Definition of Done

Photo by Glenn Carstens-Peters on Unsplash

WANN: Zu jedem Zeitpunkt während des agilen Entwicklungsprozesses

WER: Entwicklungsteam, Scrum Master, Product Owner

Eine in Scrum gebräuchliche Phrase für „Fertig“ ist die „Definition of Done“. Sie definiert eine Sammlung von Kriterien, die genau festlegen, wann eine User Story als abgeschlossen gilt.

ENTWICKLUNG DER „DEFINITION OF DONE“

Jedes Team muss seine **eigene „Definition of Done“** erstellen. Das Team einigt sich auf die initiale Version des „Definition of Done“, wenn es sich mit der*m Kund*in und den Stakeholdern erstmalig auf die gewünschten Anforderungen an das Produkt geeinigt hat.

Wenn man während des **Sprint Planning Meetings** die User Stories für den kommenden Sprint auswählt, geht das Entwicklungsteam gemeinsam mit dem **Product Owner** die “Definition of Done” durch. Das dient dem besseren Verständnis davon, wann das zu erstellende Produktinkrement als abgeschlossen gilt.

Die „Definition of Done“ muss dann im **Sprint** gelebt werden. Ziel ist es, nicht erst am Ende des Sprints sicherzustellen, dass die Stories auch wirklich fertig sind, sondern einen kontinuierlichen Fertigstellungsfluss zu etablieren, in dem Story für Story überprüft wird, ob sie fertig ist, um erst dann mit der nächsten Story fortzufahren.

Es liegt in der Verantwortung des **Scrum Master** sicherzustellen, dass alle die “Definition of Done” kennen und verstehen. Er sorgt dafür, dass keine User Stories präsentiert oder geliefert werden, die nicht der “Definition of Done” entsprechen. Solche User Stories gehen zurück ins Product Backlog und werden von dem Product Owner neu priorisiert.

In der **Sprint Retrospektive** kann festgestellt werden, dass die Definition of Done angepasst und verbessert werden sollte.

RELEVANZ

- Ist „Done“ gar nicht oder nur schwammig definiert, kann es vorkommen, dass das Team ewig lange an einer einzigen Story arbeitet.
- Deswegen ist es so wichtig eine klare, eindeutige und von allen Beteiligten verstandene „Definition of Done“ zu haben.
- Die Missachtung der „Definition of Done“ holt das Team ein. Eine User Story kann lieferbar, aber nicht fertig sein. Wenn die Story in so einem Fall trotzdem geliefert wird, geht das Team eine **technische Schuld** ein. Schulden müssen irgendwann beglichen werden, das heißt, die unterlassene Arbeit an der Story muss in einem der folgenden Sprints nachgeholt werden.

AKZEPTANZKRITERIEN

- Ein wichtiger Teil einer User Story sind ihre **Akzeptanzkriterien**, die bestimmen, wann eine Story vollständig umgesetzt und im Sinne der geltenden „Definition of Done“ fertig ist.
- Die Akzeptanzkriterien werden für jede User Story spezifisch formuliert. Die „Definition of Done“ enthält somit die Akzeptanzkriterien, die für jede User Story gelten.

WEITERFÜHRENDE INFOS

Wirdemann, R. (2011). Scrum mit User Stories. 2., erweiterte Auflage. München: Hanser, Carl, 37.

https://agilescrumgroup.de/definition-of-done/

Mittelstand-Digital

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

Probieren Sie es jetzt aus!

PRAXISTIPPS

DEFINITION OF DONE AM BEISPIEL EINER USER STORY

- Die User Story ist programmiert und erfüllt die im Sprint Planning Meeting festgelegten Anforderungen.
- Die User Story ist Unit-getestet und der Quellcode wurde refaktorisiert (umstrukturiert).
- Der Quellcode ist für alle Teammitglieder verständlich.
- Die User Story führt zu keinem Anstieg der „Technischen Schuld“.
- Die User Story wurde ins Gesamtsystem integriert.
- Die User Story wurde auf Akzeptanz getestet. Alle auf der Rückseite der Story-Karte notierten Akzeptanzkriterien sind erfüllt.
- Die User Story wurde vom Product Owner abgenommen.
- Die User Story wurde vom Quality Acceptance (QA)-Team abgenommen.

IHRE AUFGABE

Stellen Sie sich vor, Sie arbeiten in einem IT Scrum Team und wollen eine neue Simulationssoftware zum Testen des autonomen Fahrens entwickeln.

Überlegen Sie sich, wie eine ganz allgemeine Formulierung der „Definition of Done“ für dieses Szenario aussehen könnte.

Sie finden unseren Vorschlag umgedreht weiter unten.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

RÜCKBLICK AUFGABE

Bitte reflektieren Sie kurz folgende Fragen:

- Was ist schwer gefallen?
- Wo könnten Probleme liegen?

WEITERFÜHRENDE INFOS

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The scrum primer. http://www.brianidavidson.com/agile/docs/scrumprimer121.pdf

https://erfolgreich-projekte-leiten.de/akzeptanzkriterien-mit-gherkin/

LÖSUNGSVORSCHLAG

- Der gesamte Code ist geschrieben (alle To Do’s bzgl. des Codes sind erledigt).
- Die benötigte Benutzerdokumentation wurde erstellt und ist verfügbar.
- Es gibt eine Installationsanleitung für die Software.
- Alle funktionalen Tests wurden durchgeführt.
- Der Code wurde einer Peer Review unterzogen.

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages