

# Hackathon Brief

## **Use Case IV:** Deliverable & Sprint Sign-Off Hub

Automating Acceptance, Visualising Sprint Performance, and Capturing Client Approval — All in One Tool

### **Context**

Khonology's delivery teams commit to cadence based acceptance of scope and artifacts. In practice, though, sign-offs are scattered across emails and meetings, and "what exactly was accepted in which sprint" is hard to answer quickly. We also want a clear, repeatable way to show clients how the team performed during the sprints leading up to a deliverable without stitching together multiple tools.

This use case builds a single internal hub to: 1 create deliverable records, 2 link them to one or more sprints, 3 generate a sign-off report with sprint performance visuals, and 4 let client users review and approve directly inside the tool. Sign-off metadata and the final report are saved in the tool for audit and re-use.

### **Business Objective**

Build a self contained web application that:

1. Lets delivery teams create deliverables (title, description, definition of done, related artifacts/links).
2. Lets the team create and manage sprints (dates, scope/points, outcomes, test & quality metrics) inside the tool.
3. Associates one deliverable with multiple sprints (e.g., a feature delivered over 2–3 sprints).
4. Generates a sign-off report that summarises the deliverable and visualises sprint performance (velocity, scope completion, defects, test pass rate, burndown/burnup, etc.).
5. Presents the report in a client review page, including Approve / Change Request actions.
6. Captures the client's digital approval (date/time, identity, optional comment) and saves the signed report in the tool.
7. Maintains an audit trail of submissions, reminders, decisions, and report versions.

**Core Concept:** One Flow from Delivery ,Evidence, Approval

Everything lives inside the tool:

- Deliverable Register
  - Create deliverable entries with Definition of Done (DoD) and attach evidence (demo link, repo, test summary, user guide).
  - Map the deliverable to N sprints that contributed to it.
- Sprint Performance Workspace
  - Record sprint details (start/end, committed vs completed points, carried-over items).
  - Capture quality signals: test pass rate, escaped defects, defect severity mix, code review completion, documentation status.
  - Add notes on risks, mitigations, scope changes.
- Client Review & Approval
  - Generate an on-screen report combining the deliverable summary and visuals for the linked sprints.
  - Client user reviews in the tool, leaves comments, and Approves or Requests Changes.
  - Approved reports are sealed and archived; change requests reopen the submission with a clear action list.

No external files, formats, or spreadsheets are required for this flow; data entry, visuals, report generation, and approvals all happen within the application.

## **System Design**

### Key Screens & Flows

1. Deliverable Setup
  - Create deliverable; define DoD; attach evidence links.
  - Select the contributing sprint(s) from a list.
2. Sprint Console
  - Create/maintain sprints with dates and scope.
  - Enter outcomes: committed vs completed, carried-over, defects opened/closed, pass rate, coverage snapshot, UAT notes.
  - Optional free-text notes: risks, blockers, and decisions.
3. Performance Visuals (auto-generated from the sprint data you entered in the tool)
  - Velocity trend (per sprint).
  - Committed vs Completed bars.

- Burndown/Burnup within sprint.
  - Defect curve and severity mix.
  - Test pass rate / coverage snapshot.
  - Scope change indicators (added/removed during sprint).
4. Sign-Off Report Builder
    - Compose the report: deliverable header, DoD checklist, artifact list, linked sprint visuals, known limitations, next steps.
    - Preview the full report in the tool.
  5. Client Review Page
    - Secure review link for the client user.
    - Actions: Approve (with optional comment) or Request Changes (with mandatory comment).
    - On approval: the system stamps date/time & identity, stores the signed report and marks the deliverable Accepted.
    - On change request: the system records reasons, re-opens the deliverable, and notifies the team.
  6. Reminders & Escalation
    - Automatic reminders for pending client approvals.
    - Optional escalation after configurable timeouts.
  7. Repository & Audit
    - Repository of all signed reports and their history (who submitted, who viewed, who approved/declined, when).
    - Search by project, sprint, deliverable, or timeframe.

### **Wildcard Challenge: Release Readiness Gate**

Implement a Release Readiness indicator that blocks submission if basic acceptance hygiene is missing—e.g., no DoD checklist, missing sprint outcomes, or absent test evidence. The gate should:

- Evaluate completion of required fields/evidence.
- Display a single readiness status (Green/Amber/Red) with reasons.
- Prevent sending to the client until Amber/Red items are resolved or explicitly acknowledged by an internal approver.

Use of an AI component to further improve your solution.

## **Deliverables**

- A working, interactive tool showing the end-to-end flow:
  - Creating deliverables and sprints inside the tool.
  - Linking one deliverable to multiple sprints.
  - Auto-generated sprint performance visuals (velocity, burndown/burnup, defects, test pass rate, etc.).
  - Building a sign-off report and presenting it on a client review page.
  - Capturing client approval and saving the signed report in the tool.x
- Dashboard showing deliverables by status (Draft, Submitted, Approved, Change Requested) and average sign-off time.
- Audit trail view for at least one deliverable.