

☑ Requirements Engineering

Die **Anforderung** ist als User Story beschrieben

💡 "Als **Rolle** möchte ich **Anforderung** um / weil **Beweggrund**", wobei der Beweggrund oft vergessen wird, aber sehr wichtig ist!

Die **Testbeschreibung** ist ausgefüllt

💡 Welche Handgriffe muss ich tun, um die Funktionalität zu testen?

💡 Welcher User muss ich sein, welche Rechte sind nötig? Welche Seite muss ich aufrufen? Auf welchem System muss ich mich befinden?

Die **Entwickler-Informationen** sind vorhanden

💡 Gibt es noch Klärungsbedarf? Geklärtes sollte auch hier stehen, damit die Frage nicht mehrfach gestellt wird.

💡 Gibt es wichtige technische Informationen, die zur Bearbeitung des Tickets notwendig sind? Welche Stelle muss ich anfassen? Gibt es schon ähnliche Implementierungen?

Es ist ein **Screenshot** hinterlegt, der den betroffenen Bereich beschreibt

💡 Jira-Capture eignet sich gut, um Screenshots anzulegen und mit Informationen anzureichern

Es ist ein **Link** hinterlegt, der auf die betroffene Seite führt

Die **Akzeptanzkriterien** sind angelegt

Das Ticket steht auf *Bereit für's Backlog*

☑ Definition of Done (2.1)

Die Umsetzung ist abgeschlossen

Die **Akzeptanzkriterien** sind erfüllt

Die Deployment-Informationen sind geschrieben

Eine **Redakteurs- und Entwickler-Dokumentation** ist geschrieben, falls nötig

Ein **funktionales Review** ist durchgeführt

Das Ticket wurde **in allen aktuellen Browsern geprüft**, d.h. **IE 11** und **Edge**, **Firefox** innerhalb der letzten drei Versionen, **Chrome** und **Safari**

Ein Review der **Deployment-Informationen** ist durchgeführt

Das **Code-Review** ist durchgeführt

Der Feature-Branch ist in den *master* integriert

Der aktuelle *master*-Branch ist auf das Testsystem ausgerollt

Der Kunde ist informiert, das Ticket ihm zugewiesen und auf *Testsystem Abnahme*

Sicherheitskritische Informationen wurden dem Kunden übermittelt

☑ Definition of Done (2.1) (cont)

Behaviour-Tests wurden geschrieben und decken die Akzeptanzkriterien ab

Jenkins-Stages führen die automatisierten Tests erfolgreich aus und die Stage ist grün

☑ Code-Review

Der Code ist - wenn sinnvoll - in Gitlab kommentiert

Im Code ist kein `TODO` mehr zu finden

Suche dir Unterstützung, wenn das Review komplex ist

Die automatisierten Tests wurden lokal auf dem Feature-Branch ausgeführt

🔧 Manifesto for Software Craftmanship

Als engagierte Software-Handwerker heben wir die Messlatte für professionelle Softwareentwicklung an, indem wir üben und anderen dabei helfen, das Handwerk zu erlernen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Nicht nur funktionierende Software,
sondern **auch gut gefertigte Software**

Nicht nur auf Veränderung zu reagieren,
sondern **stets Mehrwert zu schaffen**

Nicht nur Individuen und Interaktionen,
sondern **auch eine Gemeinschaft aus Experten**

Nicht nur Zusammenarbeit mit dem Kunden,
sondern **auch produktive Partnerschaften**

Das heißt, beim Streben nach den Werten auf der linken Seite halten wir die Werte auf der rechten Seite für unverzichtbar.

❓ Hilfe zur Selbsthilfe

Ich brauche eine bestimmte Anleitung.

Hast du schon im **Confluence** geschaut? Alternativ gibt es nahe am Code in der Regel *readme*-Dateien, die deine gesuchte Anleitung beinhalten könnten.

Ich suche ein Passwort.

Anwendungsspezifische Passwörter findest du in der **Developer-Datenbank für Keepass**. Wenn du **Server-Passwörter** suchst, versuche es mit der Server-Datenbank.

Ich brauche eine technische Hilfestellung bezüglich einer Extension.

In den Extensions liegt meistens ein *Documentation*-Ordner mit einer *readme*.

Außerdem gibt es jeweils einen **Slack** für TYPO3 und Neos:

<http://neos-project.slack.com/>

<http://typo3.slack.com/>