

## Inspection:

1.

Discuss the purposes, basic course of events, and alternate path of inspection review technique.

### Answer:

Purpose: To run a moderated inspection meeting.

Name	Inspection meeting script
Entry Criteria	A moderator must be selected, as well as team of 3 to 10 people. A work product must be selected, and each team member has read it individually and identified all wording that must be changed or clarified before he or she will approve the work product. A unique version number has been assigned to the work product.
Basic Course of Events	<ol style="list-style-type: none"><li>1. <b>Preparation.</b> The moderator distributes a printed version of the work product (with line numbers) to each inspector, along with a checklist to aid in the review. Each inspector reads the work product and identifies any defects to be brought up at the meeting.</li><li>2. <b>Overview.</b> The inspection meeting begins. The moderator verifies that each team member is prepared.</li><li>3. <b>Page-by-page review.</b> The moderator runs through the work product page by page. Inspectors indicate where there are defects. Each defect is either resolved or left as an open issue. The moderator adds each defect to the inspection log.</li><li>4. <b>Rework.</b> The author repairs the defects identified in the inspection meeting.</li><li>5. <b>Follow-up.</b> Inspection team members verify that the defects were repaired.</li><li>6. <b>Approval.</b> The inspection team approves the work product.</li></ol>
Alternative Paths	<ol style="list-style-type: none"><li>1. During Step 2, if any team member has not read the work product, then the inspection is halted. The meeting is rescheduled and the script returns to step 1.</li><li>2. During Step 4, if an inspection team member discovers additional defects in the work product, then the moderator calls another meeting and the process returns to step 1.</li></ol>
Exit Criteria	The work product has been approved.

### Basic Course of Events (Main Steps)

1. **Prepare** 📄 – The moderator gives the work and checklist to the team. Each member reads and notes defects.
2. **Start Meeting** 👥 – The moderator checks if everyone is ready before starting.
3. **Review Page by Page** 📖 – The team reviews defects one by one. The moderator logs them.
4. **Fix Issues** 🔧 – The author corrects the defects found in the review.
5. **Verify Fixes** ✅ – The team checks if everything is fixed properly.
6. **Final Approval** 👍 – If all defects are fixed, the work is approved.

### Alternative Paths (What If Scenarios)

- 🚨 If someone hasn't read the work (Step 2) → Stop! Go back to Step 1.
- 🚨 If new defects are found during fixing (Step 4) → Call another meeting & restart from Step 1.

### Memory Trick 🎯

"Prepare → Meet → Review → Fix → Verify → (↓)rove"

(PM-RFVA: "Please Make Reviews Faster, Verify Approvals!")

Techniques	Definition	Usage	Applicable Scenarios
Inspection	A work product is selected for review and a team is gathered for an inspection meeting to review the work product.	Inspections are often conducted during the early stages of a software project when documents like requirements specifications or design documents are being developed	- Critical work products like <b>Software Requirements Specifications (SRS)</b> and <b>Test Plans</b> where consensus is crucial.
Deskchecks	A deskcheck is a simple review in which the author of a work product distributes it to one or more reviewers.	There are times when a full inspection is neither necessary or useful.	- Low-stakes documents like <b>vision and scope documents</b> or <b>test scripts</b> that don't require team approval.
Walkthroughs	The author presents the work product to the audience in a manner that makes sense.	When the author of work product needs to take into account the perspective of someone who does not have the technical expertise.	- Documents requiring feedback from non-technical stakeholders, such as <b>use cases</b> or <b>design specifications</b> .
Code Reviews	A code review is a special kind of inspections which the team examines a sample of code and fixes any defects in it.	By systematically reviewing code before it is integrated into the codebase, teams can maintain code quality standards and prevent the introduction of substandard or vulnerable code.	- Complex, critical, or risky code portions, such as new <b>algorithms</b> or <b>core system modules</b> , especially in high-stakes projects.
Pair Programming	Pair programming is a software technique	Instead of waiting until after the code is	- Agile teams and environments needing

	development where two programmers work together on a single computer, collaborating on the same piece of code.	written to review it, issues can be addressed in real-time.	<b>high-quality, maintainable code</b> and <b>cross-training</b> among developers
--	--	---	---

#### Summer 24:

Suppose you are leading the development of an all-in-one travel management platform with moderate complexity designed to simplify the vacation planning process as a project manager of a team composed of individuals with a mix of experience levels on an ongoing project. The project has relatively simple complexity when handling inputs, outputs, and inquiries. While the core interactions within the system are uncomplicated, the challenges primarily arise in managing the high complexity of internal and external data files and their integration into the system. The platform allows users to enter their preferences, which include dates, locations, and the number of travelers to choose from for available travel services. Users provide personal details during the booking process, and they can also request cancellations or provide feedback after completing their trips. Users can ask the system to view available flights based on their criteria. The system maintains a user database that stores profiles, preferences, and travel histories. Once a user completes a booking, they receive an in-app confirmation for flights. The system generates a detailed trip schedule, including flight details, accommodation check-in/check-out times, and scheduled tour activities. After successful payment, users receive a receipt summarizing the amount, transaction details, and applicable booking reference numbers. The system also maintains a comprehensive log of all booking details for flights, rentals, and tours and keeps track of payment transaction records, including amounts, payment methods, and transaction statuses in the system directory. Users receive real-time notifications (via email or mobile alerts) in case of any changes to their booking status, such as flight delays, cancellations, or room availability. The platform interfaces with external airline, vacation rental, and tour operator databases to retrieve real-time availability and pricing. Furthermore, users can check the status of any bookings (current or past bookings) based on the booking date. Additionally, it integrates with a third-party secure payment gateway to process transactions and handles all types of payments. Now, your team is responsible for handling a combination of strict, well-defined requirements and more flexible, evolving ones. All the technical complexity factors from F1 to F6 are *moderate*; F8 and F9 are *average*, and the remaining technical complexity factors are *significant*. The average line of code required per function point is 128.

**Compare among inspection vs walkthrough review methods with respect to selecting for given scenario.**

Answer:

#### **Scenario Analysis: Travel Management Platform**

### Scenario Context:

1. **Moderate complexity system** with integrated internal and external data sources.
  2. Needs clarity in workflows (e.g., booking, payment, user preferences).
  3. Includes various stakeholders (e.g., users, developers, QA, travel operators).
  4. Involves critical integration points requiring defect-free design (e.g., external APIs, secure payments).
- 

### Which Review Method to Use?

#### Inspection:

- **When to use:**
  - For the **technical documentation** (e.g., system architecture, API design specifications) to ensure defect-free integration points.
  - Prevents expensive rework by detecting flaws in early design or technical documents.
- **Justification:** The platform interfaces with external APIs and handles sensitive payment information. Inspections can ensure no defects exist in these critical components.

#### Walkthrough:

- **When to use:**
    - For **requirements or use case validation** to ensure all stakeholders (technical and non-technical) understand the platform's workflows.
    - Helps gain alignment on user-centric features like booking flows and notification systems.
  - **Justification:** Non-technical stakeholders (e.g., travel operators or end-users) need to understand how the system addresses user preferences and trip management.
- 

### Recommendation for the Scenario:

- **Use Inspection** for **technical documentation** (e.g., integration with external APIs, payment gateway design).
- **Use Walkthrough** for **requirements and user workflows** to ensure stakeholder understanding and validation of use cases.

This dual approach balances technical defect prevention and stakeholder feedback.

## Spring 24:

“Redefine the landscape of education by providing a dynamic learning environment to every student” - with the above vision, ‘X University’ is building a learning platform for both online and combined in-person and online classes. The learning platform is designed to accommodate diverse features, serving various stakeholders such as students, faculty, and administrators. When a new student registers for courses, the system records their personal information, including name, address, contact details, program of study, academic records, and enrollment history in the database. After a student successfully registers for courses, the system generates a confirmation email containing their course plan. Students can submit assignments through the system by attaching files or providing links to external resources. The details about assignments, including assignment types, due dates, submission requirements, grading rubrics, and student submissions are recorded in assignment directories. Faculty members can grade different assessments such as assignments, quizzes, and exams submitted by students. At the end of each semester, students can complete course evaluations to provide feedback on course content. Students can access their grade reports through the system, displaying their grades for individual assessments. Upon request, students can generate official academic transcripts through the system. The course repository stores information about all courses offered within the university, including course details, syllabi, materials, counseling time, and instructor profiles. The system lets prospective students check if a specific course is offered next semester, including any prerequisites and whether there are still open spots. Students can see their course teachers' counseling time through the system. The learning platform integrates with the university's independent library system to provide access to course materials, e-books, journals, and research databases. Suppose there is simple complexity in inputs, outputs, and inquiries and high complexity in external files. The degree of influence for each of the components in technical complexity factors is Significant. Overall, this project has a high level of complexity with a large team size by considering all sets of parameters.

Considering the nature of the work product for the above project scenario, which review technique— would be more appropriate? Provide the rationale for choosing the review technique.

### Answer:

The **Inspection** review technique is the best choice for this project. Here's why:

1. **High Complexity:** The platform involves complex systems, like student records, course evaluations, library integration, and transcript generation. Inspections are ideal for identifying and fixing defects in such critical systems.
2. **Critical Documents:** Important documents, like system designs, requirements, and test plans, need a detailed review to ensure accuracy. Inspections focus on catching errors in these early stages.

3. **Large Team:** With a big team, inspections help ensure everyone agrees on the system's design and requirements. This prevents misunderstandings later.
4. **Error Prevention:** Mistakes in this platform can lead to big problems, like incorrect grades or issues with transcripts. Inspections catch these errors early, saving time and cost.

### **Here's why other techniques aren't the best fit for this project:**

#### **Deskchecks:**

- Too simple and informal for such a complex system.
- Doesn't provide the detailed review or teamwork needed for critical components.

#### **Walkthroughs:**

- Good for early feedback on designs or ideas but not detailed enough to catch serious issues.

#### **Code Reviews:**

- Useful during coding but won't cover the big-picture designs or system-level plans.

#### **Pair Programming:**

- Works for small tasks but isn't suitable for reviewing large and complex systems like this platform.

#### **Fall 23:**

You are leading an intermediate project to develop a comprehensive Medical Management System named "HealthTrack" for healthcare professionals of Bangladesh. You have a medium sized team consisting of both senior developers and interns. The system aims to streamline and enhance various functions related to patient records, medical inventory, and transactions within the healthcare facility. Breaking down the scenario, healthcare professional can add new patient records to the system, the system allows users for updating existing patient records and enabling users to add new medical items to the inventory. Healthcare professionals can inquire about a patient's medical history and can check the availability of specific medical items. The system generates different types of reports such as patient summary report, medical inventory report and transaction receipts. The patient record will be stored in the database. Patient record includes patient name, gender, date of birth and appointment date. The system maintains a file for tracking medical items where each medical item consists of information like item name, buy date and expiry date. For generating transaction receipts, the Transaction Track table in the database records transaction details, including transaction type, received amounts, and transaction date. The system interfaces with a government-regulated Health Statistics database to retrieve and update public health data. In addition, the system interfaces with a third-party

Telemedicine Platform to integrate remote consultation records. Assume that moderate complexity in inputs, outputs, inquiries, and high complexity in external files.

Provide a comparative analysis among two of the following review techniques: Code Reviews, and Pair Programming.

**Answer:**

Here's a comparative analysis of **Code Reviews** and **Pair Programming** for the "HealthTrack" Medical Management System project:

Aspect	Code Reviews	Pair Programming
Definition	A structured review process where a team inspects a sample of code for defects, readability, and maintainability.	Two developers work together in real-time at a single workstation to write, review, and refine code.
Main Objective	Identify and fix defects, ensure code quality, and improve maintainability.	Reduce defects during development and improve collaboration through continuous real-time review.
Formality	Formal: Involves a defined process, including preparation, review meetings, and defect tracking.	Informal: No formal process; developers collaborate directly while coding.
Collaboration	Requires a team of reviewers to analyze code after it is written.	Focuses on real-time collaboration between two developers during coding.
When to Use	Best for identifying defects in complex or critical components (e.g., integration with external databases).	Suitable for tasks that require iterative development and quick feedback (e.g., adding new features).
Scenarios	- Reviewing critical modules, such as <b>Transaction Track Table integration</b> and <b>external database interfaces</b> .	- Writing code for <b>patient record management</b> or <b>medical inventory features</b> , where fast iteration is helpful.
Knowledge Sharing	Allows team-wide knowledge transfer, as multiple reviewers provide insights.	Focuses on skill-building between senior and junior developers through close collaboration.



<b>Defect Detection</b>	Effective for systematically identifying hidden defects.	Catches defects immediately as code is being written.
<b>Productivity Impact</b>	Requires additional time after coding for review meetings and fixes.	Can increase productivity during development but requires two developers per task.

---

## Best Fit for the HealthTrack Project

- **Code Reviews:**
    - More suitable for **critical components** like the **interface with government Health Statistics** and **Telemedicine Platform integration**, where code correctness is vital.
    - Ensures a thorough, structured review process for complex modules.
  - **Pair Programming:**
    - Best for simpler, iterative tasks like **patient record updates** or **adding new medical items to the inventory**.
    - Promotes knowledge transfer between senior developers and interns, improving team skills.
- 

## Recommendation

For the HealthTrack project, use **Code Reviews** for high-complexity integrations and external files to ensure quality and reliability. Use **Pair Programming** for straightforward, iterative features where real-time collaboration and fast progress are beneficial.