Engineering HUB Playbook

Standardised delivery for data centres (Engineering + Client Fit-Out)
Built in Microsoft 365 / SharePoint Online (No new software)

Contents

1. Executive Summary

Engineering HUB is a SharePoint Online "Playbook" for how your teams deliver data centre work - built inside your existing Microsoft 365 tenant. It makes delivery more repeatable and predictable by giving everyone one place to follow the same agreed way of working and always use the latest approved guidance.

It is implemented as a **SharePoint Online site** where your existing **procedures, templates, and checklists** are stored in one controlled place and organised using simple categories (metadata) so people can find the right item quickly.

It supports the delivery flow from commercial negotiation and bid stage through design, delivery, testing/commissioning, and handover to operations.

Because it runs inside your existing Microsoft 365 tenant:

- **No new platform** is introduced.
- **No third-party tools** are required.
- It uses your existing Microsoft 365 security and compliance settings (access control, permissions, retention, audit, etc.) - no separate security model is created.

The Engineering HUB Playbook provides one controlled place for:

- **How we work:** procedures, rules, stage gates, and clear "ready to proceed" criteria.
- **What we use:** standard templates and checklists (quality checks, gate packs, testing/commissioning packs, handover packs) plus reference examples - so teams reuse approved content instead of recreating it.
- **Who owns what:** a named owner for each page/template/checklist who keeps it current; optional approval before publishing where needed.
- **What changed and when:** version history and an audit trail, so teams can trust they are using the latest version.

Strict boundary:

Engineering HUB is **not** the project document system. Issued deliverables and client submissions remain in the Project CDE (the system used for formal submissions and issued documents - for example ACC/BIM360/Dalux/Aconex or a client CDE). The HUB holds the controlled internal Playbook, templates, and checklists that guide the work.

Optional add-ons (can be enabled later):

- Microsoft Planner (Premium) (formerly Project for the web) for standard plan templates (copied per project), with key tasks linking to the right Playbook page and checklist/template.
- Portfolio Roadmap (optional) for a roll-up across multiple projects (milestones, gate status, key risks).

   These are optional - the Playbook does not depend on them.

2. Purpose of the Engineering HUB

The purpose of the Engineering HUB is to make delivery more **repeatable and predictable** by ensuring everyone works from the same controlled **execution Playbook**.

The Engineering HUB Playbook provides one standard set of:

- Procedures (ways of working)
- Templates and checklists
- Stage gates and clear "ready to proceed" criteria
- QA/QC evidence expectations (what must be checked, and what proof is required)
- Reference examples / typicals (guidance only)
- Shared registers used across teams (for example: standards, approved materials and equipment (AME), software/tools, decisions, lessons learned, and feedback)

A key principle is **standardisation and reuse**: the HUB organises and governs what teams already use, reduces duplication, and supports repeatable execution - instead of allowing teams to keep creating new local versions of the same content.

3. Why the HUB Is Needed

Most organisations already have useful content such as procedures, templates, and checklists. The problem is how this content exists in practice:

- It is **spread across many places** (shared drives, different SharePoint sites, email threads, personal folders/notes, and sometimes project CDE areas that are not suitable for a company-wide Playbook).
- There is **version confusion** (different teams use different versions of the same template).
- **QA/QC expectations vary** by project, discipline, or individual, which leads to inconsistent quality and rework.
- **Ownership is unclear** (who maintains each procedure or checklist, who approves changes, and when it should be reviewed).
- It is **hard to find quickly** under delivery pressure.
- Reuse usually means **copying files**, which creates local variants and breaks standardisation.

Engineering HUB solves this by creating **one controlled Playbook**: one place for approved procedures, templates, and checklists, with clear readiness criteria, defined ownership, and version control.

Optional (visibility and portfolio management):
Where helpful, Engineering HUB can be paired with Microsoft Planner (Premium) (standard plan templates copied per project) and Portfolio Roadmap. This helps PMs run projects in a consistent way, and gives PMO and senior managers **one place** to review milestones, gate status, and key risks across multiple projects (through the portfolio view).

4. What Engineering HUB Is and Is Not

4.1 What it is

Engineering HUB is:

- A **Microsoft 365 / SharePoint Online Playbook** for how work is executed and governed.
- A **single source of truth for approved internal working content**, including:
  - **Procedures (ways of working)**

- o **Templates and checklists**
- o **Stage-gate readiness criteria** and **QA/QC evidence expectations**
- o **Reference examples / typicals** (guidance only)
- o **Controlled registers** (for example: standards, approved materials and equipment (AME), software/tools, decisions, lessons learned, feedback)

It is also a governed system with:

- Clear roles: **Accountable Owner + Responsible Editor** (and **Approver** where approvals are enabled)
- **Version history and traceability** (a clear record of what changed and when)
- **Publishing control:** either immediate availability or approval required
- **Optional review dates:** any HUB item (page, template, checklist, register entry) can have a review date, but it is not mandatory

4.2 What it is not (CDE boundary)

Engineering HUB is not:

- A new software platform
- A replacement for the **Project CDE**
- A storage location for **issued project deliverables** or **formal submissions**
- A design tool (for example Revit, AutoCAD, ETAP, EPLAN)
- A replacement for CDE workflows (it coexists with **ACC/BIM360/Dalux/Aconex** workflows)
- Dependent on Teams, Planner (Premium), Portfolio Roadmap, Power Automate, or Copilot (these are optional add-ons)

Boundary principle (simple and strict):
- **Engineering HUB = how we work** (Playbook + governance + reusable working content)
- **Project CDE = what we deliver** (issued deliverables + formal submissions + external workflows)

5. Engineering HUB Modules (Start with One; Expand Later)

Engineering HUB is delivered as a set of **modules**. Most organisations start with **one module**, prove value quickly, and then expand - **without rebuilding** the foundation.

5.1 One foundation reused across all modules (important)

All modules reuse the same governance and structure, which means:
- Content is **created once**, **organised once** using simple **categories (metadata)**, and **owned once**.
- The same content is then **reused and surfaced** across the HUB where needed - **without copying files** or creating parallel versions.
- Shared registers act as control points across modules (for example: standards, **approved materials and equipment (AME)**, software/tools, decisions, lessons learned, feedback).

See the demo site in Figure 1. It shows the DC Design HUB home page (core module) with the main navigation and discipline entry points to controlled content.

Figure 1 – Example Home Page: DC Design HUB (core module of the Engineering HUB)

5.2 Available modules (typical)

Module	Module name	Purpose	Typical Playbook content (examples)	Typical owners/users
A	**DC Design HUB (recommended start)**	Make design execution consistent across disciplines and systems	Procedures, deliverables-by-stage definitions, templates, QA/QC checklists, reference typicals, discipline/system pages	Engineering leadership, Design Manager, discipline leads, engineers
B	**Delivery & Commissioning HUB** (incl. Fit-Out / White Space playbooks)	Make site delivery, testing and handover repeatable	Readiness gates, inspection/testing templates, commissioning checklists, handover pack templates, lessons learned	Delivery/Construction, Commissioning, QA/QC
C	**PMO / Project Controls HUB** (Planner (Premium) + Portfolio Roadmap — optional)	Standardise governance and reporting across projects	RAID packs, meeting templates, stage-gate packs, reporting templates, dashboard rules; **optional** Planner (Premium) plan templates; optional Portfolio Roadmap roll-ups	PMO, Project Managers, Project Controls

D	**Quality & HSE HUB**	Make QHSE ways of working consistent	Policies, procedures, audit checklists, incident/observation templates, training references	QHSE leadership, site teams
E	**Operations & Asset HUB**	Standardise operational readiness and handover expectations	Ops readiness checklists, handover rules, asset data templates, operational reference standards	Operations, Asset Management, FM
F	**Registers HUB (cross-team)**	Provide controlled registers used across teams and projects	Standards, AME, software/tools, decisions, lessons learned (plus optional reference libraries)	Governance, Document Control, SMEs
G	**Procurement & Contracts HUB**	Standardise procurement workflows and package governance	RFQ/RFP templates, scope templates, vendor package checklists, evaluation templates, procurement procedures	Procurement, Supply Chain, Engineering stakeholders

Note: The **DC Design HUB** is commonly the best starting point because it becomes reusable "core content" for many other modules. However, if the immediate priority is **Client Fit-Out / White Space delivery**, organisations often start with **Delivery & Commissioning HUB + PMO/Project Controls HUB**, then add DC Design HUB once the delivery playbook is stabilised.

6. How Content Works (Metadata + Ownership + Publishing Control)

Core logic: Every Engineering HUB item (a page, file, or register entry) is:

1. **Categorised** using simple **categories (metadata)**
2. **Owned** (a named person is responsible for keeping it correct and current)
3. **Published** under a clear control rule

This keeps content easy to find, properly governed, and reusable **without duplicate copies**.

6.1 Content types (generic across all modules)

Site Pages - the readable "front end"
Landing pages, guidance pages, discipline/system/stage pages, navigation, and pages that surface filtered views.

Document Libraries - controlled working files
Templates, checklists, internal guides/manuals, and reference examples/typicals.

- "Controlled" means **owned, versioned, and categorised**; approval can be enabled where needed.
- It does **not** mean issued project deliverables.

Registers (Lists) - structured records used across teams

For example: standards, approved materials and equipment (AME), software/tools, decisions, lessons learned, and feedback.

- Each register has its own **Owner** and simple handling rules (for example: accept, reject, request rework, assign to an SME).

6.2 Metadata (core + optional extensions)

Typical core metadata includes:

- **Discipline**
- **System / subsystem**
- **Stage** (for example RIBA 0–7 or internal stage gates)
- **Content type / document type**

Metadata can be extended where useful (for example: content scope, region/country, business unit, project type, internal flags) depending on how the organisation needs to **filter, report, and surface** content.

6.3 Publishing control (two modes)

By library, register, or site area, content can be configured as:

- **Immediate availability** (visible immediately, but still owned and versioned)
- **Approval required** (visible to general users only after approval)

6.4 Metadata-driven surfacing (no duplicate copies)

Once published, content is surfaced through:

- Discipline, system/subsystem, and stage pages
- Filtered library views and Highlighted Content web parts
- Consistent navigation patterns

Intent: avoid manual linking everywhere, avoid duplicate copies, prevent broken navigation, and remove "which version is correct?" confusion.

Examples from the demo site are shown below:

- See the demo site in **Figure 2**. It shows an example **Discipline Page (Electrical Power)** that surfaces system pages and discipline-wide procedures using metadata (no manual linking, no duplicate copies).
- See the demo site in **Figure 3**. It shows an example **System Page (LV Main Distribution)** that surfaces deliverable requirements and reference guidance by stage using metadata (no manual linking, no duplicate copies).

4. System Pages

Open the relevant Electrical Power system guidance page from the cards below. Each card represents one system (e.g., HV Utility Grid Interface and Substation, MV Distribution, LV Main Distribution, UPS systems, Switchgear and Panels, Earthing and Bonding, Lightning Protection, Power Distribution to IT).

5. Discipline-Wide Procedures

This section contains Electrical Power discipline-wide procedures: approved, step-by-step methods we use to develop and review Electrical Power design the same way in every project. Open a procedure to see:

- Steps (what to do)
- Checks (what to verify)
- Required inputs (what you need before you start)
- Expected outputs (what you must produce)

How documents appear in this section (hard rule)

A document is shown here only if it has all of these tags:

Figure 2 - Example Discipline Page: Electrical Power

Figure 3 - Example System Page: LV Main Distribution

6.5 Continuous improvement (high level, flexible)

Engineering HUB supports controlled improvement without forcing one rigid workflow:

- Improvement inputs are captured (feedback, register submissions, lessons learned, etc.).
- Each page, library, and register has an **Owner** who defines how inputs are handled (accept, reject, request rework, assign to an SME).
- Updates are implemented by the responsible person and approved where approvals are enabled.
- Version history remains traceable. **Review dates** can be used for any item, but they are optional.

7. Roles, Permissions & Governance

7.1 Microsoft 365 permissions (standard)

Engineering HUB uses standard SharePoint permission levels:

- **Owners** - full control
- **Members** - can create and edit content (as configured)
- **Visitors / Readers** - view-only access

User access and permission changes can be managed:

- Directly in SharePoint by a **HUB site Owner**, and/or
- Through your organisation's **Microsoft 365 tenant administration** controls.

7.2 Governance roles (content accountability)

For controlled Playbook content, Engineering HUB uses clear accountability roles:

- **Accountable Owner** - responsible for correctness, structure, and final decisions
- **Responsible Editor** - creates and updates the content on behalf of the owner
- **Approver** - checks and approves content before it becomes visible to general users (only where approvals are enabled)

8. Engineering HUB vs Project CDE vs Tools vs Project Controls

Engineering HUB works best when each system has a clear purpose:

- **Engineering HUB (Playbook):** internal procedures, templates, checklists, stage gates, and shared registers - the controlled "how we work" content.
- **Engineering tools:** create the work (models, drawings, calculations, specifications).
- **Project CDE:** store and control **issued deliverables** and **formal submissions** (the client workflow stays here).
- **Project controls tools:** plan and track delivery (schedule/tasks, milestones, progress reporting).

The HUB can include **links** to project deliverables, but the deliverables themselves remain in the **Project CDE**.

9. Typical Delivery Workflow (day-to-day use)

1. **Create the project from the master template (PM/PMO)**
   Create each new project by copying the master Planner (Premium) plan. The plan already contains the standard stages, gate milestones, and links to the right Engineering HUB Playbook items (procedures, templates, checklists).

2. **Set project-specific details (PM/PMO)**
   Set the start date (or key milestone dates), **assign a responsible person (task owner)** for each task, and add only minimal project notes (site constraints, client-specific items). Do not rewrite the standard method - the Playbook remains the standard.

3. **Do the work using the Playbook (Engineers/Leads)**
   Open the task → open the linked Playbook page → use the approved template/checklist → produce outputs in the engineering tools.

4. **Issue deliverables in the Project CDE (PM + Engineers + Document Control, if used)**
   Upload and manage **issued deliverables and formal submissions** in the Project CDE. Client review/approval and submission workflows stay in the CDE.

5. **Track progress and report (PM/PMO)**
   Update task status and milestones in Planner (Premium) / Project / P6. Use Roadmap/portfolio views if you need a roll-up across projects.

6. **Improve the Playbook in a controlled way (Playbook Owner + Editor + Approver, if enabled)**
   Feedback and lessons learned are logged. The **Playbook Owner** decides what changes are accepted. The **Editor** updates the content. If approvals are enabled, an **Approver** approves it before it becomes visible to everyone. Version history is kept automatically.

Operational rule:

Plan & track → Execute using the Playbook → Issue via the CDE → Improve the Playbook

- **Plan & track:** Planner (Premium) / Project / P6
- **Execute:** Engineering HUB Playbook (procedures + checklists) + engineering tools
- **Issue:** Project CDE (ACC/BIM360/Dalux/Aconex/client CDE)
- **Improve:** Engineering HUB (owned updates; approvals if enabled)

Workflow table

Step	What happens (plain English)	Who does it	Where it happens
1	Copy the master plan (stages, gates, Playbook links already included)	PM / PMO	Planner (Premium)
2	Set dates + assign task owners + add minimal project notes	PM / PMO	Planner (Premium)

3	Follow Playbook instructions and produce technical outputs	Engineers / Discipline Leads	Engineering HUB + engineering tools
4	Upload issued deliverables and run submission workflow	PM + Engineers + Document Control (if used)	Project CDE
5	Update status, milestones, and reporting	PM / PMO	Planner / Project / P6 (+ Roadmap/Portfolio if used)
6	Update the Playbook based on feedback (controlled change)	Playbook Owner + Editor + Approver (if enabled)	Engineering HUB

Figure 4 shows the optional portfolio view (Roadmap) where senior leads / PMO can see multiple fit-out projects in one place using the same gate milestones.

Figure 5 shows the execution view (Planner task) where each standard task includes links to the right Playbook page and the approved checklist/template, so people follow the same method and do not use old versions.
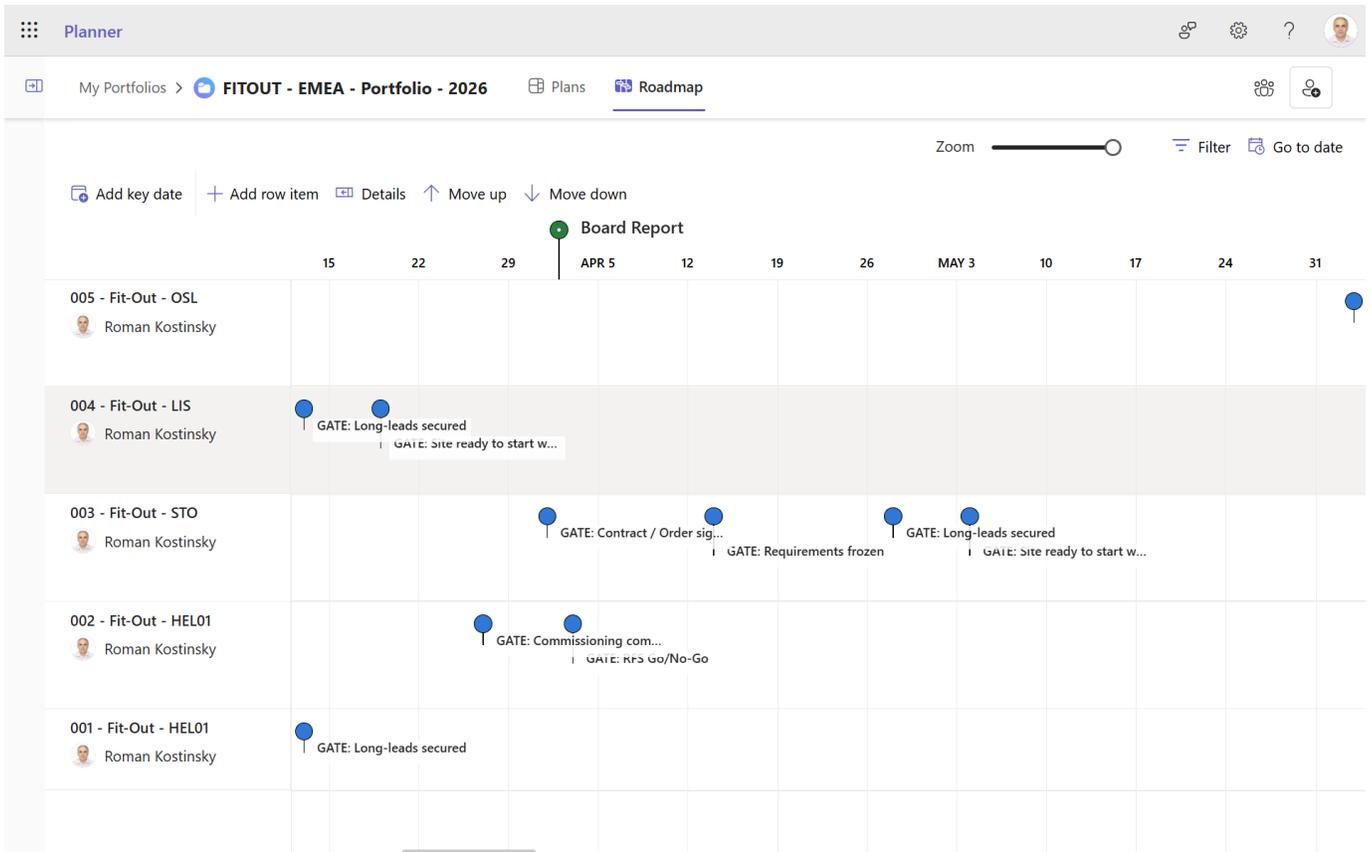

Figure 4 - Portfolio Roadmap

Figure 5 - Planner task view

10. Technical, Delivery & Financial Impact

10.1 Technical & delivery impact

- **Standard way of working** across the full lifecycle (RIBA 0–7 or your own stage gates)
- **A single current version** for internal playbooks: procedures, rules, templates, and checklists
- **Stronger QA/QC** using controlled checklists and clear readiness criteria at each gate
- **Fewer avoidable issues** (RFIs, coordination gaps, late fixes) because expectations are clear and consistent
- **Faster onboarding and training:** role-based guidance and quick-start pages inside the HUB
- **Controlled updates:** version history and audit trail, with approvals where enabled
- **Uses your existing Microsoft 365 security model** (no separate platform or security layer)
- **Best fit for repeatable delivery** (e.g., colocation and client fit-out) where standardisation reduces variation and rework

10.2 Financial impact (typical)

- **Less rework** (engineering and delivery) and fewer late design changes
- **Fewer RFIs and coordination loops**, reducing delay risk and management effort
- **Lower PMO effort** through reusable stage gates, checklists, and reporting patterns
- **Faster mobilisation** and less time spent reinventing project setup
- **No new software cost** (built on your existing Microsoft 365 baseline)

11. Conclusion

Engineering HUB makes the **Playbook ("how we work")** controlled, easy to find, and reusable - **without replacing the Project CDE**.

Implemented natively in **Microsoft 365 / SharePoint Online**, it provides a scalable foundation for consistent execution across engineering and delivery, with clear ownership, traceability, and approvals where needed - while keeping issued deliverables where they belong: **in the Project CDE**.

Contact / demo:

https://kostinsky.net/schedule-a-consultation/ | roman@kostinsky.net| +358 40 097 7001