

Klarheit, Struktur und gemeinsames Verständnis

Mit Triple-C-Card zu besseren User Storys



Dr. Tomas Bohinc

PMP, Berater für Teams,
Trainer für Projektleiter, Autor
Hochschule für
Telekommunikation Leipzig

Management Summary

- Die Triple-C-Card verbessert die Qualität von User Storys, indem sie schriftliche Beschreibung, fachlichen Dialog und überprüfbare Akzeptanzkriterien systematisch miteinander verbindet.
- Das strukturierte Gespräch (Conversation) macht unterschiedliche Annahmen von Stakeholdern und Entwicklungsteam frühzeitig sichtbar, sodass Anforderungen vor der Umsetzung gemeinsam präzisiert werden können.
- Dabei helfen gezielte Fragen zu Rahmenbedingungen, Ausnahmen, Schnittstellen, Sicherheitsanforderungen und Abgrenzungen, Missverständnisse frühzeitig zu erkennen und ein gemeinsames Verständnis zu schaffen.
- Die Confirmation übersetzt dieses gemeinsame Verständnis in überprüfbare Akzeptanzkriterien. Das Given-When-Then-Format bietet dafür eine bewährte Struktur.
- Die Triple-C-Card unterstützt eine klare und dialogorientierte Anforderungsbeschreibung, verringert Nacharbeit, Kostensteigerungen und Verzögerungen und stärkt die Qualität der Umsetzung.

Die User Story schien eindeutig. Das Entwicklungsteam setzte die Funktion zur Bestellungsfreigabe in einem Onlineshop wie beschrieben um. Doch bei der Abnahme zeigte sich, dass wichtige Geschäftsregeln wie Bonitätsprüfungen, Freigabegrenzen oder rechtliche Vorgaben nie besprochen worden waren. Die Umsetzung entsprach der User Story – aber nicht den tatsächlichen Anforderungen. Von diesen Anforderungen hatten alle Beteiligten unterschiedliche Vorstellungen.

Warum User Storys oft missverstanden werden

Das Ziel von **User Storys** ist es, Anforderungen einfach, verständlich und nutzerzentriert zu formulieren. In der Praxis zeigt sich jedoch, dass dies nicht immer der Fall ist. Die häufigsten Gründe dafür sind:

- Storys sind unklar und zu umfangreich.
- Stakeholder und Entwicklungsteam interpretieren die Anforderungen unterschiedlich.
- Akzeptanzkriterien fehlen, sind unklar oder unvollständig.
- Rückfragen werden erst gestellt, wenn mit der Bearbeitung der User Story bereits begonnen wurde.

Im obigen Beispiel lautete die User Story:

"Als Kunde möchte ich meine Bestellung abschließen können, damit sie schnell bearbeitet und versendet werden kann."

Weil dabei wichtige Detailregeln fehlten, reichte die User Story nicht aus, um ein gemeinsames Verständnis der Anforderungen herzustellen.

Missverständene User Storys können sich negativ auf Projekte auswirken. Dann entsprechen neu entwickelte Funktionen möglicherweise nicht den tatsächlichen Anforderungen oder Bedürfnissen der Anwender:innen. Werden solche Missverständnisse erst spät erkannt, kann das zu Nacharbeit und damit zusätzlichem Aufwand, Kostensteigerungen und Verzögerungen im Projektverlauf führen.

Auch Qualitätsmängel können die Folge sein, wenn Anforderungen oder Akzeptanzkriterien unterschiedlich interpretiert werden. Werden die Erwartungen von Kunden nicht erfüllt, sinkt sowohl die Akzeptanz des fertigen Produkts als auch das Vertrauen in das entsprechende Projekt. Dadurch wird der Projekterfolg insgesamt gefährdet.

An diesen Schwachstellen setzt die Triple-C-Card an: Sie sorgt dafür, dass unterschiedliche Annahmen früh sichtbar werden und Anforderungen gemeinsam präzisiert werden können.

Triple-C-Card: drei Schritte für eine gute User Story

User Storys gehören zu den am weitesten verbreiteten Werkzeugen in agilen Projekten. Obwohl die Triple-C-Card aus dem agilen Umfeld stammt, lässt sich das Grundprinzip dieser Methode überall dort anwenden, wo Anforderungen gemeinsam präzisiert und abgestimmt werden.

Die Triple-C-Card erweitert das bekannte Konzept der User Story um zwei oft vernachlässigte Bestandteile: das strukturierte Gespräch und überprüfbare Akzeptanzkriterien.

Warum Triple-C? Namensgebend sind drei Elemente, die für eine erfolgreiche Umsetzung von Anforderungen hilfreich sind und den Lebenszyklus einer User Story bilden:

Card – Conversation – Confirmation

Card (Karte): Das Projektteam hält die User Storys schriftlich auf Karten fest und hängt sie gut sichtbar, z.B. auf einer Tafel, auf. Dadurch behält es sie immer im Blick – und zugleich im Kopf.

Conversation (Gespräch): Im fachlichen Dialog entsteht ein gemeinsames Verständnis des Inhalts der Karte, und Details werden geklärt.

Confirmation (Akzeptanzkriterien): Das Projektteam legt überprüfbare Kriterien fest, die sicherstellen, dass die Storys korrekt umgesetzt werden.

Mit dieser einfachen Struktur schafft die Triple-C-Card Klarheit über Formulierung, Abstimmung und Überprüfung von Anforderungen.

1. Card (Karte)

Im ersten Schritt wird die User Story auf einer Karte beschrieben. Ursprünglich wurden User Storys auf 3 × 5 Zoll große Karteikarten (Index Cards) notiert. Die Idee dahinter: Der begrenzte Platz zwingt dazu, Anforderungen knapp und prägnant statt detailliert zu formulieren. Heute werden User Storys meist in digitalen Werkzeugen wie Jira, Azure DevOps oder Trello verwaltet.

Die Karte enthält die Anforderung in der folgenden strukturierten Form:

"Als **Rolle** möchte ich **Funktion**, damit ich **Nutzen** erziele."

Beispiel:

"Als Kunde möchte ich meine Bestellung abschließen können, damit sie schnell bearbeitet und versendet werden kann."

Der erste Teil der User Story beschreibt die Rolle, für die die Software entwickelt wird, der zweite Teil die Funktion, die ausgeführt werden soll, und der dritte Teil den Nutzen, den die Rolle davon hat. Vor allem der Nutzen hilft den Entwicklern zu verstehen, warum die Funktion benötigt wird.

Sinn der Karte ist es nicht, die Anforderung rechtssicher festzuhalten und jede nur denkbare Information schon in die Story zu schreiben. Die Beschreibung ist ein Anlass, über den Inhalt der Karte ein Gespräch zu führen. Jeffreys bezeichnete die Card deshalb als "Versprechen auf ein Gespräch". Dieses Gespräch ist der eigentliche Kern der Triple-C-Card.

2. Conversation (Gespräch)

Die auf der Karte notierte User Story lässt bewusst Interpretationsspielraum. Product Owner (im Folgenden "PO" genannt) und Entwicklungsteam bringen unterschiedliches Fachwissen, unterschiedliche Annahmen und unterschiedliche Kenntnisse des fachlichen Kontexts mit.

Eine scheinbar eindeutige User Story kann deshalb unterschiedlich aufgefasst werden. Die folgende Tabelle vergleicht typische Interpretationen der Beispiel-User-Story durch den PO und das Entwicklungsteam:

<i>Der Product Owner denkt ...</i>	<i>Das Entwicklungsteam denkt ...</i>
Vor der Freigabe wird die Bonität des Kunden geprüft.	Die Bestellung wird unmittelbar nach dem Klick auf "Bestellen" freigegeben.
Der Kunde muss den Allgemeinen Geschäftsbedingungen (AGB) zugestimmt haben.	Die Zustimmung zu den AGB wurde bereits im Warenkorb erfasst und muss nicht erneut geprüft werden.
Die Lieferadresse wurde validiert und ist vollständig.	Es wird lediglich geprüft, ob die Pflichtfelder ausgefüllt sind.
Nicht verfügbare Artikel dürfen nicht freigegeben werden.	Fehlende Artikel können später im Warenwirtschaftssystem behandelt werden.
Bestellungen über 5.000 € erfordern eine zusätzliche Genehmigung.	Jede Bestellung wird unabhängig vom Bestellwert automatisch freigegeben.
Kunden mit offenen Forderungen dürfen keine neue Bestellung abschließen.	Die Zahlungsfähigkeit wird in einem anderen System geprüft und gehört nicht zur User Story.
Erst nach erfolgreicher Prüfung aller Geschäftsregeln wird die Bestellung an das System zur Auftragsabwicklung übergeben.	Die Bestellung wird direkt an das System zur Auftragsabwicklung übertragen; weitere Prüfungen erfolgen dort.

Tabelle 1: Unterschiedliche Vorstellungen der Funktion "Bestellungsfreigabe" durch Product Owner und Entwicklungsteam

Genau hier setzt die Conversation an: Sie macht abweichende Annahmen sichtbar, bevor mit der Umsetzung begonnen wird. Erst wenn diese geklärt sind, entsteht ein gemeinsames Verständnis der User Story.

Anlässe für ein Gespräch über User Storys

Conversation findet überall dort statt, wo Anforderungen gemeinsam geschärft werden. In der Scrum-Welt geschieht dies typischerweise immer dann, wenn PO und Entwicklungsteam gemeinsam User Storys aus dem Product Backlog besprechen. Dazu gehören vor allem:

- **Sprint Planning:** Im Sprint Planning stellt der PO die wichtigsten Einträge des Product Backlogs vor. Das Team klärt offene Fragen zu den Anforderungen und legt mit dem PO den Umfang des Sprints fest.

Außerdem wird das Sprint-Ziel gemeinsam definiert.

- **Product Backlog Refinement (Backlog-Pflege):** Beim Product Backlog Refinement erläutert der PO neue Anforderungen sowie deren Prioritäten. Das Team stellt Fragen, schätzt den Aufwand der einzelnen Einträge und unterstützt dabei, die Anforderungen genauer zu beschreiben und zu strukturieren.
- **Laufende Abstimmung während des Sprints:** Während des gesamten Sprints stehen PO und Team in engem Kontakt. Das Team kann jederzeit Fragen zu Anforderungen oder Prioritäten stellen. Der PO unterstützt durch schnelle Klärung, damit die Arbeit reibungslos fortgesetzt werden kann.

Auch in klassischen oder hybriden Projekten findet ein Austausch über Anforderungen von Produkten oder Funktionen beispielsweise in Workshops oder anderen Abstimmungsgesprächen statt.

Checkliste: Fragen an User Storys

Das Team sollte hier die Initiative ergreifen und dem PO zu den User Storys beispielsweise folgende Fragen stellen:

- Welches Problem soll gelöst werden?
- Was genau soll umgesetzt werden?
- Welche Rahmenbedingungen und Annahmen gelten für die User Story?
- Gibt es Ausnahmen und wenn ja, welche?
- Welche anderen Systeme oder Prozesse sind betroffen?
- Welche Sicherheits-, Datenschutz- oder Compliance-Anforderungen sind einzuhalten?
- Was ist besonders wichtig für User:innen?
- Welche technischen Einschränkungen sind zu berücksichtigen?
- Was ist nicht Bestandteil der User Story?



Diese Checkliste ist ein Vorschlag. Legen Sie sich eine eigene Liste an, die Sie mit Fragen aus Ihren konkreten Anwendungsfällen erweitern.

Entscheidend ist nicht die Zahl der Fragen, sondern dass die richtigen gestellt werden. Ziel ist es, implizite Annahmen sichtbar zu machen und zwischen PO und Entwicklungsteam ein gemeinsames Verständnis der Anforderungen zu erzielen.

Wie ein solches Gespräch in der Praxis aussehen kann, zeigt der folgende Auszug:

Beispielgespräch zwischen Product Owner (PO) und Entwicklungsteam

Entwickler: Welche Rahmenbedingungen und Annahmen gelten für diese User Story?

PO: Wir gehen davon aus, dass sich der Kunde oder die Kundin bereits im Bestellprozess befindet, alle benötigten Artikel ausgewählt hat und eine gültige Liefer- sowie Zahlungsadresse angegeben hat.

Entwickler: Gibt es Ausnahmen, die wir berücksichtigen müssen?

PO: Ja. Beispielsweise dürfen Bestellungen bei fehlgeschlagener Zahlungsautorisierung, unvollständigen Adressdaten oder negativer Bonitätsprüfung nicht automatisch freigegeben werden. Außerdem gelten für bestimmte Produkte wie Gefahrgut oder altersbeschränkte Artikel zusätzliche Prüfungen.

Entwickler: Welche anderen Systeme oder Prozesse sind von der User Story betroffen?

PO: Für die Freigabe greifen wir auf den Zahlungsdienstleister, den Bonitätsdienst und das Warenwirtschaftssystem zu. Nach erfolgreicher Freigabe wird automatisch der Versandprozess gestartet.

Entwickler: Welche Sicherheits-, Datenschutz- oder Compliance-Anforderungen gelten?

PO: Personenbezogene Daten dürfen ausschließlich gemäß den Datenschutzvorgaben verarbeitet werden. Außerdem müssen alle Freigaben revisionssicher protokolliert werden, damit sie später nachvollzogen werden können.

Entwickler: Was ist aus Sicht der Nutzenden besonders wichtig?

PO: Kund:innen erwarten einen schnellen und zuverlässigen Bestellabschluss. Falls eine Bestellung nicht freigegeben werden kann, möchten sie den Grund dafür wissen und verständlich erklärt bekommen, wie das Problem behoben werden kann.

Entwickler: Was gehört ausdrücklich **nicht** zu dieser User Story?

PO: Die Zahlungsabwicklung, die Lagerreservierung und der Versandprozess sind bereits in anderen User Storys beschrieben worden und werden hier nicht umgesetzt.

3. Confirmation (Akzeptanzkriterien)

Während die Conversation ein gemeinsames Verständnis schafft, macht die Confirmation dieses Verständnis durch Akzeptanzkriterien überprüfbar. Mit anderen Worten: Akzeptanzkriterien legen fest, wodurch nachgewiesen wird, dass die Story erfüllt ist.

Eine bewährte Möglichkeit, Akzeptanzkriterien zu formulieren, ist das **Given-When-Then-Format**, das aus dem Behaviour-Driven Development (BDD) stammt. Bei diesem Ansatz der Softwareentwicklung wird das gewünschte Verhalten eines Systems aus Sicht der Anwendenden beschrieben, d.h. BDD hilft, Anforderungen eindeutig und testbar zu formulieren.

Dabei werden Akzeptanzkriterien in der folgenden Struktur beschrieben:

- **Given** (Gegeben): Beschreibt die Ausgangssituation oder die Voraussetzungen.
- **When** (Wenn): Beschreibt die Aktion oder das Ereignis.
- **Then** (Dann): Beschreibt das erwartete Ergebnis.

Dadurch wird klar festgelegt, unter welchen Bedingungen eine Funktion wie reagieren soll.

Für unsere User Story wird das folgende Akzeptanzkriterium formuliert:

Given: Ein registrierter Kunde befindet sich auf der Warenkorb-Seite.

When: Wenn der Kunde auf die Schaltfläche *Absenden* klickt ...

Then: ... dann erfolgen Prüfungsroutinen und die Bestellung wird an das Auftragsabwicklungssystem übergeben, wenn diese erfolgreich waren.

Die Akzeptanzkriterien werden typischerweise im Product Backlog Refinement und spätestens im Sprint Planning gemeinsam festgelegt.

Fazit

Die Triple-C-Card schafft Klarheit – genau dort, wo sie gebraucht wird. Sie wirkt unscheinbar, ist aber ein äußerst effektives Instrument, um User Storys verständlicher, dialogorientierter und überprüfbarer zu machen. Die Triple-C-Card strukturiert den gesamten Prozess der Anforderungsbeschreibung – ohne Teams zu überfrachten.

In einer Arbeitswelt, in der Anpassungsfähigkeit, Geschwindigkeit und klare Kommunikation entscheidend sind, ist die Triple-C-Card überall dort hilfreich, wo Anforderungen gemeinsam entwickelt, präzisiert und umgesetzt werden – in agilen ebenso wie in klassischen oder hybriden Projekten. (kt)

Literatur

Jeffreys, Ron: Essential XP: Card, Conversation, Confirmation. Online abrufbar unter:

<https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/> (Zuletzt aufgerufen am: 23.07.2026)