

Sprinttagebuch

Key Elements eines Sprinteintrages:

1. Sprint:

- Kunde anlegen/bearbeiten
- Events + Performances anzeigen, selektieren
- Saalplan für Sektoren, weiter zur Kundenauswahl und Detailansicht der Transaktion
- Transaktionsansicht generell mit Suche

2. Sprint:

- Pagination der Kunden-, Events- und Transaktionslisten inklusive der Kunden- und Transaktionssuche
- Ticketverkauf / -reservierung / -stornierung sowohl serverseitig als auch clientseitig erstellen und anbinden
- Saalplan für Sitzplätze inklusive der Abwahl bereits ausgewählter Tickets

3. Sprint:

- PDF Druck von Rechnungen und Stornorechnungen
- Suchen und Filtern der Veranstaltungen
- News erstellen und anzeigen

4. Sprint:

- Zahlungsarchitektur
- Nutzerverwaltung
- Suchen und Filtern der Veranstaltungen abschließen

Ergebnisse der Sprint Retrospective:

1. Sprint

- Aufteilung in die Branches – Abhängigkeiten zwischen den Branches waren zu hoch, dadurch haben wir die Branches untereinander immer wieder gemerged was dann im Endeffekt zu einem großen Branch am letzten Tag geführt hat mit dauernden Konflikten.
 - Maßnahme: Branches vorher schon klar definieren und vor allem auch Schnittstellen klar definieren
- Aufteilung der Arbeiten klarer absprechen – zu Beginn ist es ein paar Mal vorgekommen, dass 2 Leute dieselbe Arbeit auf ihren Branches gemacht haben wegen mangelnder Absprache
 - Maßnahme: das ganze sollte jetzt nicht mehr auftreten, war eher ein Problem zu Beginn – aber auch jetzt sollte jeder wirklich nur in seinem Branch seine vorher ausgemachten Tätigkeiten erledigen und nur dann Änderungen in den Zuständigkeiten einer anderen Person erledigt, wenn er Bescheid gibt und das abspricht.

2. Sprint

- nur sehr sporadische Verwendung von Redmine
 - Maßnahme: wir haben sofort nach dem MR2 (=Sprintende) damit begonnen, ein Wiki in Redmine aufzubauen und alle Dateien mit aktuellem Stand darin zu verlinken. Jeder im Team ist ab jetzt dazu angehalten, seine Dateien immer in Redmine aktuell zu halten und die Links zu aktualisieren
- Nicht lauffähiger Code nach dem Sprintende am master-branch ist nicht akzeptabel
 - Maßnahme: bevor der Code dann wirklich auf den master-branch gepushed wird, sollte eine Person den Code nochmal außerhalb von IntelliJ und jeder Entwicklungsumgebung nur über die Command-Line manuell kompilieren und testen.

3. Sprint

- realistische Testdaten
 - wir haben das Problem, dass wir vom Programm aus nicht die automatisch generierten LastModifiedAt und CreatedAt Timestamps in der Datenbank zu manipulieren, was ja grundsätzlich gut ist, aber für die Tests nicht sehr hilfreich. Als Maßnahme könnten wir ein SQL Script schreiben, das uns alle gewünschten Daten direkt in die Datenbank einfügt falls so etwas überhaupt in unserer Hibernate / Spring Einstellung möglich ist.

4. Sprint

- im letzten Sprint gab es keine wirklichen Probleme mehr (es war auch nur mehr weniger zu programmieren) und deswegen auch keine Ergebnisse