

Scrum Master Checkliste

Der komplette Sprint zum Abhaken, von Refinement bis Retro

Diese Checkliste richtet sich an Scrum Master und Projektleiter in agilen Softwareprojekten. Sie deckt einen kompletten Sprint ab und ergänzt die Punkte, die im Alltag gern untergehen: Definition of Ready und Done, Stakeholder, Metriken und Remote-Arbeit. Am besten einmal pro Sprint durchgehen und abhaken.

Vor dem Sprint: Refinement und Planning vorbereiten

- ☐ Mindestens einen Refinement-Termin pro Sprint fix einplanen.
Ad-hoc-Refinement mitten im Planning kostet alle Zeit und Qualität.
- ☐ Nur Items ins Planning nehmen, die die Definition of Ready erfüllen.
- ☐ Das Sprint-Ziel vor der Auswahl der Items in einem Satz formulieren.
Ohne Ziel wird der Sprint zur reinen Ticketliste.
- ☐ Kapazität ehrlich rechnen: Urlaube, Feiertage und Support-Anteil abziehen.
- ☐ Abhängigkeiten zu anderen Teams oder externen Partnern vor dem Planning klären.

Im Sprint: Daily und Arbeitsfluss

- ☐ Das Daily auf 15 Minuten begrenzen, Detailfragen in Folgetermine auslagern.
- ☐ Im Daily auf den Fortschritt Richtung Sprint-Ziel schauen, nicht auf eine Statusrunde pro Person.
- ☐ Blocker noch am selben Tag angehen und eine verantwortliche Person benennen.
- ☐ Das Board täglich mit der Realität abgleichen und veraltete Tickets sofort korrigieren.

Definition of Ready und Done pflegen

- ☐ Beide Definitionen schriftlich festhalten und für alle sichtbar ablegen.
Die Definition of Ready ist eine ergänzende Praxis, im offiziellen Scrum Guide steht nur die Definition of Done.
- ☐ Akzeptanzkriterien pro Story testbar formulieren.
Ein Kriterium ist testbar, wenn es eine außenstehende Person prüfen kann.
- ☐ In die Definition of Done Tests, Code Review und Doku aufnehmen, nicht nur „Code fertig“.
- ☐ Beide Definitionen regelmäßig prüfen und nachschärfen, etwa einmal pro Quartal.

Review und Retrospektive

- ☐ Im Review nur Funktionalität zeigen, die die Definition of Done erfüllt.
- ☐ Stakeholder-Feedback aus dem Review direkt ins Backlog übernehmen.
- ☐ Unfertige Items am Sprint-Ende zurück ins Product Backlog geben und neu einordnen.
Kein automatischer Übertrag in den nächsten Sprint, offene Items werden neu priorisiert und bei Bedarf neu geschätzt.
- ☐ Das Retro-Format regelmäßig wechseln.
Sonst kommen jedes Mal dieselben Antworten.
- ☐ Pro Retro höchstens ein bis zwei Maßnahmen beschließen, jeweils mit Verantwortlichem und Termin.
- ☐ Jede Retro mit dem Check starten, ob die Maßnahmen der letzten Retro umgesetzt wurden.

Stakeholder und Metriken

- ☐ Pro Sprint klären, welche Stakeholder welchen Stand bis wann brauchen.
- ☐ Nach jedem Sprint einen kurzen Bericht schicken: geliefert, Risiken, nächste Schritte.
- ☐ Velocity nur teamintern als Planungshilfe verwenden, nie als Leistungsvergleich.
Sobald Velocity bewertet wird, wachsen die Schätzungen, nicht die Lieferung.
- ☐ Metriken mit Aussagekraft verfolgen: Durchlaufzeit, fertige gegenüber begonnenen Items, Sprint-Ziel erreicht ja oder nein.
- ☐ Vanity-Metriken streichen: Story Points pro Person, Commit-Anzahl, Auslastung in Prozent.

Remote-Teams

- ☐ Ein tägliches Überlappungsfenster definieren, in dem alle erreichbar sind.
Bei mehreren Zeitzonen entscheidend, zwei Stunden reichen oft.
- ☐ Entscheidungen schriftlich festhalten, im Ticket, PR oder Wiki, nicht nur im Call.
- ☐ Fortschritt an Artefakten messen: Pull Requests und Ticketstatus statt Online-Status im Chat.
- ☐ Antwortzeiten für asynchrone Arbeit vereinbaren, etwa Review-Feedback binnen 24 Stunden.

Wenn Ihrem Team gerade Entwicklungskapazität fehlt, unterstützt Frankford's IT-Solutions aus Innsbruck mit Remote-Softwareentwicklung im Meilensteinmodell, inklusive Definition of Done, Code Reviews und klarer Abnahme. Details finden Sie auf frankford-it.at.