# 1-Review

## What is Review

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

A **review** in software testing is a **systematic examination of artifacts** (documents, code, diagrams, test cases, etc.) to find defects, gaps, or improvements **without executing the software**.

It’s a **static testing technique** — meaning you analyze the product of development rather than running the software itself.

## Purpose

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Detect defects early (before coding or during coding).
* Ensure the artifact meets its intended purpose (business, technical, legal, or quality goals).
* Improve quality and reduce rework costs.

## Key Points

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Conducted by people** (peers, stakeholders, experts) — may be manual or tool-assisted.
* **Applies to any artifact** in the SDLC: requirements, designs, code, test cases, user manuals, etc.
* **Part of verification**: “Are we building the product right?”
* **Can also contribute to validation** when stakeholders confirm the artifact meets needs.

## Examples of Review Types

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

1. **Informal review** – quick, unstructured (e.g., a developer asking a teammate to glance at code).
2. **Walkthrough** – author leads reviewers through the artifact to gather feedback.
3. **Technical review** – experts check for technical correctness.
4. **Inspection** – formal, structured process with defined roles and checklists.
5. **Peer review** – colleagues at the same level review each other’s work.

## 💡 Example:

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Artifact:** Login feature requirement.
* **Review Finding:** Requirement says “user logs in with username.”
* **Reviewer Feedback:** Needs to include email login for flexibility.
* **Outcome:** Requirement updated before design starts.

## Success Factors for Reviews

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

**Success Factors for Reviews** are the key conditions that make a review (whether it’s informal, walkthrough, technical review, inspection, or management review) effective and worth the time invested.

### 1. Clear Objectives

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Everyone must know **why** the review is happening.  
  + Example: Detect defects early, check compliance, verify business coverage.
* Avoid vague purposes like “just to check” — define specific outcomes.

### 2. Well-Defined Entry and Exit Criteria

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Entry Criteria** → Review starts only when the artifact is ready (no work-in-progress).
* **Exit Criteria** → Review ends only after all findings are logged and action items are assigned.

### 3. Right Participants

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Select people with the **needed expertise** for the artifact being reviewed.  
  + Example: UI designer in a mockup review, security expert in a code review.
* Avoid having too many participants — 3–6 people is often ideal.

### 4. Prepared Reviewers

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Participants should **read the material in advance** and note potential issues.
* No “cold reading” in the meeting — unprepared reviewers waste time.

### 5. Use of Checklists

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Checklists help reviewers focus on **consistent quality aspects**.  
  + Example: Requirements review checklist (clarity, completeness, testability).
  + Example: Code review checklist (security, performance, readability).

### 6. Structured and Moderated Meetings

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* A **moderator** ensures the discussion stays on topic and time is used well.
* The review should be **time-boxed** (e.g., 1–2 hours) to keep focus.

### 7. Recording and Tracking Findings

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* All issues should be **documented** (not just verbally mentioned).
* Assign **owners** and **due dates** for resolving each issue.

### 8. Focus on Product, Not People

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Avoid personal criticism — focus on improving the artifact, not blaming the author.
* This keeps the review constructive and encourages participation.

### 9. Follow-Up

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Verify that agreed fixes and improvements are **actually implemented**.
* Without follow-up, reviews lose their value.

### 10. Supportive Organizational Culture

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Management should value reviews and give teams **time** to conduct them.
* Teams should see reviews as a **quality improvement tool**, not a bureaucracy.

### **Real-World QA Example**

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

**Artifact:** Test Plan for a healthcare system.

* **Objective:** Ensure all compliance requirements are covered before execution.
* **Checklist:** Includes HIPAA data handling, encryption test coverage, recovery scenarios.
* **Participants:** QA lead, compliance officer, business analyst.
* **Finding:** Disaster recovery testing not scheduled.
* **Outcome:** Recovery tests added to execution plan, avoiding a regulatory gap.
* **Follow-Up:** QA lead verifies added tests before execution starts.

If you want, I can map **these success factors** directly to **each review type** so you can see exactly how they apply differently in **technical reviews, walkthroughs, inspections, and management reviews**. That would make them much easier to distinguish.

## Roles and Responsibilities in Reviews

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

Here’s a clear breakdown of the **roles and responsibilities in reviews** (based on ISTQB principles and real-world practice):

### 1. Author

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Who they are:** The person who created the work product (document, code, design, etc.).
* **Responsibilities:**
  + Prepare the artifact for review (complete, correct format).
  + Explain the content to the review team (if needed).
  + Fix defects found during the review.
  + Participate in review meetings.

### 2. Review Leader / Moderator

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Who they are:** The person responsible for managing the review process.
* **Responsibilities:**
  + Plan the review (select participants, set schedule).
  + Distribute materials to reviewers in advance.
  + Ensure the review follows the agreed process.
  + Facilitate discussions without bias.
  + Summarize findings and next steps.

### 3. Reviewer

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Who they are:** Subject matter experts, peers, testers, business analysts, or stakeholders who examine the work product.
* **Responsibilities:**
  + Examine the artifact for defects, issues, or improvements.
  + Prepare comments before the meeting.
  + Participate in review discussions.
  + Suggest corrections or enhancements.

### 4. Scribe / Recorder

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Who they are:** Person assigned to document all identified issues during the review meeting.
* **Responsibilities:**
  + Record defects, comments, and decisions.
  + Ensure nothing is missed during discussions.
  + Share documented findings with the team after the meeting.

### 5. Manager / Sponsor (optional role)

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Who they are:** Higher-level authority who sponsors or enforces the review process.
* **Responsibilities:**
  + Ensure review resources are available.
  + Encourage participation and compliance with review policy.
  + Use review results for quality tracking.

### **Example in Practice**

**Artifact:** Software Requirements Specification (SRS)

* **Author:** Business Analyst who wrote the SRS.
* **Review Leader:** QA Lead managing the review meeting.
* **Reviewers:** Developers, testers, product owner, UX designer.
* **Scribe:** Junior tester documenting issues.
* **Manager:** Project Manager ensuring review happens on schedule.

## Early and frequent stakeholder feedback

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

**Early and frequent stakeholder feedback** means actively involving stakeholders (customers, business owners, end users, regulators, domain experts, etc.) *from the very beginning* of a project and continuing to gather their input regularly throughout development — not just at the end.

### Key Points

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

1. **Early** → Get feedback as soon as there’s something to review  
   * Could be ideas, sketches, mockups, requirement drafts, design diagrams, prototypes — *before* building the full solution.
2. **Frequent** → Keep the feedback loop short and continuous  
   * Regular check-ins, demos, review meetings, or user validation sessions.

### Why It’s Important

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Catch misunderstandings early** → Fixing them before development saves time and cost.
* **Ensure alignment** → Keeps the product in line with actual needs and expectations.
* **Avoid late surprises** → No last-minute “this isn’t what we wanted” moments.
* **Improve quality** → Continuous validation of requirements, design, and usability.
* **Increase stakeholder trust** → They see progress, know their input matters.

### Example

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Project:** Mobile banking app
* **Early feedback:** Stakeholders review wireframes and point out that login should accept account number *or* email, not just username.
* **Frequent feedback:** Weekly demo sessions with business users to validate new features like funds transfer, bill payment, and notifications.
* **Outcome:** Misunderstandings are resolved quickly, avoiding major rework later.

If you think of it simply:

**Early** feedback prevents you from building the *wrong* thing.  
 **Frequent** feedback prevents you from building the *right* thing *in the wrong way*.

If you want, I can show you **how early & frequent stakeholder feedback directly fits into the review process flow** so it’s easier to connect the two.

### **benefits of early and frequent stakeholder feedback**

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

Here are the **benefits of early and frequent stakeholder feedback** — both in general projects and specifically for software testing/validation:

#### 1. Catching Issues Before They Get Expensive

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Early detection of misunderstandings** in requirements, UI design, or workflows prevents costly rework later in development.
* Fixing a wrong requirement during the **requirements phase** costs far less than fixing it after release.
* *Example:* Stakeholders clarify early that the e-commerce site must support PayPal — avoiding code changes after launch.

#### 2. Aligning the Product with Real Needs

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Ensures the **solution matches actual business processes** and end-user expectations.
* Prevents the "we built what was asked, not what was needed" problem.
* *Example:* In a hospital booking system, staff feedback changes “Choose Doctor first” to “Choose Department first” to match real workflows.

#### 3. Improving Requirement Clarity

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Frequent review sessions **eliminate vague, incomplete, or conflicting requirements**.
* Stakeholders help refine terminology so it’s consistent and unambiguous.
* *Example:* “Secure storage” updated to “AES-256 encryption at rest and TLS 1.3 in transit” after legal review.

#### 4. Building Trust and Collaboration

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Stakeholders feel **ownership and involvement** in the product.
* Improves communication between business, developers, and testers.
* Creates a partnership rather than an “us vs them” relationship.

#### 5. Reducing Project Risks

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Early feedback helps identify **regulatory, security, or usability gaps** before they become blockers.
* Allows time to plan for changes without last-minute panic.
* *Example:* Compliance team spots HIPAA logging gaps before development begins.

#### 6. Faster Time to Market

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Continuous feedback means **fewer large course corrections** at the end.
* Development flows smoothly because requirements stay validated along the way.

#### 7. Better User Experience

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Frequent user input ensures the **interface and workflows are intuitive**.
* Accessibility, responsiveness, and navigation issues are spotted early.

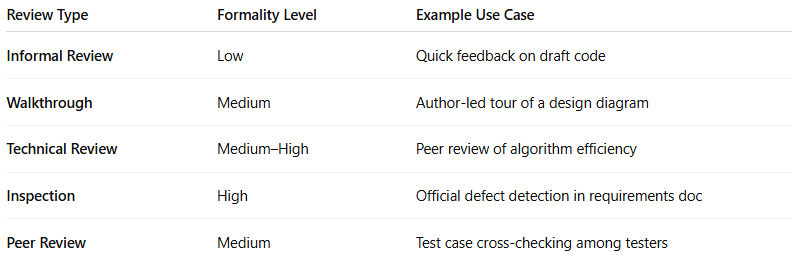
#### 8. Continuous Improvement Culture

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Regular feedback loops make the team **adaptable and responsive to change**.
* Encourages **incremental delivery and validation** — a key Agile principle.

#### Quick classification of review types for context

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**



| **Review Type** | **Formality Level** | **Example Use Case** |
| --- | --- | --- |
| **Informal Review** | Low | Quick feedback on draft code |
| **Walkthrough** | Medium | Author-led tour of a design diagram |
| **Technical Review** | Medium–High | Peer review of algorithm efficiency |
| **Inspection** | High | Official defect detection in requirements doc |
| **Peer Review** | Medium | Test case cross-checking among testers |

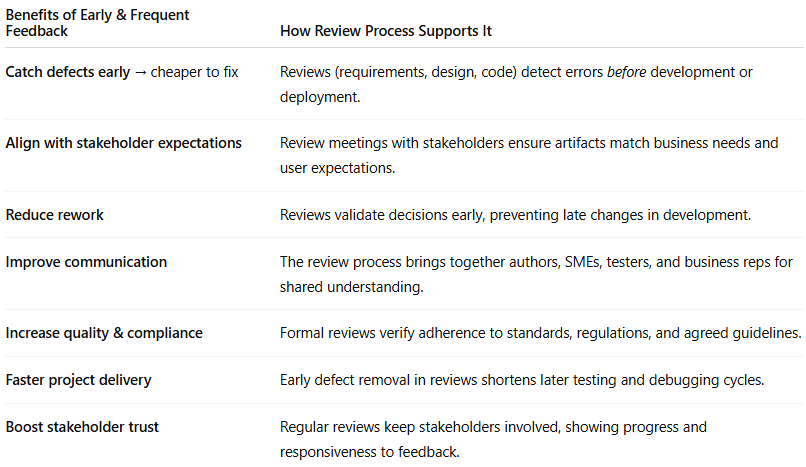
### RelationBetween benefits of early and frequent stakeholder feedback — And Review

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

The **benefits of early and frequent stakeholder feedback** and the **review process** are closely linked — in fact, reviews are one of the *main mechanisms* to get that feedback before costly mistakes happen.

#### How They Relate

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**



| **Benefits of Early & Frequent Feedback** | **How Review Process Supports It** |
| --- | --- |
| **Catch defects early** → cheaper to fix | Reviews (requirements, design, code) detect errors *before* development or deployment. |
| **Align with stakeholder expectations** | Review meetings with stakeholders ensure artifacts match business needs and user expectations. |
| **Reduce rework** | Reviews validate decisions early, preventing late changes in development. |
| **Improve communication** | The review process brings together authors, SMEs, testers, and business reps for shared understanding. |
| **Increase quality & compliance** | Formal reviews verify adherence to standards, regulations, and agreed guidelines. |
| **Faster project delivery** | Early defect removal in reviews shortens later testing and debugging cycles. |
| **Boost stakeholder trust** | Regular reviews keep stakeholders involved, showing progress and responsiveness to feedback. |

#### Example Connection

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* **Scenario:** In an **E-commerce SRS review**, stakeholders point out missing PayPal and Apple Pay support.  
  + **Benefit:** This feedback, given early, avoids costly redesigns in development.
  + **Review Link:** The formal review session is the structured activity where that feedback is collected, discussed, and integrated.

If you imagine it visually:

**Review process** = the *vehicle* 🚗  
 **Early & frequent feedback** = the *destination* 🎯  
 Without reviews, feedback is random and late; with reviews, it’s structured and timely.

#### Summary

**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

**a review can be one of the main ways to get frequent stakeholder feedback.**

**Relationship**

* **Overlap:** Reviews are a formal mechanism for collecting stakeholder feedback.
* **Difference:**
  + **Reviews** are one type of feedback activity (formal, planned).
  + **Frequent stakeholder feedback** also includes *informal* or *iterative* interactions — like quick check-ins, early prototypes, or usability sessions.