Mindset & People

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

LEARN THROUGH DELIVERY

Shared purpose → **Collaborate** → **Deliver increment** → **Inspect and adapt**

01 Agile mindset, people and culture

Being agile and doing agile

- **Agility** is the ability to embrace and respond to change. **Being agile** means adaptive thinking, collaboration and learning; **doing agile** means applying practices such as boards, timeboxes and reviews.
- The **Agile Manifesto** values individuals and interactions, working software, customer collaboration and responding to change over processes and tools, comprehensive documentation, contract negotiation and following a plan. The latter still have value.
- The **Agile Onion** moves inward from processes and practices to principles, values and mindset. Visible ceremonies need supporting beliefs; principles explain why practices are useful.
- **Iterative** work refines understanding through repeated feedback. **Incremental** delivery adds usable capability in portions. **Empiricism** bases decisions on observation and experience; inspect results and adapt.

Behaviours and rich communication

- **Transparency** makes real status and problems visible. **Collaboration** builds shared goals and ownership. **Self-organization** lets teams decide how to work within agreed boundaries.
- **Rich communication** uses an effective, interactive channel with rapid feedback. Face-to-face discussion, shared visuals and demonstrations help resolve ambiguity; distributed teams can use video and shared boards.
- **Exploration** uses experiments and stakeholder feedback to discover better solutions. Learn through small, controlled steps rather than specifying every detail before learning is possible.
- An **information radiator** makes useful information easy to see without requesting a report. Keep boards current, understandable and accessible to the intended audience.

Safety, leadership and working agreements

- **Culture** consists of shared attitudes, values and goals. **Co-creation** involves affected people in shaping solutions and ways of working, strengthening understanding and adoption.
- **Psychological safety** supports speaking up: **inclusion** means belonging; **learner** safety supports questions and mistakes; **contributor** safety supports applying skills; **challenger** safety supports questioning the status quo.

Mindset & People / Governance & Roles

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

LEARN THROUGH DELIVERY

- **Shared purpose** → **Collaborate** → **Deliver increment** → **Inspect and adapt**

- **Team working agreements** are created together: decide communication habits, availability, decision rules and how to handle conflict. Revisit them as the team learns.
- **Delegation poker** helps a team discuss and agree decision authority. Clear delegation supports autonomy; leaders provide purpose, remove obstacles and model learning.
- **Agile sustainability** needs a maintainable pace and continuous improvement. Gain leadership support by connecting agility to business outcomes, value and risk; assess maturity and support people through organizational change.

Projects, products and business as usual

- A **project** is temporary and delivers agreed products to support change. **Business as usual (BAU)** is ongoing operation, service or maintenance, often performed by established teams. Both can use agile techniques.
- **PRINCE2 Agile** blends project governance with agile ways of working. PRINCE2 supplies direction and management; approaches such as Scrum and Kanban support delivery. Agile is not limited to IT.
- **Waterfall** organizes development into sequential phases with distinct goals. Agile uses short feedback cycles to refine requirements and deliver incrementally.
- **Scaling** coordinates multiple teams, dependencies and decisions around shared outcomes. Align terminology, product priorities and governance; inconsistent language and culture can obstruct the blend.

02 Principles, governance and responsibilities

Seven principles in an agile setting

- **Continued business justification** keeps the investment worthwhile as scope evolves. **Learn from experience** uses feedback, reviews and retrospectives throughout the work.
- **Defined roles, responsibilities and relationships** connect business, user and supplier interests. **Manage by stages** creates decision points for continued investment; stages can contain several releases or iterations.
- **Manage by exception** delegates authority using tolerances and escalates forecast breaches. **Focus on products** defines intended outputs and acceptance before planning activities.
- **Tailor to suit the project** adapts controls, information and working practices to context. Preserve the principles while adjusting emphasis and detail; use suitability assessments to guide tailoring.

Direction, management and delivery

- **Commissioning** establishes the business mandate and overall constraints. The **project board** directs and authorizes; the **project manager** manages the project day to day within delegated limits.
- **Delivery teams** organize specialist work within agreed work packages and constraints. Local autonomy works alongside accountability, visibility and escalation.

Governance & Roles

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

DELEGATED GOVERNANCE

- **Board** — Direct
- **Project manager** — Manage
- **Team** — Deliver
- **Tolerances** — Escalate

- The **organization** practice establishes accountability and responsibility. **Project assurance** independently checks business, user and supplier interests; **project support** assists with administration, information and controls.
- The integrated elements are **principles, people, practices, processes and project context**. People include those doing the work and those affected; practices address recurring management concerns; processes organize lifecycle activity.

Business, user and supplier interests

- The **executive** owns the business case and balances business, user and supplier demands, keeping the project focused on value for money.
- The **senior user** represents user needs, acceptance and expected benefits. The **chief product owner** aligns user priorities and product owners across the project.
- The **senior supplier** represents those designing, building or supplying products and safeguards technical integrity, feasibility and the availability of supplier resources.
- **Stakeholders** can affect or be affected by the project. Identify their needs, influence and engagement requirements; operational staff and external parties can matter even without formal project roles.

Product and delivery roles

- The **product owner** manages and orders the product backlog to maximize value, clarifies user needs and supports acceptance. A backlog role does not automatically confer project-board authority.
- The **team manager** coordinates delivery against agreed work packages. **Developers** create specialist products; **testers** help evaluate quality and expose defects throughout development.
- The **team coach** helps a team improve collaboration and delivery practices; the **agile coach** supports wider adoption and capability. In Scrum, the **Scrum Master** coaches, facilitates effectiveness and helps remove impediments.
- **Customer representatives and customer subject matter experts** clarify user needs and business detail. **Supplier representatives and supplier subject matter experts** contribute technical expertise, constraints and standards compliance.

Fixed targets and flexible delivery

- The **fix and flex** approach protects time and cost while allowing scope to vary. Agreed quality criteria can have tolerances; essential quality is protected. Risk and benefits remain subject to agreed governance.
- **Be on time and hit deadlines** supports confidence and dependency planning. **Protect the level of quality** preserves fitness for purpose: plan testing and verification into delivery.

Governance & Roles / Value & Planning

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

FROM VALUE TO USABLE WORK

- **Vision and benefits** → **Epics and stories** → **Prioritize** → **Plan increment**

- **Embrace change** by accommodating useful learning within boundaries. **Keep teams stable** to preserve knowledge, relationships and predictable capacity; adding people late can disrupt delivery.
- **Accept that the customer does not need everything**: prioritize valuable, essential capability and defer lower-value features. A smaller usable product can deliver benefits sooner.
- **Tolerance** is permitted variation around an objective. If a forecast exceeds delegated limits, escalate before the breach. Fixed time and cost make delivered scope and remaining benefits especially important signals.

03 Value, requirements and planning

Business case and product vision

- The **business case** justifies investment through expected benefits, costs and risks. Reassess desirability, viability and achievability as delivery reveals new information.
- An **output** is a delivered product; an **outcome** is the resulting change; a **benefit** is a measurable improvement. Delivering features does not by itself prove that benefits were realized.
- Define the **minimum viable product (MVP)** and its role in learning or benefit delivery. Identify when it is expected and how reduced scope could affect the business case; an MVP is not the entire viability argument.
- **Personas** express relevant user needs, goals and contexts to help articulate benefits. **Objectives and key results (OKRs)** connect a clear objective to measurable results.
- The **project product description** gives the overall product vision, purpose and acceptance expectations. A **project canvas** concisely aligns purpose, value and key project information; revisit it at decision points and closure.

Backlogs, epics and user stories

- A **backlog** is an ordered list of work or requirements. The **project backlog** expresses major project products, often as epics; **product backlogs** contain product requirements; a **sprint backlog** captures selected work and its delivery plan.
- An **epic** is a broad, high-level user story that needs decomposition. A **user story** expresses who needs what and why, commonly: As a [user], I want [capability], so that [benefit].
- **Refinement** adds detail, splits work, estimates effort and clarifies priority as understanding develops. Near-term items need more detail than distant items.
- **Acceptance criteria** specify testable conditions for a particular story or product. They become clearer through conversation; a story is a prompt for shared understanding, not a complete specification by itself.

Value & Planning

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

FROM VALUE TO USABLE WORK

Vision and benefits → **Epics and stories** → **Prioritize** → **Plan increment**

MoSCoW and trading scope

- **Must have** is essential for a viable delivery; **Should have** is important but can be deferred with difficulty; **Could have** is desirable; **Won't have this time** is explicitly outside the current commitment.
- Apply **MoSCoW** to a stated horizon such as a release or timebox. If everything is a Must, there is little room to absorb uncertainty; identify a useful minimum and genuine flexibility.
- **Trading or swapping** replaces lower-priority work with new work of comparable effort within agreed constraints. Check dependencies, quality, risks and benefits before accepting the trade.
- **Scope** concerns what capability is delivered; **quality** concerns its required characteristics and fitness for purpose. Distinguish functional features from non-functional needs such as performance and security when prioritizing.

Product planning and delivery horizons

- The **plans** practice defines products, their delivery approach, responsibilities, timing and resources. **Product-based planning** begins with required products and descriptions, then identifies dependencies and activities.
- Use **project, stage and team plans** at appropriate detail. High-level plans coordinate outcomes and dependencies; detailed team planning adapts through evidence rather than pretending all future work is certain.
- A **release map or release plan** shows intended increments, sequence and dependencies. Establish an initial view during initiation and refine it as learning changes priorities.
- A **management stage** is a governance interval; a **release** makes capability available; an **iteration or sprint** is a delivery cycle. They are different boundaries and need not coincide.
- **Gantt charts** can show project milestones and dependencies. Backlogs and boards help teams plan detailed work. Choose a representation that supports the decision and keep plans consistent across levels.

Timeboxes, estimates and learning

- A **timebox** is a fixed period for work toward a goal. Timeboxes can be nested; a release may contain several iterations. Protect the endpoint by managing content and capacity.
- **Story points** estimate relative size, complexity and uncertainty; **T-shirt sizing** uses broad categories such as S, M and L. Team estimates support conversation rather than promising exact elapsed time.
- **Velocity** is the amount of work a team completes per iteration, commonly in story points. Use that team's history to inform forecasts; compare like work and avoid treating velocity as an individual productivity score.
- A **spike** is time-limited investigation to reduce uncertainty. **Lean Startup** uses build-measure-learn and an MVP to test assumptions with minimal effort, then adapt based on evidence.

Quality & Risk

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

PROTECT DELIVERY INTEGRITY

- **Ready** — Can start
- **Done** — Complete
- **Risk** — Uncertainty
- **Issue** — Attention now

04 Quality, risk and change control

Quality, Ready and Done

- The **quality** practice defines what makes products fit for purpose and how this will be verified. Capture user expectations early and translate them into measurable acceptance and quality criteria.
- **Definition of Ready (DoR)** describes when an item is sufficiently understood to start: clear need, measurable acceptance criteria, manageable size and understood dependencies.
- **Definition of Done (DoD)** is the shared completion standard, such as completed development, testing and required documentation. **Acceptance criteria** concern the individual item; DoD applies consistently across work.
- Build quality into each increment. Address minor control failures by revisiting development and tests; escalate according to impact and authority. **Refactoring** improves internal structure while preserving intended behavior.
- **Quality tolerance** permits agreed variation in a criterion. Prioritize criteria explicitly; do not meet a deadline by silently dropping essential tests, documentation or acceptance requirements.

Risk management and agile suitability

- A **risk** is an uncertain event that could affect objectives, positively or negatively. Risk management should be proportionate; agile feedback reduces some uncertainties but does not remove the need for risk control.
- **Identify** risks; **assess** probability, impact and overall exposure; **plan** responses; **implement** them and monitor effectiveness. Communicate throughout and give risks and responses clear owners.
- The **risk management approach** defines how risks are handled; the **risk register** records individual risks and responses. Include adoption, delivery, dependencies and transition risks.
- The **Agilometer** assesses suitability and risks of agile working. The project manager facilitates an early assessment and repeats it as conditions change; use each area to guide improvements rather than averaging away weaknesses.
- Its areas cover **scope flexibility, collaboration, communication, iterative and incremental capability, environmental conditions and acceptance of agile**. An **enterprise agility assessment** considers adoption across the wider organization.

Quality & Risk / Processes & Workshops

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

PROJECT DECISIONS

Start and initiate → **Control and deliver** → **Review boundary** → **Close and hand over**

Issues, changes and delegated decisions

- An **issue** is something requiring management attention now; a risk concerns uncertainty. Capture the issue, assess impact and options, recommend action, obtain the appropriate decision and implement it.
- The **issues** practice controls changes affecting approved baselines. **Change control** evaluates consequences for products, plans, benefits, cost, risk and quality before committing.
- Authorize teams to manage emerging detail and trade work within agreed boundaries. A change to an approved baseline or beyond delegated tolerance follows the appropriate governance route.
- Keep decisions and product versions traceable. Frequent releases increase the importance of **configuration management**, clear ownership and consistent records of what was built, tested and released.

05 Processes and collaborative workshops

Starting, directing and initiating

- **Starting up a project** checks whether there is a worthwhile basis to initiate. Establish initial roles, lessons, project brief and an initiation plan without unnecessary detailed design.
- **Directing a project** is the board's decision-making process across the lifecycle: authorize initiation, delivery stages and closure, and give direction when needed. The board manages by exception.
- **Initiating a project** establishes a sound foundation: business case, project plan, management approaches, roles and controls. Agree how agile delivery, releases, quality and reporting will work.
- **Starting up, kickoff and initiation workshops** build shared understanding of purpose, roles and working arrangements. **Agile enablement workshops** address readiness, adoption risks and common delivery standards, including Ready and Done.

Controlling stages and delivering products

- **Controlling a stage** is the project manager's cycle of authorizing work, reviewing progress, handling risks and issues, taking corrective action and reporting to the board.
- **Managing product delivery** controls the agreement between project management and delivery: accept a work package, execute the work, and deliver products meeting the agreed requirements.
- A **work package** communicates required products, quality expectations, constraints, reporting and acceptance arrangements. Agree what delivery involves before the team commits.
- **Team planning workshops** clarify goals and select achievable work. **Progress review workshops** inspect status, obstacles and forecasts. Keep team-level information connected to stage-level decisions.

Processes & Workshops / Delivery & Progress

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

PROJECT DECISIONS

Start and initiate → **Control and deliver** → **Review boundary** → **Close and hand over**

Stage boundaries, closure and handover

- **Managing a stage boundary** reviews stage performance, updates the business case and project plan, and prepares the next stage plan so the board can decide whether to continue.
- **Release planning and release review workshops** can support stage-boundary decisions by revisiting priorities, dependencies, delivered value and plans for further work.
- **Closing a project** confirms acceptance, reviews performance and lessons, and transfers remaining responsibilities. Plan benefits reviews and follow-on actions for anything outstanding.
- **Project closure workshops** consolidate outcomes and lessons. Update the project canvas and relevant records, confirm ownership and prepare the operational handover.
- **Transition to operations** needs clear functionality and support documentation, training, ownership and applicable service levels. Maintain feedback loops, suitable KPIs and accessible work information for ongoing improvement.

Facilitating useful workshops

- A **workshop** is a collaborative, facilitated event with a defined purpose and outputs. Invite the right participants, prepare inputs, timebox discussion and record decisions and actions.
- Well-run workshops build **consensus and ownership** through shared understanding. Use them for problem solving, requirements, planning, personas and alignment; include differing stakeholder perspectives.
- A **review** inspects the product and obtains feedback. A **retrospective** inspects how people worked and selects improvements. Choose a few valuable actions for the next cycle and check whether they helped.
- **Glad! Sad! Mad!** is a retrospective prompt for experiences and feelings. Combine these signals with evidence, use a safe atmosphere and focus on improvement rather than blame.

06 Delivery approaches, progress and feedback

Scrum, Kanban and Lean

- **Scrum** is an iterative, incremental framework using fixed-length sprints. **Sprint planning** agrees the goal, achievable work and delivery approach; the **Daily Scrum** inspects progress toward the goal and adapts the plan.
- A **Sprint Review** inspects the increment with stakeholders and considers what to do next. A **Sprint Retrospective** improves effectiveness and quality. An **increment** is usable work meeting the Definition of Done.

Delivery & Progress

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

EVIDENCE DRIVES ADAPTATION

Visualize work → **Deliver and measure** → **Review feedback** → **Adapt next work**

- **Kanban** visualizes workflow, limits **work in progress (WIP)** and improves flow. Pull new work when capacity exists; expose bottlenecks and help finish work before starting more.
- **Lean** emphasizes customer value, reduced waste and smooth flow. **Swarming** brings several people together to finish or unblock an item. Flow-based work does not require sprint boundaries.

Progress, reports and dashboards

- The **progress** practice compares actual results with plans, forecasts future performance and supports control of deviations. In agile delivery, emphasize completed products and remaining scope, alongside time and cost.
- A **checkpoint report** communicates team or work-package progress to the project manager. A current team board and conversations can supply the information when they satisfy agreed reporting needs.
- A **highlight report** summarizes stage status, forecasts, risks and issues for the project board at agreed intervals. A forecast tolerance breach requires exception handling rather than waiting for the next routine report.
- **Team dashboards** show delivery flow and detail; **project dashboards** summarize value, milestones and overall health. Information radiators can let managers pull current information, with clear escalation for decisions.

Delivery & Progress

PRINCE2 Agile Foundation

KNOWLEDGE CHEATSHEET

EVIDENCE DRIVES ADAPTATION

Visualize work → **Deliver and measure** → **Review feedback** → **Adapt next work**

Burn charts and evidence of progress

- A **burn-down chart** plots remaining work against time; a **burn-up chart** plots completed work and can show a separate total-scope line. Scope changes are easier to see when shown explicitly.
- The **ideal line** is a planning reference; the actual line is evidence. Flat remaining work, slow completion or rising scope prompts investigation and a revised forecast.
- Charts need consistent units, reliable completion criteria and context. Elapsed time or hours spent does not prove usable delivery; count work according to the agreed completion standard.
- Use **velocity**, WIP, lead time and cycle time to inform improvement. **Lead time** follows work from request to delivery; **cycle time** measures elapsed time through the defined active workflow.

Frequent releases and informed adaptation

- **Frequent releases** bring earlier value and real-user feedback, reduce batch size and improve handover experience. Maintain customer involvement, quality controls and configuration discipline.
- A **staging area** permits validation, integration or regulatory checks before operational use. Completing an increment, releasing it and placing it into live use are distinct events.
- **Product management** continues across the product lifecycle, beyond project closure. Keep prioritizing value, measuring outcomes and learning from operation through ongoing feedback.
- **AI support** can help summarize information, detect patterns and provide forecasts that inform iterative decisions. Validate inputs and outputs; accountable people still make governance, quality and release decisions.