

Was war gut:

Team:

Vorbereitung:

Gut war, dass nicht sofort mit der Entwicklung begonnen wurde, sondern zunächst die eingesetzten Werkzeuge (Trello, Github, Codeigniter, Testdriven Development, SCRUM) von Mitgliedern vorgestellt wurden. Dabei konnte jedes Mitglied ein Thema auswählen.

Was war schlecht:

Team:

zu wenig Kommunikation:

Eine offene Kommunikation im Projektteam ist unerlässlich. Leider hat diese am Anfang überhaupt nicht funktioniert. Dies wurde besonders dadurch klar, dass die Verantwortlichen für den Controller, ohne jegliche Rücksprache, einfach ihr eigenes Model implementierten, obwohl dieses Aufgabenfeld anderen Personen zugeteilt worden war. Ein weiteres Problem war, dass die Richtung, in die das Projekt gehen sollte, nicht ausreichend im Team kommuniziert wurde. Vielmehr wurden die Tasks im 1. Sprint durch Scrum-Master und Product-Owner explizit vorgegeben. Hier sollte das Team mehr Einfluss darauf haben, wann welche Features implementiert werden.

Zu starke Ausprägung von Unter-Teams:

Da es in jedem Team verschiedene Rollen und Spezialgebiete von einzelnen Mitgliedern gibt und auch geben sollte, steht außer Frage. Allerdings ist es bei SCRUM besonders wichtig, dass jedes Mitglied gleichberechtigt ist. Es gibt dort keine unterschiedlichen Titelbezeichnungen, sondern nur den einen Titel "Team-Mitglied". Daher sollte es keine feste Einteilung in Unterteams, wie Model View und Controller, geben. Dies führt bei Besprechungen zu dem Problem, dass nicht mehr alle Team-Mitglieder aufmerksam zuhören. Dadurch gibt es keine richtige Diskussion zu entsprechenden Themenpunkten. Außerdem bleibt das Wissen des einzelnen dann auf das jeweilige Aufgabengebiet beschränkt.

Diese starren Unterteams führten auch sofort zu einem Problem. Dadurch war es beispielsweise nicht für alle Mitglieder möglich (Controller, View) zur gleichen Zeit mit der Entwicklung anzufangen, da diese zumindest teilweise ein funktionierendes Modell voraussetzen.

SCRUM:

Bei SCRUM werden Anforderungen und die dazugehörigen Lösungen erst im Verlauf des Projekts konkretisiert. Dadurch ist es relativ schwierig, am Anfang ein geeignetes Architekturkonzept zu erstellen. Laut SCRUM soll man sich gerade zu Dinge wie Erweiterbarkeit von Software keine großen Gedanken machen. Muss das Architekturkonzept später geändert werden, ist das aber meistens sehr zeitaufwändig und der Programmcode muss evtl. nahezu komplett neu geschrieben werden. Ein konkretes Beispiel dafür war das Post-Model, das durch die Anforderung für eine "Historie von Posts" umgeschrieben werden musste.