

Extreme Programming

Die Suche nach einem geeigneten Vorgehensmodell war für uns nicht ganz einfach. Durch unsere Entscheidung, nur auf OpenSource Produkte zu setzen wussten wir, dass wir dadurch sehr großes Neuland für alle von uns betreten würden. Dadurch stand auch fest, dass unser Projekt sehr experimentell ablaufen würde und wir zumindest zu Beginn keine vollständige Architekturplanung durchführen konnten. Es waren einfach zu viele Unklarheiten über die Möglichkeiten in wxWidgets gegeben. Leider ließen es die Umstände auch nicht zu, uns bereits in der Zeit vor dem wirklichen Projektstart mit all den Neuheiten auseinander zu setzen.

So viel unsere Entscheidung mit XP auf ein Vorgehensmodell, mit dem wir sehr agil vorgehen konnten und ohne große Planung starten konnten. Ebenso passte dieses Modell gut zu unserer Idee, in diesem Projekt möglichst viele neue Sachen auszuprobieren, denen man sonst nicht jeden Tag begegnet.

Im folgenden wollen wir nur kurz auf die wichtigsten Praktiken von XP, die auch von uns umgesetzt wurden, eingehen und danach noch kurz unsere durchgeführten Anpassungen aufzeigen.

Das Prozessmodell

Bei den **UserStories** haben wir gleich zu Beginn einmal nach groben Anwendungsgebieten gesucht und diese erst einmal grob ausgeführt. Im Laufe der Zeit wurden dann zu jedem Gebiet die exakten UserStories ausgearbeitet.

Zu Beginn jeder Woche und damit auch zu Beginn von jeder neuen **Iteration** haben wir anhand der UserStories das Vorhaben für die nächste Woche geplant und auf die einzelnen Programmierer verteilt.

Permanente Integration war für uns auch ein sehr wichtiger Punkt, da wir durch unsere kleine Teamgröße immer darauf angewiesen waren, über die Änderungen anderer schnellst möglich informiert zu werden, damit sich Unstimmigkeiten möglichst schnell beseitigen ließen. Durch unser Repository stellte dies kein Problem dar.

Im Einklang mit permanenter Integration kam für uns auch das Prinzip des **kollektiven Eigentums**. Keine Code-Ownership zu haben bedeutete für uns, dass wir notwendige Änderungen sofort durchführen konnten. Wichtig hierfür war einzig eine ausführliche Kommentierung des Quellcodes, was aber auch Teil des XP-Prozesses war.

Coding Standards wurden gleich zu Beginn des Projekts festgesetzt und auch durch ein Tool wie aStyle erleichtert. Dies ermöglichte uns ein schnelles Zurechtfinden in fremden Code.

In regelmäßigen Abständen wurden ausgewählte Teile einem kurzen **Codereview** unterzogen, wodurch wir versucht haben die Qualität unseres Codes zu verbessern und alle Team-Mitglieder über den aktuellen Stand der Software zu halten.

Einfaches Design kam uns auch sehr stark zu Nutzen, da sich unser 5. Mann, der sich eigentlich um Softwaredesign gekümmert hatte, einen Tag vor Projektbeginn von unserem Team abmeldete. Dieses Design ermöglichte es uns, ohne große Planung direkt in die Materie einzusteigen und gleich mit funktionierendem Code zu beginnen. Damit verbunden war für uns dann regelmäßiges **Refactoring**, wodurch versucht wurde die Qualität des Codes ständig zu verbessern und die Struktur an aktuelle Gegebenheiten anzupassen.

Das Prinzip der **Kundeneinbeziehung** wurde zumindest durch die wöchentlichen Statusmeetings eingehalten.

Den wohl wichtigsten Punkt in XP, nämlich die enge **Kommunikation** zwischen den Teammitgliedern wurde durch das stetige Arbeiten an der Hochschule selbst stark erleichtert.

Die Anpassungen

Leider konnten wir nicht alle Prinzipien von XP umsetzen, da wir manche für nicht sinnvoll in einem so kurzen Projekt hielten.

Pairprogramming war leider nur in sehr kritischen Bereichen bzw. bei größeren Problemen möglich, da wir sonst einfach zu wenige aktive Programmierer gewesen wären um die vielen Anforderungen umzusetzen.

Ebenso hatten wir nicht die Zeit um vollständige **Unit Tests** zu schreiben. Für kritische Klassen haben wir darum automatisierte Tests durchgeführt um wenigstens einmal einen Eindruck von Unit Tests und deren Vorteile zu bekommen.

Die **Kundeneinbeziehung** war auch nicht so möglich, wie es vom Prozess eigentlich gedacht war, da kein richtiger Kunde vorhanden war.

Die wohl größte Anpassung war die Abschaffung des **Überstundenverbots**. Dies konnte durch die Teamgröße von nur 4 Mann unter keinen Umständen eingehalten werden.