

6-Month WIL Programme Plan Framework (Template)

For employers / SMEs to tailor per ICT role and complexity

How to use this template (read first):

- 1) Complete the header and role context.
- 2) Choose your complexity level (Low / Medium / High).
- 3) Populate the Month-by-Month plan using your real work tasks.
- 4) Use the Daily Activity Plan to allocate hours (not time-of-day).
- 5) Capture evidence weekly using your WIL Activity Record and portfolio.

Programme Header

Employer / Site	[Company name, site, city]
Compiled by	[Name, role, email/phone]
Job Role	[e.g., Junior Service Desk Analyst]
Role family	[e.g., IT Support, Software Dev, Data, Cyber]
Complexity level	[Low / Medium / High]
Programme dates	[Start date] to [End date] (6 months)
Delivery mode	[Onsite / Hybrid / Remote]
Supervisor / Mentor	[Name, role] (backup: [Name])
Tools / systems used	[Ticketing, repos, cloud tools, etc.]

1. Role Context (keep it practical)

In 5-8 bullets, describe the real work the learner will do in your business. This should be normal work tasks, not classroom training.

Example bullets (edit or replace):

- Handle and close routine support tickets (or equivalent work items).
- Work on small features / configurations under supervision.
- Follow your standard operating procedures and quality checks.
- Produce evidence of work completed (tickets, commits, documents, logs).

2. Daily Activity Plan (hours per day - you choose the times)

Use this as a default daily structure. Replace activities with role-specific equivalents. The goal is a balanced day that includes real work, supervision, quality checks, and evidence capture.

Activity type	Typical hours/day	What this looks like (examples)
1) Planning & alignment	0.5 - 1.0	Daily priority check-in; clarify expected outputs; check safety/compliance for the day.
2) Core work (delivery)	3.0 - 4.0	Primary role tasks (tickets, builds, configs, analysis, monitoring, etc.).
3) Quality & verification	1.0 - 1.5	Testing; peer/supervisor review; rework; checklist completion; validation against SOPs.
4) Support & learning-in-work	0.5 - 1.0	Shadowing; short demos; guided practice; reading runbooks; tool walkthroughs.
5) Evidence & reflection	0.5 - 1.0	Update WIL Activity Record; save artefacts/links; short reflection on what changed/was achieved.

3. Programme Structure (6 months)

Below is a simple month-by-month progression. Use it as a framework. For each month, describe: (a) the main work activities, (b) expected outputs, and (c) evidence to keep.

Month	Progression focus	Typical WIL activities (edit)	Expected outputs (examples)	Evidence (examples)
Month 1	Onboarding + routine tasks	Induction; tools access; simple tasks; shadowing; basic SOP adherence.	Completed setup; first closed work items; basic documentation.	Access checklist; first tickets/commits; screenshots; supervisor notes.
Month 2	Consistent routine delivery	Repeat core tasks; handle common exceptions; begin small improvements.	Stable throughput; fewer errors; escalations done correctly.	Ticket/issue history; work logs; QA checklists; peer review comments.
Month 3	Simple ownership	Own a small workstream/module queue; plan work	Delivered a small feature/config or equivalent;	Change records; commits; test evidence; short handover note.

Month	Progression focus	Typical WIL activities (edit)	Expected outputs (examples)	Evidence (examples)
		items; deliver to acceptance criteria.	documented handover.	
Month 4	Applied problem-solving	Diagnose causes; propose fixes; implement under review; strengthen quality discipline.	Resolved non-trivial issues; improved SOP/runbook; reduced repeat incidents.	Root-cause notes; before/after metrics; updated runbook; supervisor sign-off.
Month 5	Independent delivery (with check-ins)	Deliver end-to-end routine work with limited supervision; coordinate with stakeholders.	Predictable delivery; good stakeholder comms; self-checking quality.	Work items; comms samples; quality evidence; mentor verification.
Month 6	Consolidation + capstone	Capstone deliverable (role-appropriate); prepare portfolio; final assessment.	Capstone delivered; portfolio complete; readiness for next role level.	Portfolio index; PoE artefacts; final supervisor assessment; verification record.

4. Complexity Guidance (choose the relevant one and adapt the plan)

Complexity level	How to scale the plan	Example: what changes
Low complexity	Focus on high-volume, well-defined tasks. More repetition; simpler tools; tighter SOPs. Increase autonomy by reducing prompts over time.	Service desk L1: more tickets, more scripts/checklists, limited systems access.
Medium complexity	Mix routine tasks with defined problem-solving. Introduce small ownership earlier; add peer review and stronger quality checks.	Junior sysadmin / data analyst: routine tasks + investigations + simple automation.

Complexity level	How to scale the plan	Example: what changes
High complexity	Introduce deeper problem-solving and broader context. Capstone should be a meaningful work product; include risk/security and stakeholder alignment.	Software dev / cyber analyst: feature ownership, threat triage, CI/CD, change control, QA discipline.

5. Minimum Evidence & Verification (keep it simple)

Each week, ensure you have evidence of work completed and that a supervisor/mentor can verify it. Evidence can be system logs, ticket references, commits, documents, dashboards, runbooks, or sign-offs.

Weekly minimum (recommended):

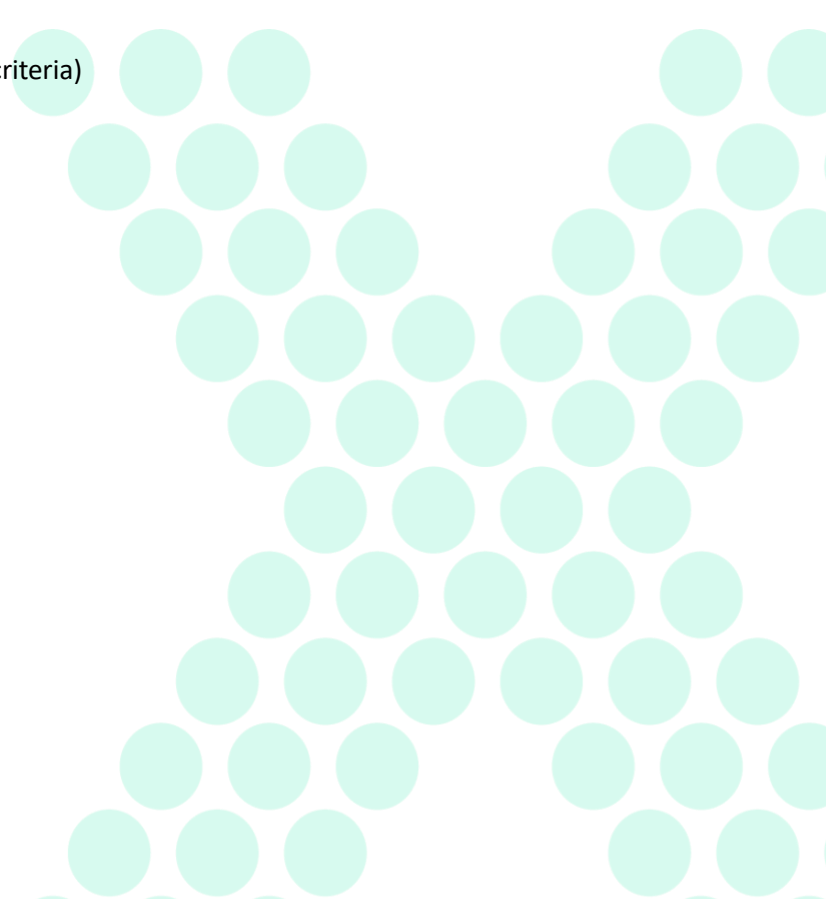
- 3-5 evidence items saved/linked (per learner).
- 1 short weekly summary in the WIL Activity Record (what was done, what changed, evidence links).
- Supervisor/mentor review at least monthly (and at mid-point and end).

6. Link to Skills Programme Mapping

This 6-month plan provides the 'what work will be done and when'. Use the Skills Programme Mapping document to link that work to skills, learning outcomes, and evidence/verification.

When you complete this plan, you should be able to copy/paste into the Mapping Template:

- Programme scope and purpose
- Duration and expected hours
- WIL activities (activity + description + proficiency criteria)
- Evidence captured and who verifies it



Appendix A: Quick Examples by Role Family (optional)

Use these as inspiration. Replace with your real tasks.

Role family	Core WIL activities (examples)
IT Support / Service Desk	Log/resolve incidents; user comms; device setup; password/access; escalation; runbook updates.
Software Development	Backlog tasks; code changes; tests; reviews; deployments; documentation; defect fixes.
Data / BI	Data cleaning; report/dashboard updates; basic SQL; data quality checks; documentation; stakeholder feedback loop.
Cyber / Security	Ticket triage; vulnerability scans; policy checks; alert review; evidence handling; incident runbooks.
Cloud / Infrastructure	User provisioning; monitoring; backups; routine config; change control; documentation; performance checks.

