Scrum foundations

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

EMPIRICAL LEARNING

Transparency Make visible → **Inspection** Observe → **Adaptation** Adjust

01 Scrum foundations and team accountabilities

Empiricism and Scrum values

- **Scrum** is a lightweight framework for complex problems. Iterative, incremental delivery makes learning possible and limits risk; lean thinking reduces waste.
- **Transparency, inspection and adaptation** work together: make work and results visible, inspect them frequently, then adjust promptly. Events create regular opportunities for this feedback.
- **Commitment, focus, openness, respect and courage** guide behavior. Commit to goals and support colleagues; make problems visible and address difficult issues together.

A cohesive, self-managing team

- A **Scrum Team** consists of one Product Owner, one Scrum Master and Developers, typically **10 or fewer people** in total. There are no internal subteams or hierarchies.
- **Cross-functional** means the team collectively has the skills needed to create value each Sprint. A person need not possess every skill; testing and other specialist work belong within the shared delivery capability.
- **Self-managing** teams decide internally who does what, when and how. Let people help shape cohesive teams and resolve collaboration problems; the Scrum Master coaches rather than assigns tasks.
- The **entire Scrum Team** is accountable for a valuable, useful Increment every Sprint. Membership may change when needed, but changes can temporarily disrupt collaboration and productivity.

Developers and Scrum Master

- **Developers** create and adapt the Sprint Backlog, uphold the Definition of Done and hold one another accountable. Work and progress toward the Sprint Goal are their shared responsibility.
- **Developer** includes people doing any work needed for the Increment, including testing. Scrum defines no separate Tester, team lead or Project Manager accountability.
- The **Scrum Master** establishes Scrum and is accountable for team effectiveness: coach self-management, cause impediments to be removed and ensure useful events within their timeboxes.
- The Scrum Master helps the Product Owner with **Product Goal definition, backlog management and empirical planning**. Neither accountability is a replacement name for a traditional project manager.

Product ownership & value

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

PRODUCT VALUE DECISIONS

- **Customer need** → **Value objective** → **Usable Increment** → **Observed outcomes**

02 Product ownership, value and stakeholders

One accountable Product Owner

- The **Product Owner** is accountable for maximizing product value and effective Product Backlog management: communicate the Product Goal, express items clearly, order them and ensure transparency.
- The Product Owner may **delegate backlog work**, including item creation and ordering activities, while retaining accountability. Others can contribute ideas and clarification throughout development.
- A product has **one Product Owner**, providing clear value accountability and a single decision point for backlog order. The organization must respect these decisions; stakeholder wishes inform them.
- The Product Owner participates as a **Scrum Team member**, including in Planning, Review and Retrospective. Spend enough time with Developers to build shared understanding; Scrum prescribes no attendance percentage.

Measure outcomes and value

- **Product value** depends on the product and its stakeholders. Look at customer satisfaction, useful outcomes, revenue and costs together; financial return is not the only possible benefit.
- Measure **customer satisfaction frequently** and seek direct feedback. Observe adoption, actual feature usage and how easily customers can begin benefiting from the product.
- **Time to market** reveals how quickly an idea can reach customers and generate learning. A usable Increment provides capability; real usage helps test whether the expected benefit occurs.
- **Velocity, utilization and percentage of scope delivered** describe activity or output. They do not establish value; compare observed customer and business results with the intended outcomes.

Stakeholders and product direction

- Use **customer feedback, competitive awareness, feasibility and product direction** to make value decisions. Clarify whose problem the product solves and what evidence would demonstrate improvement.
- Collaborate with **users, customers and other stakeholders** throughout the Sprint. The Sprint Review is a key feedback opportunity, rather than the only permitted contact.
- Share the **Product Backlog, inspectable results, forecasts and impediments** to make progress understandable. External management concerns deserve transparent discussion; the Daily Scrum serves the Developers.
- When delivered results differ from expectations, increase **Product Owner and Developer collaboration**. Clarify outcomes and quality expectations early enough to adapt, rather than waiting for a handoff.

Backlog & forecasting

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

DIRECTION AND REFINEMENT

- **Product Goal** → **Ordered backlog** → **Refine and size** → **Sprint selection**

03 Product Goal, backlog and forecasting

Direction and a single source of work

- The **Product Goal** describes a future product state and gives the team a longer-term planning target. Fulfill or abandon one Product Goal before taking on the next.
- A shared Product Goal connects **individual Sprints to product direction**, helps evaluate decisions and supports inspection of progress at the Sprint Review.
- The **Product Backlog** is an emergent, ordered list of what is needed to improve the product and the single source of work for the Scrum Team. Keep it visible, understood and current as learning occurs.
- Include **features, defects, maintenance, documentation and non-functional requirements** as appropriate. The backlog evolves; a complete, detailed requirements inventory is not needed before the first Sprint.

Ordering and continuous refinement

- Order items to maximize **value and progress toward the Product Goal**, considering customer importance, risk, dependencies and feasibility. Keeping everyone busy is not the primary ordering objective.
- **Refinement** continuously breaks down and clarifies items, adding details such as description, order and size. It can happen during a Sprint and during Planning; it is not a separate required Scrum event.
- **Developers doing the work size the items**. The Product Owner helps explain needs and trade-offs. Refine likely near-term work in more detail; reassess stale items to avoid maintaining unnecessary inventory.
- Items ready for selection can be **Done within one Sprint** and are understood well enough to plan. Scrum does not require a particular user-story format, exhaustive acceptance criteria or a formal Definition of Ready.

Forecasts, uncertainty and optional techniques

- A **forecast** uses evidence such as past delivery, upcoming capacity and the Definition of Done. Update forecasts and backlog plans when evidence changes rather than treating early scope as guaranteed.
- The **cone of uncertainty** illustrates uncertainty in forecasts; longer horizons generally carry more uncertainty. Learning from completed work helps refine future expectations.

Backlog & forecasting / Sprints & events

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

A CONTINUOUS SPRINT CADENCE

Planning → **Build + Daily Scrum** → **Review outcomes** → **Retrospective**

- **Burn-downs, burn-ups and cumulative flow charts** may help visualize progress. Scrum does not mandate a chart, estimation unit, Gantt chart, critical-path analysis or ordering formula.
- **Acceptance criteria** can clarify an item's expected behavior. The **Definition of Done** describes the product's required quality state; satisfying an item's criteria alone does not replace it.

04 Sprints and inspection events

Sprint boundaries and change

- A **Sprint** has a fixed length of **one month or less**. Planning, delivery work, Daily Scrums, Review and Retrospective all happen inside it; the next Sprint starts immediately afterward.
- A **timebox** is a maximum duration. Choose a cadence that supports learning and acceptable risk; do not extend a Sprint because the forecast was not completed.
- During a Sprint, protect the **Sprint Goal** and do not decrease quality. Developers and Product Owner can clarify and renegotiate scope as they learn; new requests can enter the Product Backlog.
- Only the **Product Owner** can cancel a Sprint, if its Goal becomes obsolete. Difficulty completing selected work calls for collaboration and replanning rather than automatic cancellation.

Sprint Planning: why, what and how

- **Sprint Planning** is collaborative work of the whole Scrum Team. The Product Owner brings important items and a value objective; the team may invite others to provide advice.
- **Why:** the whole team crafts the Sprint Goal and finalizes it before Planning ends. It gives the Sprint a coherent purpose beyond a list of unrelated tasks.
- **What:** Developers select how much work to forecast through discussion with the Product Owner. Consider capacity, past performance and quality requirements; unclear items reduce confidence.
- **How:** Developers plan the work to create a Done Increment. The result is the Sprint Backlog; detail emerges during the Sprint. Planning is at most **8 hours for a one-month Sprint**, usually less for shorter Sprints.

Sprint Goal and the Developers' plan

- The **Sprint Backlog** contains the Sprint Goal (**why**), selected Product Backlog items (**what**) and an actionable delivery plan (**how**). It is a plan by and for the Developers.
- The **Sprint Goal** is the Developers' commitment and a single objective. The exact work can change while preserving that objective; selected scope is a forecast rather than a fixed-scope contract.

Sprints & events

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

A CONTINUOUS SPRINT CADENCE

Planning → **Build + Daily Scrum** → **Review outcomes** → **Retrospective**

- Developers keep the plan **visible and current throughout the Sprint**, adding or decomposing work as soon as needed. Tasks, tests, stories or other useful representations can appear in it.
- When the forecast is too large, **collaborate with the Product Owner promptly** to adjust selected scope while protecting the Sprint Goal. Do not lower quality, mandate overtime or wait until the Review.

Daily Scrum: inspect and adapt the plan

- The **Daily Scrum** is a **15-minute event** for Developers every working day, at the same time and place to reduce complexity. Inspect progress toward the Sprint Goal and create an actionable next-day plan.
- Developers choose the **format and who starts**. Standing up, three prescribed questions and a Scrum Master status round are not required. Detailed follow-up discussions can happen afterward.
- A Product Owner or Scrum Master actively working on Sprint Backlog items participates **as a Developer**. Neither is otherwise required to attend; the Scrum Master ensures the event takes place.
- Less frequent Daily Scrums lose opportunities to **adapt the plan, identify impediments and make decisions**. Developers can also inspect and replan throughout the day.

Review outcomes; improve the way of working

- The **Sprint Review** brings the Scrum Team and key stakeholders together to inspect outcomes, progress toward the Product Goal and changed conditions. Collaborate on what to do next and adapt the Product Backlog.
- The Review is a **working session**, not merely a demonstration or approval meeting. The Product Owner remains accountable for backlog ordering. Its maximum is **4 hours for a one-month Sprint**.
- The **Sprint Retrospective** is for the whole Scrum Team to improve quality and effectiveness. Inspect interactions, processes, tools, skills and the Definition of Done; the Product Owner participates as a member.
- Address the most useful **improvements** promptly; they may enter the next Sprint Backlog. The Retrospective concludes each Sprint and is at most **3 hours for a one-month Sprint**. Review and Retrospective are usually shorter for shorter Sprints.

Quality & multiple teams

Professional Scrum Product Owner I

KNOWLEDGE CHEATSHEET

ARTIFACTS AND COMMITMENTS

- **Product Backlog** — Product Goal
- **Sprint Backlog** — Sprint Goal
- **Increment** — Definition of Done

05 Increments, quality and multiple teams

Usable Increments and release

- An **Increment** is usable, additive to previous work and verified to work with it. Create a valuable, useful Increment every Sprint; a plan or unfinished design alone does not establish usable product value.
- **Product Backlog -> Product Goal; Sprint Backlog -> Sprint Goal; Increment -> Definition of Done:** each artifact has a commitment that makes progress and quality easier to inspect.
- Multiple Increments can be created in a Sprint and **released before its end**. Every Sprint does not require a customer release, and the Sprint Review is not a release gate.
- An item that does not meet the **Definition of Done** is excluded from the Increment, cannot be released or presented as completed work at the Review, and returns to the Product Backlog for future consideration.

Definition of Done and non-functional needs

- The **Definition of Done** creates a shared understanding of completed work and makes the Increment transparent. Developers perform all work needed to satisfy it, including applicable verification and integration.
- Follow **organizational quality standards** as a minimum. If there is no organizational Definition of Done, the Scrum Team creates an appropriate one; the Product Owner cannot independently waive quality.
- Make **security, performance, stability and other non-functional needs** visible through suitable Product Backlog items and recurring Definition of Done criteria. Include relevant specialists without transferring delivery accountability.
- Inspect and strengthen the **Definition of Done** as the team learns, often during Retrospectives. Hidden unfinished work and technical debt can increase total cost of ownership; quality must not fall to rescue a forecast.

One product, coordinated delivery

- Multiple Scrum Teams working on one product share **one Product Owner, one Product Goal and one Product Backlog**. Separate team backlogs for the product fragment ordering and value decisions.
- Teams must mutually define and follow the **same Definition of Done**. Their work must integrate with the product; integration is part of creating usable Increments, not a later handoff.
- Developers coordinate **dependencies and end-to-end work** with one another. Analyze and split work collaboratively, broaden skills and raise external specialist bottlenecks as impediments.
- Scrum requires no **Sprint 0, hardening Sprint or release Sprint**, and does not mandate identical team Sprint start dates. Organize collaboration so each team can deliver integrated, Done value every Sprint.