# Technical Review

## What is Technical Review

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

A **Technical Review** is a **formal, structured type of review** focused on evaluating a work product’s **technical quality** and **conformance to specifications, standards, or best practices**—without actually executing it.

It’s usually performed by **technical peers** (developers, architects, testers, subject matter experts), not end-users or business stakeholders.

## **Key Characteristics**

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

* **Focus:**
  + Technical accuracy
  + Feasibility
  + Compliance with requirements and standards
  + Maintainability and performance considerations
* **Led by:** A **trained moderator** or facilitator (not always the author).
* **Preparation:** Reviewers receive the work product in advance, study it, and prepare comments.
* **Formality:** More structured than walkthroughs; issues are formally logged.

## **Typical Participants**

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

* **Moderator** — runs the meeting and ensures the process is followed.
* **Author** — answers technical questions, explains logic.
* **Reviewers/Experts** — bring domain or technical expertise.
* **Recorder/Scribe** — documents findings.
* **Manager** (optional) — observes progress.

## **Example**

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

📄 **Artifact:** Database design document for a hospital management system.  
 👤 **Activity:** A technical review meeting is scheduled with senior developers, the DBA, and the security architect.

1. Reviewers check if **foreign key constraints** match the requirements in the ER diagram.
2. Security architect points out that **patient data encryption** is not specified at the database level.
3. DBA notes that some tables **lack indexes** needed for performance.
4. All findings are logged in the review report.
5. Author updates the design before implementation begins.

## **Benefits**

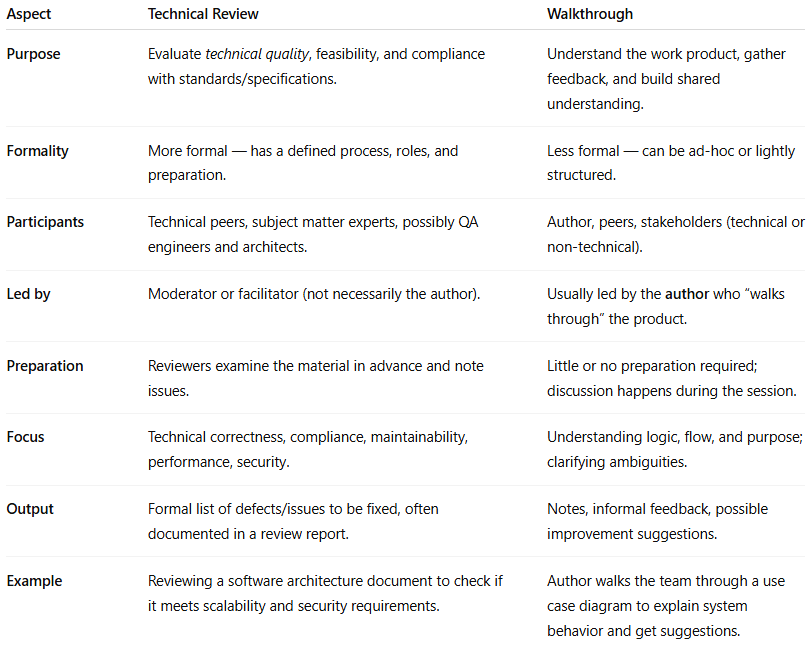
**# Source**: **Chatgpt (GPT-5-Standard) at [8/14/2025]**

* Detects **design flaws** early.
* Ensures compliance with **coding/design standards**.
* Improves **system performance, maintainability, and security** before coding starts.
* Saves cost by preventing expensive rework later.

## Technical Review vs Walkthrough review

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

Here’s a **side-by-side comparison** of **Technical Review** vs **Walkthrough** so you can clearly see the differences:



| **Aspect** | **Technical Review** | **Walkthrough** |
| --- | --- | --- |
| **Purpose** | Evaluate *technical quality*, feasibility, and compliance with standards/specifications. | Understand the work product, gather feedback, and build shared understanding. |
| **Formality** | More formal — has a defined process, roles, and preparation. | Less formal — can be ad-hoc or lightly structured. |
| **Participants** | Technical peers, subject matter experts, possibly QA engineers and architects. | Author, peers, stakeholders (technical or non-technical). |
| **Led by** | Moderator or facilitator (not necessarily the author). | Usually led by the **author** who “walks through” the product. |
| **Preparation** | Reviewers examine the material in advance and note issues. | Little or no preparation required; discussion happens during the session. |
| **Focus** | Technical correctness, compliance, maintainability, performance, security. | Understanding logic, flow, and purpose; clarifying ambiguities. |
| **Output** | Formal list of defects/issues to be fixed, often documented in a review report. | Notes, informal feedback, possible improvement suggestions. |
| **Example** | Reviewing a software architecture document to check if it meets scalability and security requirements. | Author walks the team through a use case diagram to explain system behavior and get suggestions. |

💡 **Quick way to remember**:

* **Technical Review** → “Check if it’s *technically right* before we build it.” ✅🔍
* **Walkthrough** → “Let me *show you what I made*, and you tell me if it makes sense.” 🗣️👥

Alright — let’s strip away the jargon and make **Technical Review** vs **Walkthrough** crystal clear.

### 1️⃣ Imagine two scenarios

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

#### Scenario A — Technical Review

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

* You’ve drawn the **blueprints** for a new house.
* Before building, you call in **architects and engineers**.
* They don’t care about your “story” — they check **math, safety codes, and structural integrity**.
* They bring a checklist, mark problems, and give you a **formal defect list** to fix.

💡 This is **Technical Review**:

A **formal, structured** check for *correctness* against **technical standards** and requirements.

#### Scenario B — Walkthrough

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

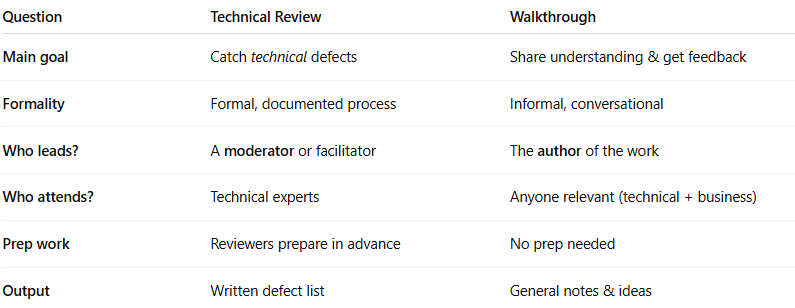
* You’ve drawn the **same house blueprint**.
* You gather your **friends, family, and maybe a contractor**.
* You walk them through the drawing: *“Here’s the living room, here’s the kitchen…”*
* They ask questions: *“Can the kitchen be bigger? Where’s the window?”*
* You take notes and decide later what to change.

💡 This is **Walkthrough**:

An **informal** “show-and-tell” session to explain your work and get feedback for clarity and improvements.

### 2️⃣ Key Distinction

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



| **Question** | **Technical Review** | **Walkthrough** |
| --- | --- | --- |
| **Main goal** | Catch *technical* defects | Share understanding & get feedback |
| **Formality** | Formal, documented process | Informal, conversational |
| **Who leads?** | A **moderator** or facilitator | The **author** of the work |
| **Who attends?** | Technical experts | Anyone relevant (technical + business) |
| **Prep work** | Reviewers prepare in advance | No prep needed |
| **Output** | Written defect list | General notes & ideas |

Think of it like this:

* **Technical Review** → *“Quality control checkpoint”* ✅🔍
* **Walkthrough** → *“Story time + open feedback”* 📖💬

### Technical Review – Real Software Example

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

**Artifact:** API Design Document for a Payment Gateway  
 **Activity:** Technical Review session  
 **Attendees:** Senior backend developers, security engineer, architect, QA lead

**What happens:**

* Reviewers have the API spec *before* the meeting.
* Security engineer checks that card data handling follows **PCI DSS**.
* Architect verifies scalability and error handling.
* Backend devs spot inconsistent field names and a missing authentication flow.
* Findings are logged as **formal defects** in Jira.

**Goal:** Catch **technical issues** before coding starts.  
 ✅ *Defect list is the outcome.*

### Walkthrough – Real Software Example

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

**Artifact:** New UI Prototype for a Banking Mobile App  
 **Activity:** Walkthrough session  
 **Attendees:** UX designer (author), product owner, QA tester, business analyst

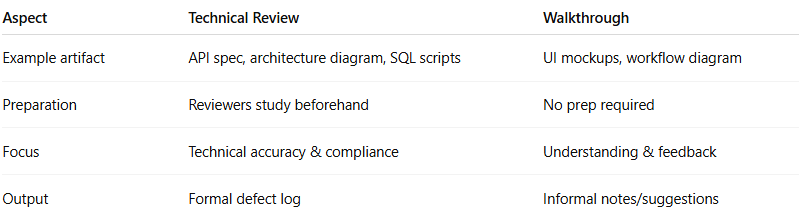
**What happens:**

* Designer shares Figma screens during the meeting.
* Walks everyone through the login, account summary, and transfer flow.
* Product owner suggests adding “Quick Balance” view.
* Tester points out “Forgot Password” link is too small on mobile.
* Notes are taken, but **no formal defect list** — just improvement suggestions.

**Goal:** Build **shared understanding** and get feedback for clarity & usability.  
 💬 *Conversation is the outcome.*

### **Side-by-side in software terms**

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



| **Aspect** | **Technical Review** | **Walkthrough** |
| --- | --- | --- |
| Example artifact | API spec, architecture diagram, SQL scripts | UI mockups, workflow diagram |
| Preparation | Reviewers study beforehand | No prep required |
| Focus | Technical accuracy & compliance | Understanding & feedback |
| Output | Formal defect log | Informal notes/suggestions |

### Real Testing Context: Technical Review vs Walkthrough

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

Got it — you don’t just want the **textbook definition**, you want to see **exactly how Technical Review vs Walkthrough play out in real-world software testing**, step-by-step, so you can tell them apart when you actually see them happening.

Let’s dive in.

### **1️⃣ Walkthrough in Real Testing**

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

**Scenario:** A QA engineer has just written **20 new test cases** for a mobile banking app’s “Money Transfer” feature.  
 They want to make sure everyone understands the approach before executing them.

**How it happens:**

1. **Author-led session** – The QA engineer shares their screen on Zoom with the Product Owner, another QA, and a developer.
2. They say:  
     
     
    “Here’s Test Case #5: User transfers money to a new payee. Steps are here. Expected result: confirmation message.”
3. The audience asks questions like:  
   * “Should we also test when the amount exceeds the daily limit?” (Business analyst)
   * “What happens if the recipient account number is invalid?” (Developer)
4. The QA engineer writes down these suggestions to consider updating the test cases.
5. **No defect log** is created — just informal feedback.
6. **Outcome:** Understanding improves, and new ideas for tests are noted.

**Purpose:** ✅ Share knowledge  
 ✅ Gather broad feedback  
 🚫 Not a formal defect-finding process

### **2️⃣ Technical Review in Real Testing**

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

**Scenario:** Before the same test cases are approved for execution, the **test lead** wants to ensure they meet the testing standards and cover all requirements.

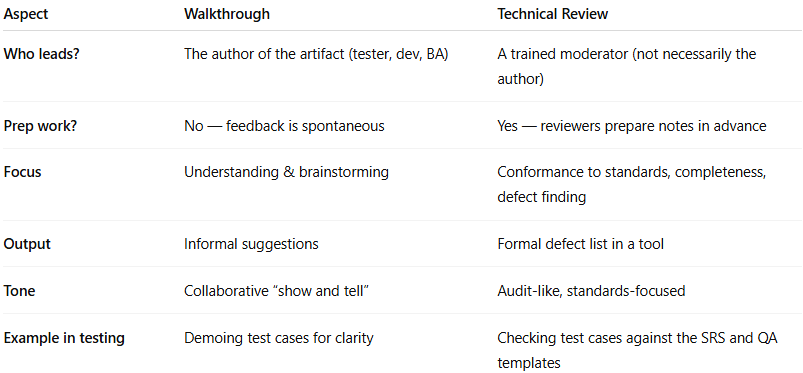
**How it happens:**

1. **Moderator-led process** – The test lead sends the test cases to 3 senior QA engineers **before the meeting**.
2. Reviewers **prepare in advance** by:  
   * Checking traceability matrix → Does each requirement have a test?
   * Checking format → Are preconditions, steps, and expected results clear?
   * Checking for **technical coverage gaps** → Missing boundary value cases, missing negative tests.
3. In the review meeting:  
   * Reviewer:  
       
       
      “Test Case #5: Missing validation for when the transfer amount equals the maximum allowed limit.”
   * Reviewer:  
       
       
      “In Test Case #8, the expected result says ‘confirmation message,’ but requirement REQ-12 specifies the exact message text — please update.”
4. **All issues are logged** in a defect tracking system (e.g., JIRA) as “Test Case Review Defects.”
5. QA engineer updates the test cases based on these **formal defects**.
6. **Outcome:** Verified, standard-compliant, complete test suite.

**Purpose:** ✅ Formal defect detection  
 ✅ Technical correctness & compliance  
 ✅ Documented results

### Key Real-World Differences You’ll Notice

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



| **Aspect** | **Walkthrough** | **Technical Review** |
| --- | --- | --- |
| **Who leads?** | The author of the artifact (tester, dev, BA) | A trained moderator (not necessarily the author) |
| **Prep work?** | No — feedback is spontaneous | Yes — reviewers prepare notes in advance |
| **Focus** | Understanding & brainstorming | Conformance to standards, completeness, defect finding |
| **Output** | Informal suggestions | Formal defect list in a tool |
| **Tone** | Collaborative “show and tell” | Audit-like, standards-focused |
| **Example in testing** | Demoing test cases for clarity | Checking test cases against the SRS and QA templates |

💡 **Shortcut memory hook:**

* **Walkthrough** → *“Show me what you’ve got, let’s talk about it.”* (Brainstorm)
* **Technical Review** → *“Let’s inspect and mark every flaw against the checklist.”* (Inspection)