

KM/QS Konzept: Aufgabenaufteilung auf Quellcodeebene

KM und PM:

- KM und PM stellen das Jira Board auf und erstellen in Jira Tickets und Labels für alle Arbeitspakete.
- KM überprüft regelmäßig die Versionierung in GitHub und unterstützt die anderen Teammitglieder bei Jira und Git.
- PM koordiniert die Aufgabenzuweisungen an SE und Überprüfungen an QS.

SE:

- SE teilen die Arbeitspakete in kleineren Aufgaben auf und erstellen dafür neue Tickets („child issues“), die sie mit dem entsprechenden Label des Arbeitspakets versehen.
- Zur Erarbeitung von Tickets dürfen SE nicht in der Main-Branch des Git-Repository arbeiten. Sie müssen stattdessen neue Branches erstellen und dort Änderungen vornehmen.
- SE müssen ein Pull-Request in GitHub für jedes neue Feature oder Änderung im Code einreichen und das entsprechende Ticket zur „Ausstehende QS-Überprüfung“ Spalte in Jira verschieben. SE können QS über die Anfragen auf Slack informieren, um den Prozess zu beschleunigen.
- SE können Tickets für QS erstellen und sie sowie weitere Tickets mit den folgenden Labels versehen:
 - **Rev-Req:** Anfrage für Review durch QS: Geschriebener Code noch nicht reviewed. Vor dem Testen muss mindestens einmal ein Review durchgeführt werden.
 - **Test-Req:** Anfrage den Code auf dem Testroboter durch QS zu testen.
 - **Other:** Andere Anfragen, die nicht um den Code gehen.
- SE verarbeiten nach jedem Review von QS ihren Code und führen für die neue Version denselben Vorgang durch bis QS feststellen, dass der Code keine Fehler mehr enthält.

QS:

- QS überprüfen das GitHub-Repository und das Jira-Board regelmäßig auf neue Pull-Requests bzw. neue Tickets.
- QS führen auf Anfrage Reviews auf GitHub durch. Die folgende Einstufung ist bei der Eingabe des Titels des Reviews zu beachten:
 - **Prototyp Review** (nur zu Beginn relevant): ausführliches Review durch QS, erfordert ein Skype QS-zu-SE-Gespräch.
 - **Hmm:** Kleinigkeit im Code die in keiner Weise dem Nutzer auffällt oder dem Ablauf des Systems schadet.

- **Hickup:** Ein Fehler oder Problem, das dem Nutzer auffallen kann oder das System beeinträchtigt jedoch nicht davon abhält die Aufgabe korrekt zu erfüllen.
 - **Oops:** Ein Fehler, der das System bei seiner Arbeit so beeinträchtigt, dass die Aufgabe nicht korrekt erfüllt wird oder ein Fehler, der es dem Nutzer unmöglich macht, ab Auftreten des Problems, das System zu bedienen.
 - **Fatal:** Ein Fehler, der so gravierend ist, dass in beinahe jedem Fall die Aufgabe nicht erfüllt werden kann.
- QS sind die einzigen, die Commits auf Pull-Requests durchführen dürfen.