the current state and its consequences in business terms. Identify who is affected, the benefit of improvement and the boundary of the problem. Avoid embedding a chosen technology or product into the problem statement before alternatives have been considered.

The course's first-pass method is: initial statement; narrow/focus it; refine it; identify affected stakeholders; define possible solution context; identify constraints. Narrowing reduces breadth; refining improves precision through evidence and shared understanding. The initial statement needs to provoke useful questions, not answer every one. *(Oct 7, 00:27:06-00:37:08; segments 64-85.)* (Workbook, PDF pp. 56-59 and 92-97.)

Technique	What it reveals	What it does not establish by itself
Fishbone/cause-and-effect	Organized hypotheses about possible causes.	Proof that each branch is a root cause.
Pareto	Categories contributing most to measured frequency or severity.	A guaranteed exact 80/20 split or causal proof.
Process model	Sequence, decisions, inputs/outputs and handoffs.	Every stakeholder's underlying motivation.
Swim-lane model	Who performs each step and where handoffs occur.	That the modeled process is effective merely because it is documented.

These methods complement one another: possible causes, priority causes and operational context. **Instructor emphasis:** process models help locate the questions to ask stakeholders. The regional Pareto example's spoken percentages total **97.5%**, despite the statement that they total 100%; do not force its cumulative series to 100% without resolving the data. *(Oct 7, 00:37:10-00:52:30; segments 86-117.)* (Workbook, PDF pp. 57-59 and 94-95.)

Identify stakeholders and decision makers

Find stakeholder groups, their representatives and the people authorized to approve requirements or resolve conflicts. Include users, customers, support, operations, regulators and people whose work changes even if they never operate the solution. A missed group can mean missed requirements.

The course distinguishes **primary customers** who directly use the result, **operational customers** who operate/support it, and **strategic customers** who judge organizational benefit. Project stakeholders can include delivery participants; solution stakeholders are affected by the deployed result. **Instructor emphasis:** know who has decision authority. *(Oct 7, 01:00:17-01:09:31; segments 138-163.)* (Workbook, PDF pp. 60 and 96.)

Kano: fulfillment and satisfaction

Category	If absent or unmet	If provided	Planning implication
Implicit/basic	Strong dissatisfaction.	Usually taken for granted.	Find hidden must-haves; protect the basic purpose.
Explicit/performance	Dissatisfaction increases as delivery falls short.	Satisfaction tends to grow with performance.	Compare value and cost; negotiate performance levels.
Above-and-beyond/delighter	Usually no disappointment from absence.	Can produce a large positive reaction.	Consider after basics and approved priorities.

The horizontal axis is degree of requirement achievement; the vertical axis is customer satisfaction. Kano is a lens for thinking about needs and technique choice, not a method that automatically discovers every requirement. Interviews and surveys often surface explicit requests; observing real work can expose expected capabilities never mentioned. A feature described as a delighter in one classroom scenario is not permanently a delighter for every user or project. *(Oct 7, 01:15:39-01:22:16; segments 175-187.)* (Workbook, PDF pp. 98 and 154.)

October 14 expands the discussion to five Kano groups. In addition to basic, performance and excitement features, **indifferent** features make little difference to the customer's satisfaction, while **reverse** features reduce satisfaction when present. The extra login steps in the lecture illustrate a possible reverse feature, not a reason to remove a security requirement without assessing risk and obligations. This is additional lecture coverage beyond the workbook's three-curve drawing. *(Oct 14, 00:36:38-00:40:25; segments 99-106.)*

Choose complementary elicitation techniques

Technique	Best contribution	Good practice
Interviews	Individual depth and explanation.	Prepare a purpose; use open/probing questions; listen; reflect understanding; avoid leading questions.
Observation	Actual workflow, workarounds and implicit needs.	Obtain permission, explain purpose, keep a low profile, record promptly and validate notes; debrief afterward.
Surveys	Breadth across many people.	Decide what decisions the results will inform; choose suitable respondents; use neutral questions; pilot-test.
Workshops	Shared discovery, competing perspectives and prioritization.	Set boundaries; facilitate equal participation; separate generation from evaluation; share outcomes promptly.
Document/interface review	Existing rules, constraints and connections.	Confirm whether legacy information is still valid rather than treating it as unquestionable.

Technique	Best contribution	Good practice
Prototyping	Concrete feedback on a possible solution.	Use sketches, wireframes or interactive models to reveal misunderstandings before committing.

Instructor study guidance: know what techniques are available and when they help; he did not ask students to memorize every listed technique. This is not a promise that a particular technique will or will not appear on an exam. *(Oct 7, 01:10:10-01:30:16; segments 165-204.)* (Workbook, PDF pp. 61-66 and 98-100.)

For observation, awareness of being watched may change behavior. Explain that the goal is to understand the process, not evaluate the employee. Ask questions after the work rather than disrupting it, and focus on decisions, handoffs, repeated entry, errors and workarounds. Review observations with the person who performed the work. *(Oct 7, 01:30:17-01:37:08; segments 205-218.)* (Workbook, PDF p. 62.)

For surveys, separate facts from opinions; use one concept per question, balanced response options and defined terminology. Pilot with a small representative group and revise confusing items. The instructor's 10-15-question and 5-10-person pilot suggestions are rules of thumb, not statistical guarantees. Consider response bias and whether different stakeholder groups need different versions. *(Oct 7, 01:37:12-01:43:40; segments 219-231.)* (Workbook, PDF p. 63.)

For workshops, **brainstorming is divergent**: produce ideas without criticism. **Idea reduction is convergent**: clarify, organize, combine duplicates and prioritize. Announce the transition between phases. Multi-voting ranks options by votes; nominal group technique adds structured individual selection/ranking to reduce domination and groupthink. Publish the prioritized list quickly so stakeholders can correct misunderstandings. *(Oct 7, 01:43:40-01:53:32; segments 232-263.)* (Workbook, PDF pp. 64-66.)

Rules and constraints

Business rules govern what the business and its systems must enforce. The lecturer groups them into stimulus/response, operation/precondition, structure, inference and computation. Design constraints restrict implementation choices or the development process: a mandated platform, an existing database, an architectural requirement or a regulatory restriction. Make hidden rules explicit and verify their continuing applicability. *(Oct 7, 01:54:46-02:02:11; segments 266-280.)* (Workbook, PDF pp. 67 and 102.)

A requirement candidate can be prioritized; an applicable legal obligation cannot simply be removed because it receives few votes. Organizational standards may also become mandatory through contracts or governance. The workbook's simple voluntary-standard/mandatory-regulation distinction is a teaching simplification, not a complete rule about every adopted standard.

---

4. Analysis, modeling, documentation and verification

Analysis turns elicited material into actionable requirements: model unclear concepts, evaluate feasibility and prioritize. Feasibility includes technical, economic, operational and schedule questions. Modeling should reveal ambiguity, missing paths and inconsistent assumptions rather than decorate a specification. *(Oct 7, 02:19:48-02:29:54; segments 322-340.)* (Workbook, PDF p. 68.)

Model	Main question	Important elements
Entity-relationship diagram	What information/entities exist and how are they related?	Entities, relationships, cardinality and optionality.
State-transition diagram	What state is an entity in, and what event moves it?	States, transitions, triggering events and edge cases.
Use case/scenario	How does an actor accomplish a goal through interaction?	Actor, goal, sequence, system responses, alternate and exception paths.
Interface analysis	What does the solution exchange with its environment?	Users, other systems, devices and external data.
Prototype	What can stakeholders experience and comment on?	A provisional representation, with fidelity appropriate to the question.

In the course's simple state model, the entity occupies one modeled state at a time. A clock may be unset, set/not ringing, ringing or waiting after snooze. Setting the alarm, reaching the alarm time and pressing snooze are **events**, not states. Do not generalize the example into a claim that every advanced state-modeling notation forbids concurrent states. *(Oct 7, 02:31:23-02:34:33; segments 345-351.)* (Workbook, PDF pp. 69 and 105.)

Actors and use cases

An actor is external to the system being modeled and represents a role, not a named individual. Actors can be people, systems or devices. One person can play several roles, and many people can fill one role. A use case names a goal with a verb phrase: “Store fertilizer,” rather than the capability label “Fertilizer storage.” A scenario is one specific interaction path. *(Oct 7, 02:34:36-02:39:43; segments 352-361.)* (Workbook, PDF pp. 70-74 and 106.)

Association lines show which actors participate in which use cases. **Instructor emphasis:** understand actors and their use-case associations. Identify direct users, departments/locations, support/maintenance roles and non-human interfaces. A requirement-actor discovery matrix uses requirements as rows and actors as columns; examining either direction can expose missing actors or missing requirements. *(Oct 7, 02:39:45-02:44:31; segments 362-372.)* (Workbook, PDF pp. 71-74 and 107.)

Textbook supplement: finish the documentation loop

The October 7 lecture stops at the discovery matrix. The following detailed documentation/verification material is from the workbook, not a later lecture supplied here.

Give each requirement a unique ID, use consistent naming and definitions, record assumptions, and associate models/use cases with functional and non-functional requirements. A feature-attribute matrix can capture benefit, effort, architectural impact, stability, risk and target release. A **vision-scope document** records stakeholder agreement about the proposed solution and boundaries; a **vision statement** explains the principal benefit and why this solution is worth adopting. (Workbook, PDF pp. 76-80 and 109-113.)

Verification includes stakeholder review, traceability to related project elements, signoff and placing controlled documents under change management. Trace both directions: each approved requirement needs delivery/verification evidence, and each planned deliverable should have a justified need or requirement. Stable identifiers and version history help avoid reviewing an obsolete specification. (Workbook, PDF pp. 81 and 114-116.)

---

5. Define scope with a usable WBS

A work breakdown structure organizes the **total project scope** into a hierarchy of deliverables and increasingly detailed work. It is neither a free-form task list nor an organization chart. It helps estimate, schedule, resource, assign, communicate and control the work. The scope baseline is broader than a standalone WBS diagram; the course uses the WBS as a principal scope-control reference. **Instructor emphasis:** understand development/decomposition, requirement-to-WBS links and good WBS design. *(Oct 12, 00:02:27-00:12:43; segments 12-31.)* (Workbook, PDF pp. 119-122 and 128-130.)

Define scope before decomposing it

Scope definition connects stakeholder requirements to a bounded description of the result. The scope statement identifies product scope, deliverables, acceptance criteria and explicit exclusions. Requirements documentation is refined as the team makes these decisions. Check assumptions and stakeholder expectations before treating a boundary as agreed; scope detail may develop progressively in iterative work, while contractual work may need more detail upfront. *(Oct 14, 00:11:45-00:17:31; segments 22-34.)*

The lecture traces inputs to their origins: the charter authorizes high-level direction; requirements documentation supplies requested capability; the assumption log records important uncertainties; project plans supply agreed approaches; organizational assets and environmental factors supply context. The scope statement and refined requirements then feed WBS and quality planning. A vague upstream output can produce a flawed WBS, weak metrics and acceptance criteria that cannot be verified. *(Oct 14, 00:17:40-00:20:36; segments 36-43.)*

Each quality metric should trace to a specific requirement and WBS element. Conversely, ask how each relevant deliverable will be checked. Monitoring scope includes tracking approved work and baseline changes, measuring deliverable quality and routing changes through formal control. A built item that fails its agreed quality criteria is not ready for acceptance just because the physical work exists. *(Oct 14, 00:06:01-00:07:35; segments 11-13.)* *(Oct 14, 00:55:49-01:04:58; segments 139-165.)* (Workbook, PDF pp. 171-172.)

Design principles and effectiveness tests

100% rule: include all the work necessary for approved objectives and only that work. Apply it at every level: the children collectively cover their parent's scope without unexplained gaps or overlap. “100%” describes scope coverage, not a completion percentage or a budget utilization target. Include project management, integration, testing, training, transition and closing where the project needs them. *(Oct 12, 00:14:14-00:23:02; segments 40-68.)* (Workbook, PDF pp. 122 and 130.)

Deliverables rather than actions: name the outcome the team must produce. “Approved design” communicates a different thing from “Hold design meetings.” Scheduling activities can describe actions beneath work packages; that does not make the WBS a meeting calendar.

Effective decomposition: go deep enough to schedule the work, resource it and identify a specific deliverable. Fixed depth is not the rule. The lecturer's three-to-six-level range is an illustration, not a requirement that every WBS have that many levels. A work package is at the lowest WBS level chosen for that project. (Workbook, PDF pp. 122, 130 and 301.) *(Oct 12, 00:09:21-00:23:02; segments 25-68.)*

Creation methods

Method	Approach	Main risk to check
Top-down	Start with major outputs and decompose.	Missing small but necessary deliverables.
Bottom-up	Generate detailed outputs and group upward.	Duplicates, poor grouping, losing the overall objective.
Standards	Use a required organizational, industry or contractual structure.	Forcing a poor fit or including irrelevant work.
Templates	Adapt a useful prior-project starting point.	Inheriting obsolete items or missing unique work.
Hybrid	Combine methods to suit the project.	Failing to recheck completeness and boundaries after combining.

A template accelerates work; it does not replace validation. Ensure group boundaries remain coherent and requirements are covered. *(Oct 12, 00:23:03-00:32:58; segments 69-89.)* (Workbook, PDF pp. 123 and 131.)

Use the WBS to diagnose problems

Symptom	WBS questions
Deadlines repeatedly slip	Is work missing? Are packages too vague to estimate and sequence?
Costs exceed expectations	Can accomplishment and costs be summarized at meaningful points?
Product cannot be used	Were integration, testing, training and transition included?
Scope grows uncontrollably	Are boundaries clear? Are approved changes reflected in the baseline?
No end in sight	Is there a defined endpoint and closing work? Is this actually continuing operations?
Ownership is unclear	Do elements overlap? Are outputs and responsible people identifiable?
Planned work is not done	Is all work represented and linked to actionable packages?

These are diagnostic prompts, not proof that the WBS is the sole cause of every project problem. Update it when changes are approved; an outdated baseline is unhelpful, while one that changes without authorization is unreliable. *(Oct 12, 00:33:02-00:51:41; segments 90-138.)* (Workbook, PDF pp. 124 and 131.)

6. Project quality foundations and the QA/QC distinction

Quality concerns how well the inherent characteristics of a deliverable support the project's agreed objectives. Define standards and acceptance criteria before delivery. Assess outputs against what was agreed, not an undefined ideal of perfection. Quality contributes to success but does not guarantee business success on its own. Responsibility rests with the project team and its leadership, supported by appropriate specialists. *(Oct 12, 01:17:13-01:19:58; segments 208-212.)* (Workbook, PDF pp. 136 and 298.)

Embed quality rather than add it at the end

Inspection detects defects after creation; designing suitable processes and deliverables aims to prevent them. The WBS's completeness, deliverable orientation and work-package tests are already quality practices. Reviews during development support quality; a final test is not a substitute for following effective processes. *(Oct 12, 01:31:23-01:34:14; segments 236-240.) (Oct 12, 01:49:29-01:53:46; segments 273-282.)* (Workbook, PDF pp. 138-145.)

Nine quality considerations

Dimension	Main question	Example
Performance	Does it fulfill its intended purpose?	Search produces relevant results.
Conformity	Does it meet specifications and suit use?	Construction uses the specified materials.
Reliability	Does it work consistently over expected use?	A server performs predictably over time.
Resilience	Can it cope with disruption and recover?	A fallback process handles a system outage.
Satisfaction	Does the user's experience meet expectations?	A functional app is also usable.
Uniformity	Are comparable instances consistent?	Different training groups receive equivalent instruction.
Efficiency	Does it avoid unnecessary resource use?	A recurring report avoids needless manual compilation.
Sustainability	What are the long-term economic, social and environmental effects?	Materials, energy use and waste are considered.
Compliance	Does it satisfy applicable external and organizational obligations?	Required rules are incorporated into design and testing.

Consider every dimension consciously, but negotiate project-specific weights and criteria. Reliable normal operation and resilience after disruption are distinct. Good performance does not excuse nonconformity. Meeting a weak specification does not automatically prove fitness for use. A retailer comparison requires a particular

product, service and quality attribute; brand prestige alone is not a measurement. *(Oct 12, 01:34:17-01:47:06; segments 241-267.) (Oct 12, 02:01:44-02:04:32; segments 321-325.)* (Workbook, PDF pp. 138 and 144.)

Quality versus grade

Grade is a category or rank among items with similar functional use but different technical characteristics. **Quality** is whether the item meets its intended requirements. A basic laptop can be high quality for office work but the wrong grade for intensive editing. An expensive laptop can be poor quality if it fails its requirements. Low quality is a problem; low grade is not necessarily one. **Instructor emphasis:** know the distinction. *(Oct 12, 02:05:21-02:09:50; segments 329-340.)* (Workbook, PDF p. 138.)

QA versus QC

Process in course terminology	Focus	Typical evidence/output
Plan Quality Management	Which standards/criteria apply and how they will be met.	Quality plan, metrics, responsibilities and planned checks.
Manage Quality Assurance	Whether planned processes are appropriate and followed.	Audits/reviews, improvement recommendations and change requests.
Monitor and Control Quality	Whether actual deliverables meet defined criteria.	Measurement results, inspected/verified outputs and corrective action.

Use **QA = process** and **QC = product/deliverable** as the primary distinction. The textbook's process table calls QC outputs *verified deliverables*; the lecturer says *validated deliverables*. Keep the wording difference visible. The shared point is evidence of conformity; stakeholder acceptance is a related decision, not automatically the same event as a test passing. *(Oct 12, 01:47:11-01:53:46; segments 268-282.)* (Workbook, PDF pp. 139, 145, 156, 160 and 178.)

October 14 repeatedly reinforces the process/product distinction and adds a useful response loop: define the standard, follow and review the process, measure the deliverable, then respond. If results fail, ask whether the agreed process was followed and whether it was adequate; repeating inspection alone may not correct a deficient process. *(Oct 14, 00:07:38-00:11:09; segments 14-20.) (Oct 14, 00:42:55-00:45:13; segments 112-116.) (Oct 14, 01:10:08-01:11:03; segments 178-179.)*

Source difference retained: at segment 136 the lecturer briefly describes checking whether the built item works as QA, conflicting with his repeated QA/process and QC/product explanation and workbook PDF pp. 156 and 170. Use the repeated distinction above for recall; keep the conflicting passage in the review record rather than memorizing it as a second definition. Later in the same lecture he calls passed inspection outputs **verified deliverables** and sends them to **Validate Scope** for formal acceptance, making the checking-versus-acceptance separation explicit. *(Oct 14, 00:54:46-00:55:33; segments 136-137.) (Oct 14, 00:58:40-00:59:14; segment 149.) (Oct 14, 01:10:08-01:10:38; segment 178.)*

Gold plating and adaptable change control

Gold plating adds unrequested features or functionality in hopes of pleasing the customer. Approved enhancement is different from uncontrolled addition. Even a useful request should be explicit, assessed and decided through the established process. Ask what changes: scope, time, funding/resources, risk or quality. Help stakeholders obtain valuable changes while making their consequences visible.

The plan must adapt when needs change or gaps are discovered. Formal control makes adaptation accountable; it is not a reason to ignore stakeholder needs. Conversely, "be flexible" is not permission to accept unlimited work inside unchanged time and budget. *(Oct 12, 02:12:09-02:22:21; segments 351-374.)* (Workbook, PDF pp. 139 and 239-242.)

When discussing exceeding quality standards, distinguish approved target performance from unauthorized additional scope. Do not infer a general rule to exceed every requirement or accept missing mandatory criteria. The lecturer's comparison is a discussion prompt with context-dependent answers.

Quality history: learn contributions, check spellings

Contributor	Course association
Walter Shewhart	Statistical quality control, control charts and the improvement-cycle lineage.
W. Edwards Deming	Management responsibility and systematic process improvement; the Japanese Deming award.
Joseph Juran	Quality planning, control and improvement trilogy; application of Pareto thinking to quality.
Kaoru Ishikawa	Cause-and-effect diagrams and company-wide quality improvement.
Philip Crosby	Zero defects and conformity to customer-based specifications.
Genichi Taguchi	Robustness, variation/noise and design-of-experiments methods.
Malcolm Baldrige	U.S. quality-award context.

The instructor advises against memorizing every historical detail; learn the associated ideas. Use **Plan, Do, Check, Act**, confirmed in the workbook, instead of "plan, view, check, act." Names such as "Schuhart," "Duran" and "Debbie" are transcription issues. The exact Taguchi loss-function wording and some award/edition attributions remain qualifications; no unsupported historical claim has been made into an answer requirement. *(Oct 12, 01:20:02-01:31:21; segments 213-235.) (Oct 12, 02:36:54-02:38:56; segments 437-450.)* (Workbook, PDF pp. 137, 143 and 184-185.)

---

7. Quality planning, control tools and assurance reviews

Lecture coverage added: October 14. Module 6 is now supported by the recording as well as the workbook. Material drawn from the shared instructional video is identified separately; appearance in that video is not an additional instructor exam designation.

Plan quality alongside scope

The quality management plan is part of the integrated project management plan. It applies organizational policies, procedures and guidelines to the project's particular quality objectives, activities and resources. Tailor the rigor to scope, complexity and risk instead of copying a template unchanged or inventing unrelated standards. Use it during delivery, not merely as a document filed at kickoff. *(Oct 14, 00:02:50-00:05:58; segments 6-10.)* (Workbook, PDF p. 152.)

Quality-planning step	What it establishes
1. Identify customers and stakeholders	Include primary users, operational/support roles and strategic sponsors; their quality needs may differ.
2. Determine needs and expectations	Uncover explicit requests, assumed basic needs and potential delighters.
3. Negotiate requirements	Reconcile competing needs with risk, value, cost, complexity, timing and obligations.
4. Identify standards and performance criteria	Define the QC metrics and planned QA reviews that make quality assessable.
5. Document tasks and responsibilities	Name who inspects, approves results and owns corrective action.
6. Develop the quality plan	Consolidate the approach, standards, activities, resources and responsibilities.

The steps depend on each other: a precise test of the wrong need still misses the purpose. Negotiation should produce achievable, measurable and agreed requirements. Applicable regulations, contractual obligations and adopted organizational standards are constraints to address, not optional features to vote away. The lecture and workbook simplify standards as voluntary and regulations as mandatory; actual obligations still depend on the project's context. *(Oct 14, 00:20:39-00:25:39; segments 44-57.) (Oct 14, 00:32:27-00:36:37; segments 90-98.) (Oct 14, 00:45:17-00:47:14; segments 117-119.)* (Workbook, PDF pp. 152-153.)

Define metrics before judging results

Every usable quality metric needs **what is measured, the acceptable threshold and how it is measured**. Define conditions, data source, collection/analysis method and responsibility so two reviewers can reach the same conclusion. Response time, defect rate, uptime, first-pass yield and cycle time are examples; select measures relevant to this project's requirements rather than memorizing a generic list. A vague goal such as user-friendly is not yet a usable metric. *(Oct 14, 00:40:28-00:42:53; segments 107-111.) (Oct 14, 01:30:28-01:32:58; segments 202-206.)* (Workbook, PDF pp. 152 and 173.)

Instructor study emphasis: make QA/QC explicit and define what results will be measured against; the outcome must be grounded in documented criteria and data. This is a designation of study importance, not a guarantee about an exam question. *(Oct 14, 01:11:05-01:12:39; segments 180-183.)*

QFD / House of Quality

Quality Function Deployment relates customer needs to product/technical characteristics so a cross-functional team can compare priorities and tradeoffs. In the course diagram, needs are at the left, product features across the top, relationship strengths in the center, market analysis at the right, performance targets at the bottom and tradeoffs in the roof. A relationship matrix supports reasoned choices; it does not itself prove that the final product satisfies every need. *(Oct 14, 00:47:14-00:49:12; segments 120-123.)* (Workbook, PDF pp. 154 and 169.)

Instructor study guidance: recognize QFD as a way to map project quality measurements; the lecturer explicitly says students do not need to know how to construct the complete House of Quality for this preparation. Do not turn the diagram into an unsupported guaranteed-exam task. *(Oct 14, 00:49:14-00:49:35; segment 124.)*

Perform QC as a repeatable cycle

1. Identify the specific deliverable or interim result to check.
2. Select suitable tools and techniques for that result.
3. Collect objective performance data.
4. Compare the data with the defined standard.
5. Respond: proceed when it passes; otherwise investigate, rework/reinspect, escalate or request an authorized change.

An unrealistic standard may need a change request; a failed result is not permission to quietly lower the criterion. Repeated failures may lead to a QA investigation of the process itself. A verified deliverable can proceed to formal acceptance; passing a test does not itself record the customer's acceptance decision. *(Oct 14, 01:05:04-01:11:03; segments 166-179.)* (Workbook, PDF pp. 156, 171-172 and 178.)

Choose among the seven basic QC tools

Tool	Main use	Reading or application check
Cause-and-effect / fishbone / Ishikawa diagram	Organize possible causes of a defined effect.	Branches are hypotheses to investigate, not proof of root cause; categories depend on the context.
Flowchart / process map	Show steps, decisions and handoffs.	In the lesson: oval = start/end, rectangle = task, diamond = decision, arrows = flow.
Check sheet / tally sheet	Collect and organize observations consistently.	Record who, when and where; distinguish a data tally from a checklist of steps to verify.
Pareto diagram	Rank categories by measured frequency or impact and show cumulative percentage.	Bars descend by contribution; the line accumulates it. Prioritize the vital few without forcing an exact 80/20 split.
Histogram	Show the frequency distribution of continuous measurements.	Inspect spread, peaks and shape; it does not display the time order of individual observations.

Tool	Main use	Reading or application check
Control chart	Follow process behavior over time and investigate signals.	Distinguish ordinary variation from signals requiring investigation; see the course rules below.
Scatter / correlation diagram	Plot pairs of measurements to explore a relationship.	In the lesson, the independent variable is on x and response on y. Correlation alone does not prove causation.

The seven-tool list is stated by the instructor and confirmed in the workbook. Fishbone categories should be developed with people who understand the work; arbitrary categories are not an investigation. Flowcharts make process boundaries and decision paths explicit. *(Oct 14, 00:49:37-00:54:13; segments 125-134.) (Oct 14, 01:33:00-01:33:30; segment 207.)* (Workbook, PDF pp. 155, 169-170 and 173.)

The **shared lesson**, rather than a separate instructor designation, adds the check-sheet metadata, histogram shape and scatter-plot examples. Its toaster example shows how complementary tools can organize data, prioritize defects and investigate possible causes. Its claimed 95% problem-solving figure and example-specific humidity/defect targets are not universal facts or memorization requirements in this guide. *(Oct 14, shared lesson, 01:48:53-02:04:26; segments 249-290.)*

Control limits, specifications and sampling

A control chart examines **process behavior over time**; specification/tolerance limits describe **required acceptability**. A stable process can still produce unacceptable output. Do not silently substitute customer specifications for statistically derived control limits or treat the course's recommendation for tighter internal performance as a guarantee that every process is already capable. *(Oct 14, 01:37:12-01:41:25; segments 219-226.)* (Workbook, PDF pp. 174-175.)

For classroom recall, investigate a point outside a control limit or **seven consecutive points on the same side of the mean**, even when those seven points lie within the limits. These are the course's taught signal rules, not a universal substitute for a documented real-process charting method. *(Oct 14, 01:39:36-01:41:25; segments 223-226.)* (Workbook, PDF p. 175.)

Sampling inspects a subset to inform a decision about a population. The sample may misrepresent the population; risk is not automatically small just because the sample is called statistical. **Producer / Type I / alpha risk** rejects a good lot. **Consumer / Type II / beta risk** accepts a bad lot. Selection method, sample size and decision rules matter. *(Oct 14, 01:33:31-01:35:06; segments 208-210.)* (Workbook, PDF pp. 159 and 174.)

Numerical qualification: the lecture and workbook pair about 68% and 99.7% bell-curve coverage with a Six Sigma target of 3.4 or fewer defects per million opportunities. Preserve these as their stated classroom figures, while distinguishing distribution coverage from a defect-rate convention. They are not interchangeable conversions; assumptions and the opportunity definition would need checking. No disputed conversion is made into a graded question. *(Oct 14, 01:35:09-01:36:19; segments 211-212.)* (Workbook, PDF pp. 159 and 174.)

Make a QC plan actionable

Specify the product/service and purpose; performance standard; how and when measurements are collected; responsible owner; and what happens when the criterion is not met. The marker example links writability, erasability and odor to its purpose, then specifies tests/samples and stop, expand-sample or reject responses.

The outpatient-system example connects uptime, retrieval speed and processing time to measurements and actions. Learn the structure rather than adopting those sample rates or thresholds as universal requirements. *(Oct 14, 02:05:09-02:10:49; segments 294-311.)* (Workbook, PDF pp. 176-177 and 311-312.)

The workbook service example calls a 99.7% uptime threshold a control limit. In that example it is a specified performance/action criterion; the page does not establish it as a statistically estimated process-control boundary. Keep that distinction when designing a plan.

Original example: for the study-guide app, QC could check that every displayed citation resolves to its source and that calculations return expected values. QA could review whether the source-validation/review process is followed before publication. A pass rate without a defined denominator or test method is not a usable metric.

Plan and conduct QA reviews

Manage Quality Assurance uses planned, systematic reviews to examine whether project processes meet stakeholder expectations and follow agreed policies and procedures. A process followed perfectly can still be unsuitable; findings may therefore concern either adherence or process design. Reports, change requests and approved plan/document updates close the loop. *(Oct 14, 02:11:04-02:18:15; segments 313-328.)* (Workbook, PDF pp. 160 and 178.)

The five-step QA workflow is: **identify aspects to review; select review type; set schedule and scope; conduct reviews according to plan; initiate changes**. Give more attention to high-risk or unfamiliar processes and processes producing tightly constrained deliverables. Reviews should produce documented findings and assigned action, with necessary changes routed through integrated change control. *(Oct 14, 02:18:17-02:23:00; segments 329-339.)* (Workbook, PDF pp. 160 and 179.)

Review type in this course	Main focus	Planning implication
Team review	Team work output and progress.	Often short and frequent; surface new risks, issues and requirements.
Formal review	A particular deliverable.	Plan appropriate participants, criteria, records and signoff.
Management / phase-gate review	A phase or major continuation decision.	Identify decision authority and evidence for go/no-go.

A **review type** is distinct from its **technique**: surveys, walkthroughs, focused inspections and roundtables can support review work. A quality audit is a structured, independent check of compliance with organizational/project policies, processes and procedures. *(Oct 14, 02:22:22-02:23:39; segments 339-340.)* (Workbook, PDF pp. 160-161 and 179.)

Each review-plan row needs focus, type/technique, responsible person, date/frequency and scope. The example uses weekly team progress, formal deliverable reviews and management phase gates. A product inspection may still be QC even when listed in a broadly titled QA template: classify the actual activity rather than the document label. Quality responsibility is shared by the whole project team; appointing a specialist does not transfer it away. *(Oct 14, 02:24:06-02:26:22; segments 342-346.)* (Workbook, PDF pp. 180-181 and 313.)

8. Quality improvement: October 19 lecture and workbook

October 19 supplies the Module 7 lecture coverage for the improvement sequence, benchmarking, root-cause analysis and methodology comparisons below. The earlier textbook explanations are retained and expanded with dated lecture navigation; Modules 8-9 remain separately labeled preparation.

The instructor contrasts **doing the right things** (effectiveness: achieving useful goals) with **doing things right** (efficiency: output relative to resources). A fast process that repeatedly produces the wrong result is efficient in a narrow sense but ineffective. Aim for both, and assess proposed improvements through the project's change process. *(Oct 19, 00:00:58-00:02:59; segments 3-6.)* (Workbook, PDF pp. 184 and 198-199.)

Quality improvement increases effectiveness and/or efficiency for stakeholders. **Effectiveness** concerns achieving the intended result; **efficiency** concerns resource use in doing so. An improvement may require an approved change. (Workbook, PDF pp. 183-185 and 198-199.)

PDCA and benchmarking

PDCA step	Work
Plan	Define the improvement objective, baseline/benchmark and proposed action.
Do	Implement the plan or improvement trial.
Check	Compare actual results with the baseline and agreed criteria.
Act	Standardize a successful improvement, or revise and cycle again.

Benchmarking compares a selected process or outcome against a useful reference. **Internal** comparisons stay inside the organization; **competitive** comparisons involve a direct competitor; **functional** comparisons examine comparable functions elsewhere; **generic** comparisons look across unrelated functions or industries. Define the measure and understand your own process before interpreting a comparison. (Workbook, PDF pp. 184-186 and 200-201.)

Apply the improvement cycle

During Plan, specify the objective, the measure, the current baseline and the comparison criterion. During Do, execute the plan or trial and investigate problems. During Check, evaluate the results against that objective and benchmark. During Act, make a successful change official through procedures, standards, training or product specifications; if results do not meet the objective, refine the solution and cycle again. The lecture spends much of Do investigating alternatives before implementation; it does not eliminate the later implementation and follow-up steps. *(Oct 19, 00:03:03-00:05:04; segments 7-10.) (Oct 19, 01:59:55-02:01:55; segments 395-398.)* (Workbook, PDF pp. 184-185, 199 and 227-228.)

Select and maintain a benchmark

Benchmark type	Comparison	Example
Internal	Processes within the same organization.	Two departments' handling of comparable requests.
Competitive	A competitor or similar company.	Comparable service performance across insurers.
Functional	A similar function outside the company, potentially in another industry.	Transaction handling in insurance and banking.
Generic	Processes across unrelated industries or functional areas.	Learning from a different kind of workflow.

The instructor groups competitive, functional and generic comparisons as **external benchmarking**. Choose the comparison to answer a specific improvement question. The insurer/bank example is from the lecture; the department and workflow examples are editorial illustrations. *(Oct 19, 00:06:21-00:09:05; segments 14-18.)* (Workbook, PDF pp. 185 and 200.)

The seven-step sequence is: decide what to benchmark; plan the study; understand the process; collect data; analyze it; implement specific actions; and incorporate benchmarking into strategic/tactical planning with periodic reevaluation. A comparison is useful only when its measure, conditions and intended performance criterion are clear. The lecture calls a benchmark a minimum standard for success; distinguish that chosen acceptance criterion from the current baseline and from an external reference value. *(Oct 19, 00:09:06-00:11:25; segments 19-23.)* (Workbook, PDF pp. 185-186 and 201.)

Solve the underlying problem

Problem solving is shared work. A capable problem solver connects activities to organizational goals, understands the process, anticipates issues, investigates causes, remains open to alternatives and follows through to verify the result. The ten barriers discussed include symptom-focused fixes, resistance to change, weak buy-in or management support, technology limitations, inadequate resources, unrealistic expectations, inadequate training, unrealistic timeframes and difficulty adapting to new technology. Several can occur together. *(Oct 19, 00:15:32-00:19:07; segments 48-54.)* *(Oct 19, 00:22:49-00:28:47; segments 73-85.)* (Workbook, PDF pp. 187, 202-205.)

A **symptom** is an observed effect or condition that needs explanation. A **root cause** is an underlying causal explanation reached through systematic investigation. The course defines it as having no deeper cause, while also allowing several causes and recognizing that the problem definition influences the investigation. Do not infer that the first plausible explanation is proven. *(Oct 19, 00:30:04-00:33:24; segments 95-110.)* *(Oct 19, 01:04:53-01:10:22; segments 238-248.)* (Workbook, PDF pp. 188-189 and 206-210.)

The non-starting car, repeated lateness, low morale and increased rework are discussed as symptoms that need investigation. The oversized-hole discussion is less consistent: the lecturer calls it a root cause but then mentions possible deeper explanations, including a drill-bit or requirements issue. Treat that example as a boundary

question about the defined problem; do not memorize that an oversized hole is always a root cause. *(Oct 19, 00:32:52-00:39:15; segments 110-154.)* (Workbook, PDF pp. 206-207.)

Course cause category	What to investigate	Original example
Physical	A tangible component or material fails.	A server component stops working.
Human	A required action is omitted or performed incorrectly.	A maintenance task is missed.
Organizational	A process, policy or system of responsibility is faulty.	No owner or reminder exists for recurring maintenance.

These are related investigation lenses, not mutually exclusive verdicts. The examples here are editorial summaries of the taught distinctions. A physical failure does not establish that a person caused it; trace each proposed connection with evidence. *(Oct 19, 01:10:35-01:12:40; segments 250-254.)* (Workbook, PDF p. 210.)

Carry an RCA through to prevention

Step	Useful output
1. Define factually	A specific event/problem statement, without assuming a cause.
2. Gather data	Observations, measurements and relevant records.
3. Ask why	Testable causal connections rather than a string of guesses.
4. Identify causes	Supported causes, including multiple contributors when present.
5. Choose corrective action	A feasible response to the identified causes.
6. Implement	Completed changes with assigned responsibility.
7. Monitor	Evidence of effectiveness and reduced recurrence.

The instructor gives these seven steps before showing an IBM RCA lesson. That lesson emphasizes causal evidence, improvements to monitoring/logging, short-term restoration plus longer-term corrections, and communication with affected customers after implementation. The media narration appears under the instructor's speaker label in this transcript; it remains **shared-lesson material**, not independently spoken instructor emphasis. *(Oct 19, 00:42:10-00:52:27; segments 163-189.)* (Workbook, PDF pp. 188-189 and 209.)

The expired-certificate example separates **restoring service** from **preventing recurrence**. Replacing the certificate fixes the immediate incident; establishing renewal ownership, reminders or automation addresses the process that allowed it to recur. Select a feasible solution that meets constraints and resolves the cause, then monitor and follow up. The simplest/lowest-cost advice is not permission to ignore compliance or agreed quality. *(Oct 19, 01:05:33-01:08:58; segments 239-245.)* (Workbook, PDF pp. 189 and 210.)

Select a tool for the question

Tool	Contribution	Limit
Five Whys	Follow successive explanations of a defined problem.	Five is a guide, not a mandatory stopping point; test the links.
Fishbone / Ishikawa	Organize possible causes of an effect by category.	Candidate branches still need investigation.
Process / swim-lane map	Trace steps, decisions, responsibilities and handoffs.	Drawing the process does not prove it is effective.
Tree diagram	Break an objective or problem into smaller parts.	Decomposition alone does not establish causation.
Pareto analysis	Rank measured contributions and add cumulative percentages.	The 80/20 relationship is a heuristic, not a forced result.
Brainstorming	Generate diverse possibilities before evaluation.	Agreement and confidence do not prove a cause.
Affinity diagram	Group and label related ideas.	Categorization does not validate the ideas.
Scatter diagram	Explore the relationship between paired numeric variables.	Correlation alone does not establish causation.

The lecture covers these tools from Five Whys through scatter diagrams. Its 50-100-point scatter suggestion is a classroom guideline, not a guarantee that a particular sample establishes a relationship or its cause. The earlier Pareto arithmetic qualification still applies; do not force data to produce exactly 80/20. *(Oct 19, 01:12:44-01:59:51; segments 255-394.)* (Workbook, PDF pp. 190-194 and 211-226.)

Protect independent thinking

In brainstorming, include people with relevant and varied knowledge, give everyone room to contribute, record ideas and delay criticism until the generation phase ends. **Groupthink** substitutes conformity for critical examination: dissent is pressured, leaders' beliefs are reinforced, or apparent agreement is accepted without checking it. Confirm the decision and next actions explicitly. Constructive dissent differs from reflexive opposition. *(Oct 19, 01:42:29-01:53:16; segments 352-378.)* (Workbook, PDF pp. 193 and 220-222.)

Nominal group technique begins with silent individual ideas, then round-robin sharing, private voting and tabulation/reporting. **Affinity diagramming** groups related ideas into labeled categories during or after brainstorming. The lecture's closing shorthand about generating ideas from scratch does not replace the grouping step. The classroom group sizes are examples, not universal staffing requirements. *(Oct 19, 01:53:19-01:58:14; segments 379-391.)* (Workbook, PDF pp. 193 and 223-225.)

Root cause analysis

Describe what happened factually; collect observations and data; investigate why; identify supported causes; choose corrective actions; implement; monitor whether recurrence falls. More than one cause may exist. A fishbone suggests possibilities; evidence and investigation are needed to decide which are causal. Five Whys is iterative inquiry, not a requirement to stop at exactly five answers. (Workbook, PDF pp. 187-193 and 204-219.)

Use affinity diagrams to group ideas, process maps to examine steps, tree diagrams to decompose objectives/causes, Pareto to prioritize measured categories and scatter diagrams to explore relationships. A visible association between variables does not alone establish causation. Avoid blaming a person when the underlying process, training or conditions also need investigation. (Workbook, PDF pp. 189-194 and 221-227.)

Improvement approaches

Approach	Course focus
TQM	Customer-defined quality, management responsibility, systematic process analysis and organization-wide continuous improvement.
Six Sigma / DMAIC	Define, Measure, Analyze, Improve, Control. Use data to understand and improve a process and sustain the result.
Lean	Reduce waste, time and unnecessary cost while protecting/improving value and quality.

(Workbook, PDF pp. 194-195 and 228-231.)

The instructor's TQM comparison keeps customer needs, management responsibility, systematic process improvement and participation throughout the organization together. Six Sigma emphasizes data, variation/defect reduction and sustaining results through DMAIC. Lean emphasizes customer value, waste reduction, continuous improvement and respect for people. These are introductory comparisons, not full certifications in the methods. *(Oct 19, 02:01:58-02:07:21; segments 399-413.)*

The workbook's Lean slide spells **Kaizen** and **Genchi Genbutsu**. Use Kaizen for small, continuing improvements involving people across the organization, and Genchi Genbutsu for going to the source to observe the actual work. The transcript's "Tizen" and "Genshi Genbutsu" are recorded transcription differences. The slide also lists Challenge, but its short lecture description and historical founding claim are not used as definitive definitions or graded facts. *(Oct 19, 02:05:50-02:07:21; segments 411-413.)* (Workbook, PDF p. 230.)

Keep the numerical qualification: October 19 repeats the classroom 3.4 defects-per-million figure. It does not resolve the assumptions behind converting a sigma level to a defect rate; this update adds no graded conversion question. *(Oct 19, 02:03:18-02:05:12; segments 402-409.)*

Study-emphasis limit: the instructor says bold/highlighted skills are important, but the transcript does not identify which items are highlighted. The October 19 evidence record stays uncertain, with no invented circled/starred topic designation. The end-of-module questions are useful recall practice without being guaranteed exam items. *(Oct 19, 00:18:34-00:19:07; segment 54.) (Oct 19, 02:07:24-02:13:56; segments 414-451.)*

9. Textbook supplement: establish and use project control

The textbook contains Modules **8 and 9** even though the front contents page stops at Module 7. These sections extend beyond the supplied Course 3 lectures. October 12 introduces change control and October 14 reinforces scope/quality monitoring and routing changes; neither establishes complete lecture coverage of Modules 8-9. October 19 covers Module 7 improvement; it does not add detailed Module 8-9 instruction.

Performance control

Define necessary and sufficient control factors, their thresholds, measurement method, collection ownership and review cadence. Include interim results, major deliverables, milestones and phases; “levels of control” are not merely warning-versus-critical labels. Use tangible, measurable work and forecast as well as actual performance. (Workbook, PDF pp. 233-238 and 247-250.)

Original example: a baseline milestone is due in ten working days, with an agreed escalation threshold of more than two days' forecast slippage. A forecast finish on day 13 triggers escalation. “Monitor the schedule” without a baseline, trigger and owner is not a control plan.

The control cycle is: collect actuals/forecasts; compare with plan; analyze variance causes and impact; take corrective action; update approved plans; report status and revisions. Completing activities on time alone does not establish project success if value, quality, costs or stakeholder requirements are missed. (Workbook, PDF pp. 238, 243 and 258.)

Change control and emerging requirements

Use a controlled intake/log, assess the need and benefit, evaluate effects across the project, and route the decision to the authorized individual or board. The textbook flowchart allows acceptance, rejection or deferral. Communicate the outcome; integrate accepted changes into the plan and controlled records. Configuration management keeps product characteristics, versions and approved changes aligned with what is actually delivered. (Workbook, PDF pp. 239-242 and 251-254.)

For a new requirement, establish why it is needed, assess effects on the current environment/system/users, use change control and update requirement records. Reprioritize features, compare tradeoffs across the whole project and verify decisions with stakeholders. Size governance to the project while retaining clear authority and records. (Workbook, PDF pp. 257-261 and 268-273.)

10. Apply the chain to one project

The following is an **original practice example**, not an additional instructor transcript or a mandated solution.

Artifact	Study-guide app example
Business goal	Reduce time spent turning class recordings into reliable study material.
Problem	Manual quote-correction cycles delay usable guides.

Artifact	Study-guide app example
Need	Preserve trustworthy source support while recovering from straightforward output errors.
Functional requirement	Save a rejected proposal and produce a reviewable correction or explicit failure.
Non-functional requirement	Never publish a response that fails the agreed source validator.
Transition requirement	Existing approved checkpoints remain usable after the update.
Deliverable	Validated repair stage with review records and operator instructions.
QC	Exercise valid and invalid source/quote cases and inspect generated outputs.
QA	Review whether repair changes follow the prescribed evidence and approval rules.
Change control	Assess the added behavior, approve its scope and record the new version.

Ask: which stakeholder approves recovery behavior? What are acceptable automatic changes? What ambiguity must stop the process? Can the result trace from need to requirement to code/tests and user-visible output? This is the course's integration of requirements, scope, quality and governance in practice.

Coverage boundary and future updates

Module 6 has October 14 lecture support and Module 7 has October 19 lecture support. Keep Modules 8-9 and the documentation supplement clearly labeled. Add new lecture references only where later recordings actually cover them; add any instructor-designated emphasis without promoting every existing regex match. Revisit unresolved edition details and numeric examples. Preserve the original session records and review history so corrections remain traceable.