

5. Abstract

Das Projekt startete mit dem Ziel, eine Verwaltungssoftware für Sporttrainer zu erstellen, mit der Trainingseinheiten vorbereitet und geplant werden können. Dieses Ziel wurde über 3 Phasen hinweg in eine Anwendung umgesetzt, die in Python programmiert wurde. Dazu wurden nach dem Scrum Softwareprozess fünf Sprints durchgeführt, die jeweils eine Woche dauerten. Im Folgenden werden einige Aspekte des Projekts noch einmal genauer beleuchtet und kritisch reflektiert.

Direkt zu Beginn des Projekts wurden einige konzeptuelle Entscheidungen getroffen, die im späteren Projektverlauf überarbeitet werden mussten. Die meisten hiervon wurden jedoch bereits von Anfang an als Projektrisiken festgehalten, sodass die Alternativlösung jeweils keinen Zusatzaufwand bedeutete. So wurde zum einen die geplante MySQL-Datenbank durch eine SQLite-Datenbank ersetzt, da keine Userverwaltung notwendig war und somit Komplexität eingespart werden konnte. Zum anderen wurde die ursprünglich geplante Klassenstruktur verallgemeinert, sodass weniger Inhalte hart codiert werden mussten. Die anfängliche Idee, Videodateien direkt in die Datenbank hochzuladen, wurde ebenfalls nicht umgesetzt und durch einen einfachen URL-Link ersetzt, da die Datenbank bzw. das Anwendungsverzeichnis zu groß geworden wäre. Der geplante Export eines Plans als PDF hatte zum Ende der Phase 2 noch Probleme bereitet, konnte in der Finalisierungsphase dann aber noch umgesetzt werden. Für Phase 2 wurde hier eine Zwischenlösung eingeführt, die lediglich einen Screenshot der Anwendung erstellt und abspeichert. Diese Zwischenlösung ist auch weiterhin für die anderen Objekte im Einsatz. Durch das Feedback nach Phase 2 konnte die Benutzerfreundlichkeit und Intuitivität der Anwendung noch einmal überarbeitet und stark verbessert werden. Hierfür wurde in Phase 2 zu wenig Zeit eingeplant, stattdessen wurde mehr auf technische Funktionalität geachtet. Dieses Projektrisiko war ursprünglich nicht eingeplant, daher musste hierfür noch eine spontane Lösung gefunden werden. Die Definition der Projektrisiken half von Anfang an dabei, schnell auf unerwartete Probleme reagieren zu können und sollte in zukünftigen Projekten weiter fortgeführt werden.

Der in Phase 1 erstellte Zeitplan war sehr grob und konnte in Phase 2 detaillierter ausgearbeitet werden. Die am Anfang der Phase definierten Termine konnten bis zur Abgabe der Phase 2 alle eingehalten werden. Die für Phase 3 definierten Aufgabenpakete wurden nicht ganz in der Reihenfolge erstellt, da zum einen noch ein fünfter Sprint durchgeführt wurde und hier z.T. ein paralleles Bearbeiten der Anwendung und der Dokumentation stattfand. Die erwarteten Termine der Meilensteine waren korrekt abgeschätzt, mit Ausnahme des Meilensteins „Feature-Complete“, welcher erst eine Woche später als geplant erreicht wurde.

Die funktionalen Anforderungen und User Stories waren gut definiert und leiteten klar durch den gesamten Entwicklungsprozess. Bei Konflikten half die Einteilung in obligatorische und optionale Anforderungen sehr dabei, die Konflikte zu lösen und die Prioritäten richtig zu setzen. Hierzu passte auch der gewählte Softwareprozess Scrum, der den Fokus auf den Kundenanforderungen und funktionierenden Code legt. Diese agilen Kerngedanken haben sich durch die gesamte Entwicklungsphase und alle fünf Sprints gezogen. Die Organisation in Sprints half dabei, die Aufgabenpakete in

umsetzbare Untereinheiten aufzuteilen und für jeden Tag des Sprints einen Plan zu haben, was umgesetzt werden soll. Aus dem definierten Ziel eines jeden Sprints ließen sich dann spontan auch weitere Aufgabenpakete ableiten, die bei der Planung des Sprints noch nicht ersichtlich waren. Die Menge an erledigter Arbeit wurde insgesamt erhöht, wobei immer darauf geachtet wurde, so wenig wie möglich unnötige Arbeit zu verrichten. Das bedeutet, dass Dinge, die „man vielleicht mal brauchen könnte“, weggelassen wurden und sich immer darauf konzentriert wurde, das Sprintziel zu erreichen.

Das geplante regelmäßige „Daily“ Meeting aus Scrum wurde in der Form durchgeführt, dass zu Beginn eines Tages die noch offenen Aufgabenpakete aufgeschrieben wurden oder neue Pakete aus dem Sprintziel abgeleitet wurden. Das „Sprint Planning“ Meeting wurde meistens erst am ersten Tag des Sprints durchgeführt, was ein paar Zeitverzögerungen für diesen Tag mit sich brachte. Das „Sprint Review“ Meeting bestand darin, die Ergebnisse des Sprints in GitHub in den main-Branch zu mergen und kurz zusammenzufassen, was in diesem Sprint erledigt wurde. Falls noch offene Aufgabenpakete bestanden, wurden diese für den folgenden Sprint vorgemerkt. Die Verwaltung des Sprint Backlogs bzw. Product Backlogs hat gut funktioniert, dadurch war zu jedem Zeitpunkt klar, welche Aufgabenpakete abgeschlossen sind und wo noch etwas zu erledigen ist.

Beim Entwurf der Anwendungsstruktur wurde konsequent darauf geachtet, das MVC-Pattern umzusetzen. Dies hat für die Datenlogik problemlos funktioniert, jedoch gab es bei der Differenzierung zwischen Businesslogik und Visualisierung einige Konflikte, welche die Anwendung komplexer macht als notwendig. Außerdem musste der Grundsatz, so wenig wie möglich hart zu programmieren, bei der Implementierung des PDF-Exports eines Plans gebrochen werden. Das Singleton-Pattern, das für die Datenbankbindung zum Einsatz kam, spielt in der aktuellen Auslieferung der Anwendung kaum eine Rolle, da die Anwendung sowieso immer nur eine Datenbank und ein Frontend ansprechen kann. Dennoch konnte hier ein persönlicher Lerneffekt erzielt werden, wie man in Python das Singleton-Pattern umsetzen kann.

Der Anwendungscode ist insgesamt nicht unnötig aufgeblasen, jedoch eher unstrukturiert in den einzelnen Modulen. Hier wäre ein Refactoring notwendig, um die relevanten Methoden schnell und einfach zu finden, damit auch bei zukünftigen Entwicklungen keine Doppelarbeit gemacht wird. Bei zukünftigen Projekten sollte darauf geachtet werden, das Refactoring direkt beim Einfügen neuer Methoden durchzuführen, sodass der Code immer gut strukturiert ist.

Die definierten Testfälle haben gegen Ende der Entwicklungsarbeit dabei geholfen, die Anwendung in natürlichen Worten beschreiben zu können und genau zu definieren, welches Programmverhalten erwartet wird. Außerdem konnten hierdurch noch einige Bugs ausfindig gemacht und ausgeremert werden. Dies sollte für zukünftige Projekte beibehalten und weiter ausgebaut werden.

Der Aufbau der Anwendung als ausführbare Datei wurde erst spät im Projekt berücksichtigt und hat einige Hürden bereithalten, die schlussendlich aber noch gemeistert werden konnten. Der persönliche Lerneffekt hieraus wird in weitere Projekte getragen werden, bei denen die Auslieferung der Anwendung direkt im Konzept und in der Projektplanung berücksichtigt werden kann.