Kostinsky Oy Ltd	Engineering HUB Summary (Simplified Version)
	Issue Date: 26/01/2026
	Revision: 59

Engineering HUB Summary

Unified Execution Governance for Engineering & Delivery
Microsoft 365 / SharePoint Online Framework (No New Software)

Contents

1. Executive Summary

Engineering HUB is an execution governance framework implemented natively inside Microsoft 365 (SharePoint Online). It standardises how engineering and delivery work is executed across the data centre lifecycle: concept → design → delivery → commissioning → handover → operational readiness.

Because it runs inside your existing Microsoft 365 tenant:

- No new platform is introduced.
- No third-party tools or plugins are used (100% Microsoft-native).
- It leverages your existing Microsoft 365 security and governance baseline (identity & access, permissions, retention/compliance controls, audit logging, OneDrive policies, etc.) — no separate security model is introduced.

The HUB creates one controlled place for:

- **How we work:** procedures, methods, rules, stage gates, readiness criteria.
- **Reusable working content:** templates, checklists, internal guides, reference examples/typicals, registers — so teams reuse approved content instead of recreating it.
- **Clear accountability:** defined ownership and update responsibility for pages, libraries, and registers, with approvals where enabled.
- **Traceability:** version history and an auditable change trail for controlled content.

Critical boundary rule: issued project deliverables and client submissions remain in the Project CDE (e.g., ACC/BIM360/Dalux/Aconex/Client CDE). Engineering HUB governs internal controlled guidance and reusable working content, not issued deliverables.

Optional Microsoft-native integrations (Teams, Outlook notifications, Microsoft Planner/Microsoft Project for the Web, Roadmap, Power Automate, Power BI, Copilot) can be enabled later — **the HUB is not dependent on them**.

2. Purpose of the Engineering HUB

The HUB makes delivery more predictable by ensuring everyone works from the same **controlled execution baseline**:

- procedures / ways of working
- templates and checklists
- stage gates and readiness criteria
- QA/QC evidence requirements and patterns
- reference examples/typicals (guidance only)
- shared registers (e.g., standards, AME, software, decisions, lessons learned, feedback)

A key principle is **standardization and reuse**: the HUB **organizes and governs what teams already use**, reduces duplication, and enables repeatable execution—rather than encouraging teams to keep creating "new local versions".

3. Why the HUB Is Needed

Most organizations already have useful content (procedures, templates, checklists). The problem is how it exists in practice:

- spread across multiple locations (drives, SharePoint sites, email, personal notes, legacy folders, and sometimes project CDE areas that are not suitable as a company-wide method repository)
- version chaos (different teams using different versions of the same template)
- QA/QC expectations varying by project, discipline, and individual
- unclear accountability (who owns the method, who approves changes, when it should be reviewed)
- hard to find quickly under delivery pressure
- difficult to reuse without copying and creating local variants

Typical impact:

- teams recreate work instead of reusing controlled guidance
- readiness criteria and QA/QC expectations become clear too late → rework and late corrections
- lessons learned are captured, but not systematically converted into improved ways of working

What the HUB changes: one governed environment for internal controlled guidance and reusable working content, with ownership, publishing control, and one approved source for "how work is executed".

4. What Engineering HUB Is and Is Not

4.1 What it is

Engineering HUB is:

- a **Microsoft 365 / SharePoint Online execution governance layer**
- a **single source of truth** for approved internal working content:
    - procedures / ways of working
    - templates and checklists
    - stage-gate readiness criteria and QA/QC evidence requirements

    - reference examples/typicals (guidance only)
    - controlled registers (standards, AME, software, decisions, lessons learned, feedback, etc.)
- a governed system with:
    - **Accountable Owner + Responsible Editor** (+ **Approver** where approvals are enabled)
    - version history and traceability
    - publishing control (immediate availability or approval required)
    - optional review discipline (Next Review Date is supported, but optional)

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 (Revit, AutoCAD, ETAP, EPLAN, etc.)
- an ACC/BIM360 workflow automation layer (it coexists with CDE workflows)
- dependent on Teams / Planner / Project / Roadmap / Power Automate/Copilot (these are optional integrations)

Boundary principle (simple and strict):

- **Engineering HUB = how we work** (methods + governance + reusable working content)
- **Project CDE = what we deliver** (issued deliverables + submissions + external workflow)

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

Engineering HUB is delivered as a **collection of modules**. Organizations typically start with **one (1) module**, validate value, and expand later **without rebuilding the foundation**.

5.1 One foundation reused across all modules (important)

All modules reuse the same **metadata + governance backbone**, which means:

- content is **created once**, **classified once**, and **owned once**
- then **surfaced wherever needed** across the HUB—without duplicating files or creating parallel versions
- registers act as shared control points across modules (standards, AME, software, decisions, lessons learned, etc.)

5.2 Available modules (typical)

Module	Module Name	Primary purpose	Typical HUB content (examples)	Typical owners/users
A	DC Design HUB (recommended start)	Standardize design execution across disciplines/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	Standardize delivery readiness & commissioning evidence	readiness gates, inspection/testing templates, commissioning checklists, handover pack templates, lessons learned	delivery/construction, commissioning, QA/QC
C	PMO / Project Controls HUB	Standardize governance & reporting	RAID, decision logs (or linked), reporting templates, stage-gate packs, meeting templates, dashboard conventions	PMO, project managers, project controls
D	Quality / HSE HUB	Controlled ways of working for QHSE	policies, procedures, audit checklists, incident/observation templates, training reference	QHSE leadership, site teams
E	Operations / Asset Management HUB	Standardize ops readiness & handover rules	ops readiness checklists, handover templates, operational standards (reference)	operations, asset management, FM
F	Registers HUB (cross-discipline)	Controlled registers used across projects/modules	standards, software, AME, decisions, lessons learned (+ optional reference libraries)	governance, doc control, SMEs
G	Procurement / Contracts HUB	Standardize procurement workflows & package governance	RFQ/RFP templates, scope templates, vendor package checklists, evaluation templates, procedures	procurement, supply chain, engineering stakeholders

See **the demo site in Figure 1** - Example Home Page (DC Design HUB, core module of the Engineering HUB): the primary entry point showing site navigation and discipline entry points to governed content.

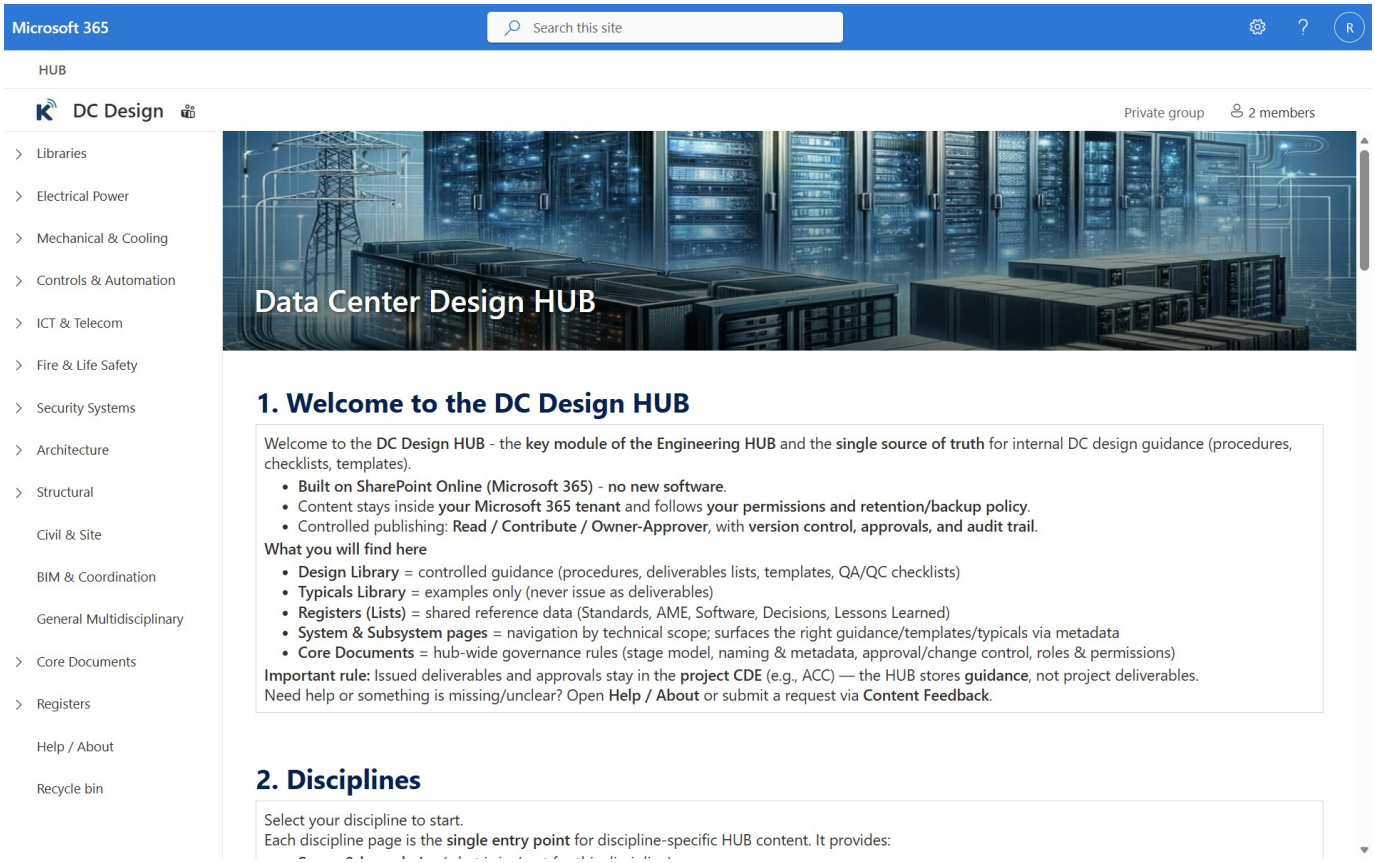

Figure 1 - Example Home Page (DC Design HUB, core module of the Engineering HUB): the primary entry point showing site navigation and discipline entry points to governed content.

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

Core logic: Every HUB item—**site page, file, or register entry**—is:

1. **classified** (metadata),
2. **owned** (accountability), and
3. **published** under a clear control rule.

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

6.1 Content types (generic across all modules)

- **Site Pages** — the readable "front end": landing pages, guidance pages, discipline/system/stage pages, curated navigation, and surfaced views.
- **Document Libraries** — controlled working files such as templates, checklists, internal guides/manuals, reference examples/typicals.
    - "Controlled" means: **owned, versioned, and classified**; **approval can be enabled** where needed.

    - It does **not** mean “issued project deliverables”.

- **Registers (Lists)** — structured records used across teams (e.g., standards, AME, software, decisions, lessons learned, feedback).
    - Each register has its **own Owner** and **its own handling rules** (accept/reject/rework/assign).

6.2 Metadata (core + optional extensions)

Typical core metadata includes:

- discipline
- system / subsystem
- stage (RIBA 0–7 or internal stage gates)
- content type / document type

Metadata can be **optionally extended** (e.g., **ContentScope**, region/country, business unit, project type, internal flags), depending on how the organization needs to filter, report, and surface content.

6.3 Publishing control (two modes)

By library/list/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/sub-system/stage pages
- filtered library views and Highlighted Content web parts
- consistent navigation patterns

Intent: avoid manual linking everywhere, duplicate copies, broken navigation, and “which version is correct?” confusion.

See **the demo site in Figure 2** - Example Discipline Page (Electrical Power): a single entry point that surfaces System Pages and Discipline-Wide Procedures using metadata (no manual linking, no duplicate copies).

See **the demo site in Figure 3** - Example System Page (LV Main Distribution): a system-specific guidance page that surfaces Deliverable Requirements (Definitions) and Reference Guidance (Typicals) by stage using metadata (no manual linking, no duplicate copies).

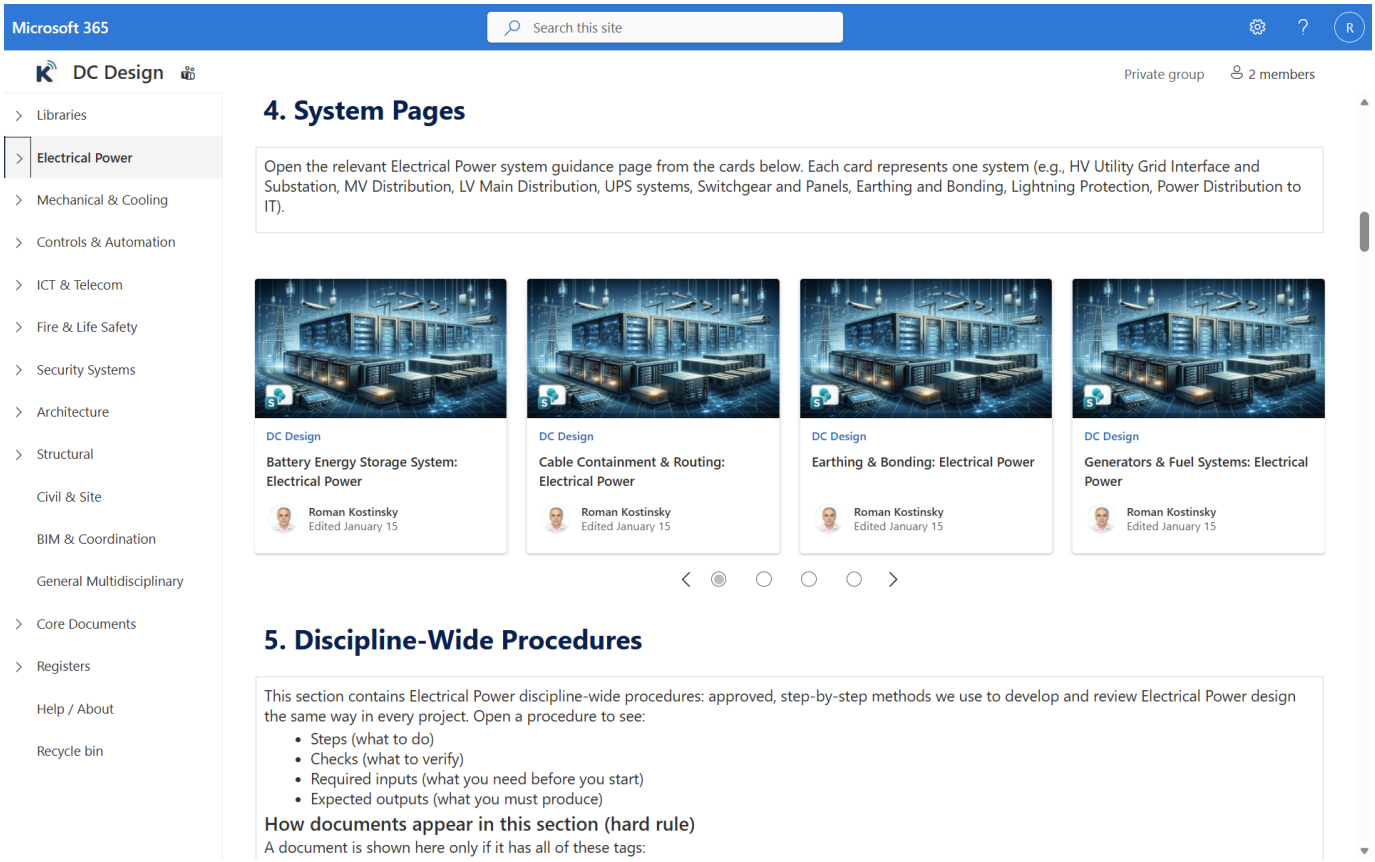

Figure 2 - Example Discipline Page (Electrical Power): a single entry point that surfaces System Pages and Discipline-Wide Procedures using metadata (no manual linking, no duplicate copies).

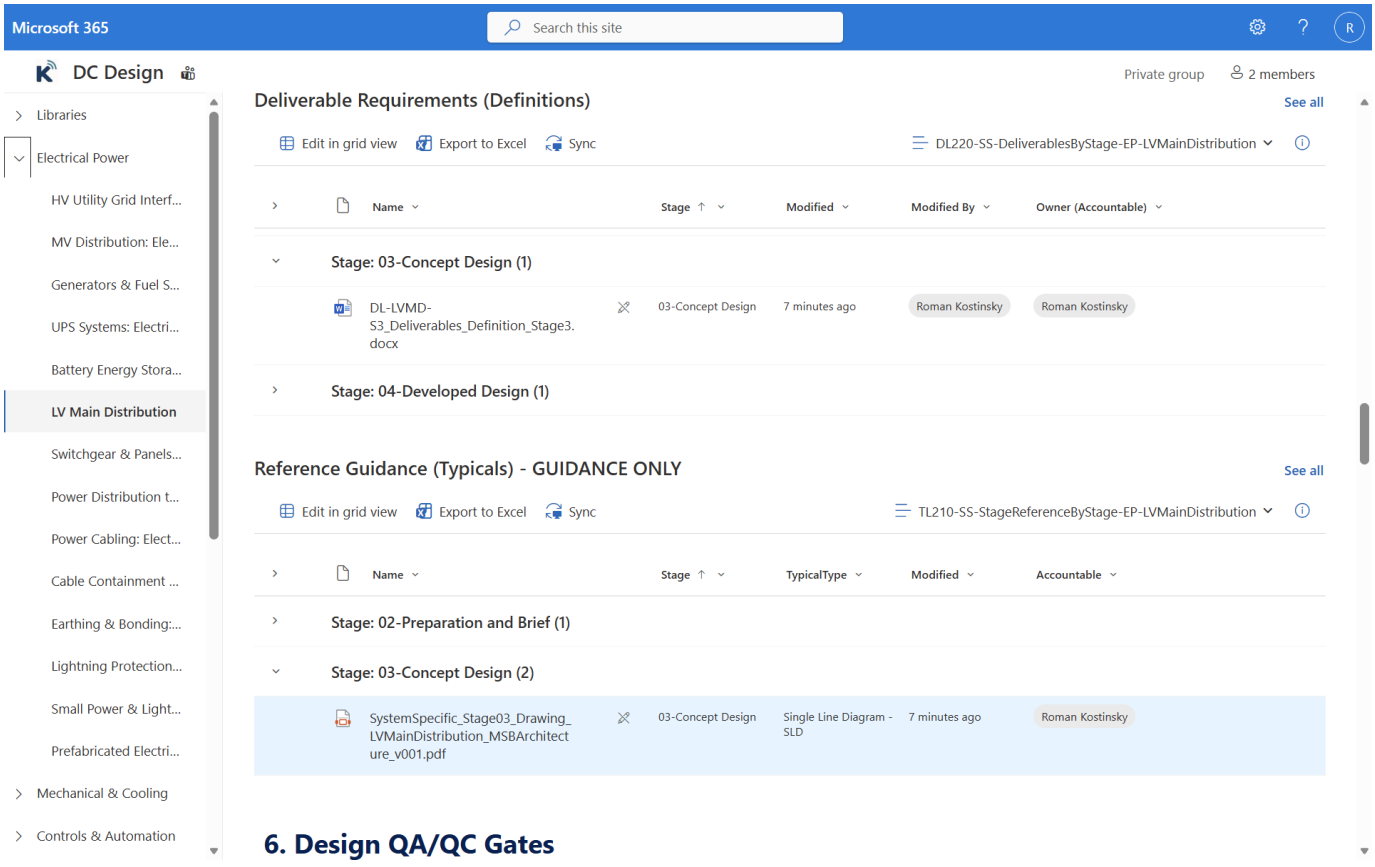

Figure 3 - Example System Page (LV Main Distribution): a system-specific guidance page that surfaces Deliverable Requirements (Definitions) and Reference Guidance (Typicals) by stage using metadata (no manual linking, no duplicate copies).

6.5 Continuous improvement (high level, flexible)

The HUB supports controlled improvement **without forcing one rigid workflow**:

- improvement inputs are captured (feedback, register submissions, lessons learned, etc.)
- each page/library/register has an **Owner** who defines how inputs are handled (accept/reject/request rework/assign SME)
- updates are implemented by the **Responsible** person and approved where approvals are enabled
- history remains traceable; **Next Review Date is supported but optional**

7. Roles, Permissions & Governance

7.1 Microsoft 365 permissions (standard)

- **Owners** — full control
- **Members** — contribute/edit (as configured)
- **Visitors/Readers** — view only

User access and permission changes can be handled:

- directly in SharePoint by a site Owner (HUB Owner), and/or
- via Microsoft 365 tenant administration controls.

7.2 Governance roles (content accountability)

- **Accountable Owner** — final authority for correctness and structure
- **Responsible Editor** — prepares and updates content
- **Approver** — validates before publication (where approvals are enabled)

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

Clear separation of purpose:

- **Engineering HUB:** internal method + governance + reusable working content
- **Engineering tools:** create the work (models, drawings, calculations)
- **Project CDE:** manage issued deliverables and submissions (client workflow stays there)
- **Project controls tools:** plan and track delivery (schedule/tasks, milestones, reporting)

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

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

1. **Plan from a standard template (PM/PMO)**
   Create the plan using a standard Microsoft Project for Web/Microsoft Planner template aligned to stage gates and disciplines. **Each standard task is pre-linked to the relevant HUB page** (procedure/template/checklist/readiness criteria).
2. **Adjust only what must vary (PM/PMO)**
   Update dates, milestones, assignments/resources, and limited project-specific notes. The execution method stays controlled by the HUB.
3. **Execute using HUB guidance (engineers/leads)**
   Open task → open linked HUB page → use the controlled template/checklist → produce outputs in engineering tools.
4. **Issue deliverables in the Project CDE (boundary)**
   Upload and manage issued deliverables in the Project CDE (client/submission workflow stays there).
5. **Track and report in project controls tools (PM/PMO)**
   Track progress in Project / Planner / P6 / Roadmaps. The HUB is not a scheduling tool—it is the execution standard.
6. **Improve under ownership (controlled updates)**
   Capture feedback and register inputs and improve HUB content under ownership (and approvals where enabled).

Operational rule:

Plan & track (project controls tools) → Execute (HUB guidance) → Create (engineering tools) → Issue (Project CDE).

10. Technical, Delivery & Financial Impact

10.1 Technical & delivery impact

- Standardized workflows across the lifecycle (RIBA 0–7 or equivalent stage gates)
- Single source of truth for internal governance and methods
- Stronger QA/QC via controlled checklists and readiness criteria
- Fewer RFIs, coordination errors, and late corrections
- Faster onboarding (one place to learn how work is executed)
- Traceable change control (versioning, audit trail; approvals where enabled)
- Microsoft-native security/compliance model and familiar user experience

- Particularly effective for repeatable work where standardization reduces variation and rework

10.2 Financial impact (typical)

- Reduced engineering rework and late design changes
- Fewer RFIs and coordination loops → fewer delays and less management overhead
- Lower PMO overhead through reusable governance patterns and templates
- Reduced training effort and faster mobilization
- No additional software platform cost (leverages existing Microsoft 365 baseline)

11. Conclusion

Engineering HUB makes “how we work” **controlled, findable, and reusable**—without replacing the Project CDE.

Implemented natively on Microsoft 365 / SharePoint Online, it provides a scalable foundation for consistent execution across engineering and delivery functions, with ownership, traceability, and optional approvals—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