

Sprint Retrospective – 1. Sprint

Im Großen und Ganzen war der erste Sprint eher chaotisch – vor allem die Branching-Aufteilung hat uns einiges an Nerven gekostet. Das Problem war, dass die Branches eine sehr hohe Abhängigkeit voneinander hatten und wir dadurch oft mergen mussten und mit Konflikten umgehen. Die „action“ bzw. die „lesson-learned“ davon ist, dass wir uns ab dem nächsten Sprint auf jeden Fall die Branches schon im Vorhinein besser unterteilen müssen und auf die Abhängigkeiten achten. Außerdem müssen die Schnittstellen noch genauer abgesprochen werden, damit es dann nicht zu Problemen kommt beim zusammenfügen / mergen.

Außerdem ist es ganz zu Beginn vorgekommen, dass zwei Personen auf verschiedenen Branches dieselben Teile implementiert haben. Wir lernen daraus, dass man sich (vor allem zu Beginn, aber auch später im Projekt) unbedingt genau absprechen muss, wer was erledigt.

Positiv im ersten Sprint haben wir die Teamkommunikation gefunden. Jeder konnte innerhalb von kurzer Zeit von einigen Personen eine Rückmeldung zu Fragen erhalten und außerdem war/ist die Bereitschaft groß, dass einem ein Projektpartner bei einem Problem auch gleich gemeinsam über Screensharing unterstützt.

Sprint Retrospective – 2. Sprint

Der zweite Sprint lief schon viel geordneter ab als der erste. Wir konnten die Abhängigkeiten zwischen den Branches eigentlich komplett eliminieren und dadurch wurde auch das mergen nicht zu so einem großen Problem wie noch beim ersten Sprint. Außerdem haben wir damit das Problem vom ersten Sprint, dass zwei Personen auf verschiedenen Branches denselben Teil implementierten, eliminieren können.

Ein Problem im zweiten Sprint stellte noch die mangelnde Verwendung von Redmine dar. Dadurch, dass wir unseren Code auf Github haben und die meisten Features von Redmine von uns nicht genutzt/bearbeitet werden, war nach außen hin (Tutoren / Assistentin) keine Transparenz gegeben. Wir müssen also in Zukunft immer darauf achten, dass wir alle Dokumente mit dem aktuellen Stand auf Redmine haben, und auch das Wiki sollte gepflegt werden und mit den aktuellen Dateien aktualisiert. Außerdem soll jede Person im Team wirklich alle Stunden loggen - auch solche, die nur zur Vorbereitung / Erarbeitung von Lösungsmöglichkeiten dienen und wo nicht direkt Code herauskommt.

Ein weiteres und eigentlich ziemlich großes Problem war, dass wir kurz vor Sprintende am master-Branch nicht lauffähigen Code hatten und wir den Fehler in so kurzer Zeit nicht finden und beheben konnten. Das darf auf keinen Fall in Zukunft noch einmal passieren.

Gut funktioniert hat, genauso wie im ersten Sprint, wieder die Teamkommunikation.

Sprint Retrospective – 3. Sprint

Im dritten Sprint hatten wir wieder die Aufteilung auf die Branches wieder sehr gut im Griff, auch deswegen, weil es zwischen den Branches nur sehr wenig Abhängigkeiten gegeben hat.

Ein Problem, auf das wir in diesem Sprint gestoßen sind, war die doch relativ ungenaue Spezifizierung der User Stories. Das haben wir aber dank der guten Kommunikation im Team auch relativ rasch lösen können und es kam zu keinen Folgeproblemen.

Nach dem 2. Sprint haben wir jetzt eigentlich wirklich alles in Redmine dokumentiert. Dazu zählt die Fortschaltung der Ticketstände, das kontinuierliche loggen der Zeit und auch das betreuen und aktuell halten des Wiki.

In diesem Sprint sind wir auch auf ein Problem beim Testen gestoßen, das wir leider bisher noch nicht beheben konnten. Wir speichern in unserer Datenbank in jeder Tabelle ab, wann derjenige Datensatz erstellt und auch das letzte Mal modifiziert wurde. Das Datum der letzten Modifikation wird dann auch verwendet um zum Beispiel den Ticketstatus zu einem Ticket zu „errechnen“. Bisher konnten wir das alles ganz gut testen, aber um den Top 10 Filter (im Balkendiagramm) zu testen, bräuchten wir Ticket Transactions die schon einige Zeit in der Vergangenheit liegen. Wir haben es leider nicht geschafft, diese Daten zu manipulieren.

Außerdem haben sich besonders in diesem Sprint (aber auch schon im vorherigen) kleinere Gruppen herausgebildet die bestimmte User Stories übernommen haben. Die Zuteilung zu diesem Sprint haben wir schon nach diesen Kleingruppen gemacht und das hat sehr gut funktioniert. Wir haben untereinander die Schnittstellen im Vorhinein abgesprochen und sind da zu keinen Problemen gekommen.

Sprint Retrospective – 4. Sprint

Im letzten Sprint hatten wir nur mehr die Zahlungsarchitektur die von einer Person gemacht wurde und die Nutzerverwaltung als große Punkte. Dabei gab es keine Abhängigkeiten unter den Branches mehr und die PRs sind eindeutig viel kleiner und übersichtlicher gewesen.

Einzig bei der finalen Erstellung der Testdatengeneratoren hat es kleinere Probleme gegeben, weil Personen die Datengeneratoren von anderen geändert haben. Dadurch sind dann aber wieder andere Tests fehlgeschlagen. Deswegen haben wir uns dann darauf geeinigt, dass man bei Änderungen einfach einen neuen Datengenerator schreiben soll der unter Umständen auf dem bisherigen aufbauen kann / soll.