

Scrum Master Checklist

A full sprint to tick off, from refinement to retro

This checklist is for Scrum Masters and project leads running agile software projects. It covers a complete sprint plus the things that tend to slip in daily practice: Definition of Ready and Done, stakeholders, metrics and remote work. Run through it once per sprint and tick things off.

Before the sprint: refinement and planning prep

- ☐ Schedule at least one refinement session per sprint as a standing meeting.
Ad-hoc refinement in the middle of planning costs everyone time and quality.
- ☐ Only bring items into planning that meet the Definition of Ready.
- ☐ Write the sprint goal as a single sentence before selecting items.
Without a goal, the sprint turns into a plain ticket list.
- ☐ Calculate capacity honestly: subtract vacations, public holidays and support duty.
- ☐ Clarify dependencies on other teams or external partners before planning.

During the sprint: daily and flow

- ☐ Keep the daily to 15 minutes and move detail discussions to follow-ups.
- ☐ Focus the daily on progress toward the sprint goal, not on a status round per person.
- ☐ Address blockers the same day and name one person responsible.
- ☐ Reconcile the board with reality every day and fix stale tickets immediately.

Keeping Definition of Ready and Done healthy

- ☐ Keep both definitions written down and visible to everyone.
The Definition of Ready is a complementary practice, only the Definition of Done appears in the official Scrum Guide.
- ☐ Write acceptance criteria per story so they are testable.
A criterion is testable when an outsider can verify it.
- ☐ Make the Definition of Done include tests, code review and documentation, not just "code complete".
- ☐ Review and sharpen both definitions regularly, for example once a quarter.

Review and retrospective

- ☐ Only demo functionality in the review that meets the Definition of Done.
- ☐ Feed stakeholder feedback from the review straight into the backlog.
- ☐ Return unfinished items to the product backlog at sprint end and re-order them.
No automatic carry-over into the next sprint, open items get re-prioritized and re-estimated if needed.
- ☐ Rotate the retro format regularly.
Otherwise you get the same answers every time.
- ☐ Agree on one or two improvement actions per retro at most, each with an owner and a due date.
- ☐ Start every retro by checking whether the last retro's actions were actually done.

Stakeholders and metrics

- ☐ Clarify each sprint which stakeholders need which update by when.
- ☐ Send a short report after each sprint: delivered, risks, next steps.
- ☐ Use velocity only inside the team as a planning aid, never as a performance comparison.
Once velocity gets judged, estimates grow, not delivery.
- ☐ Track metrics that carry meaning: cycle time, finished versus started items, sprint goal met yes or no.
- ☐ Drop vanity metrics: story points per person, commit counts, utilization percentages.

Remote teams

- ☐ Define a daily overlap window when everyone is reachable.
Essential across time zones, two hours is often enough.
- ☐ Record decisions in writing, in the ticket, PR or wiki, not just in the call.
- ☐ Measure progress by artifacts: pull requests and ticket status, not online status in chat.
- ☐ Agree on response times for async work, for example review feedback within 24 hours.

If your team is short on development capacity, Frankford's IT-Solutions in Innsbruck delivers remote software development in a milestone model, with a Definition of Done, code reviews and clear acceptance. Learn more at frankford-it.at.