

Aufgabe 14-1: (Pair Programming)

Eine der in der Vorlesung besprochenen Praktiken des Extreme Programming ist das Pair Programming (PP). Beim PP arbeiten zwei Entwickler an einem Rechner mit einer Maus und einer Tastatur. Nach (*Strengthening The Case Of Pair Programming*, Laurie Williams, Robert R. Kessler, Ward Cunningham und Ron Jeffries; 2000) ist PP wie folgt definiert:

Zwei Entwickler erstellen gemeinsam ein Artefakt. Dabei agieren sie wie ein zusammenhängender, intelligenter Organismus, der mit einem Verstand arbeitet und verantwortlich für jeden Aspekt des Artefakts ist.

Wobei derjenige, der die Kontrolle über die Maus und die Tastatur hat, als Driver bezeichnet wird und das Design bzw. den Code schreibt. Der Partner ist der Observer und beobachtet kontinuierlich die Arbeit des Drivers. Der Observer sucht nach Fehlern, denkt über Alternativen nach, beobachtet die Ressourcen und betrachtet "strategische" Auswirkungen.

Die Rollen werden bewusst periodisch getauscht.

Beide Partner sind gleichberechtigte und aktive Teilnehmer am Prozess und teilen sich den Besitz am Arbeitsergebnis.

Häufig stehen erfahrene Programmierer und Projektleiter dieser Praktik zuerst etwas skeptisch gegenüber. Sie sorgen sich unter anderem über den erhöhten Kommunikationsaufwand, über die unterschiedlichen Programmierstile, die Verdopplung der Personenstunden und über Uneinigkeiten der Entwickler bezüglich der Implementierung. Im Nachhinein beschreiben allerdings viele Entwickler das PP als eine positive Erfahrung.

Folgende Vorteile werden häufig im Zusammenhang mit PP in der Literatur beschrieben und durch Aussagen von Entwickler gestützt:

1. **Qualität:** Unter dem Begriff der Qualität werden in diesem Zusammenhang die Verringerung von Fehlern und die Verbesserung des Designs sowie der Lesbarkeit verstanden. Ein anonymes Programmierer sagte bei einer Umfrage von L. Williams:
I strongly feel pair-programming is the primary reason our team has been successful. It has given us a very high level of code quality (almost to the point of zero defects). The only code we have ever had errors in was code that wasn't pair programmed ... we should really question a situation where it isn't utilized.
2. **Zeit:** Der Faktor Zeit spielt in der Softwareentwicklung eine entscheidende Rolle. Es wird zwischen Personenstunden und Entwicklungszeit unterschieden. Während die Personenstunden sich durch PP erhöhen, belegen einige Studien, dass die Entwicklungszeit durch den Einsatz von PP abnimmt. Kent Beck:
Meiner Erfahrung nach ist es produktiver, paarweise zu programmieren, als die Arbeit zwischen zwei Programmierern aufzuteilen und die Ergebnisse dann zusammenzuführen.
3. **Motivation:** Die meisten Umfragen bestätigen, dass den Programmierern PP mehr Spaß macht als Solo-Programming. Sie sind zuversichtlicher, höhere Qualität zu erreichen, und PP steigert das Vertrauen in ihre eigenen Programmierfähigkeiten. Zudem arbeiten Programmierer im Paar meist konzentrierter. Wieder aus der Umfrage von L. Williams:
When I worked on the machine as the driver, I concentrated highly on my work. I wanted to show my talent and quality work to my partner. When I was doing it, I felt more confident. In addition, when I had a person observing my work, I felt that I could depend on him, since this person cared about my work ... When I was the non-driver, I proofread everything my partner typed. I felt I had a strong responsibility to prevent any errors in our work.
4. **Wissenstransfer:** PP ist sehr kommunikativ, so dass die Entwickler ihr Wissen untereinander weitergeben. Dies ermöglicht unter anderem eine schnelle Einarbeitung neuer Projektmitglieder, in dem ein Neuling mit einem Experten programmiert, wie auch einen kontinuierlichen Lernprozess der Mitarbeiter. In der Praxis wird der Wissenstransfer durch Rotation (Wechseln der Konstellation der Paare, also vom Driver zum Observer und andersherum) noch optimiert.

In den Tutorien der letzten Woche werden wir das Pair Programming ausprobieren. Es wird eine kleine Programmieraufgabe zu erledigen sein – im Paar natürlich. Die Einteilung in die Paare findet in der Woche zuvor statt. Eventuell möchten Sie sich schon vorher mit einem Partner/ einer Partnerin absprechen.

Die Tutorien finden in der Woche 12.2. bis 15.2. daher *nicht* in den Seminarräumen statt, sondern in den Windows-Rechnerräumen im Keller. Beachten Sie dazu unbedingt die Aufteilung auf die Rechnerräume während des nächsten Tutoriums und ggf. via Mailingliste. Die Tutorien beginnen schon zur vollen Stunde, also s.t., nicht wie üblich c.t.

Wir werden in der Übung die Entwicklungsumgebung Eclipse einsetzen. Sie sollten Eclipse starten, ein Java-Projekt anlegen, eine Klasse implementieren und das Java-Programm testen können. Es gibt reichlich guter Anleitungen im Netz: <http://www.google.de/search?q=eclipse+tutorial>.

Bitte machen Sie sich für die Übung mit Eclipse ausreichend vertraut!

Aufgabe 14-2:

Wir werden in der vorletzten Vorlesungswoche, also vom 5.2. bis zum 8.2. ausgewählte Aufgaben der regulären Übungsblätter besprechen. Bitte schauen Sie sich an, bei welchen Themen und Aufgaben Sie sich besonders unsicher fühlen oder wessen Lösung Sie nicht verstanden haben. Nennen Sie Ihrem/-r Tutor/-in diese Aufgaben, so dass er/sie sich darauf vorbereiten kann.

Zudem gibt auf der Website ein Extra-Übungsblatt zur Klausurvorbereitung, deren Lösungshinweise erst Montag vor der Klausur veröffentlicht werden. Auf der Website finden Sie auch allgemeine Hinweise zur Klausur.

Aufgabe 14-3:

Nehmen Sie an der Lehrevaluation teil. Eine Mail dazu dürfte vom Evaluationsteam bald verschickt werden. Auf Ihre Rückmeldung (Lob, Kritik, Verbesserungsvorschläge, Weiter-so-Parolen) sind wir sehr gespannt! Insbesondere interessiert uns, wie Sie die Umstellung auf die Gruppenarbeit in den Tutorien und die dort benutzen Aufgaben beurteilen.