Empirische Untersuchungen des Side-by-Side Programming

Ulrich Stärk

Institut für Informatik

FU Berlin

25.06.2008

Gliederung

- Side-by-Side Programming
- Motivation
- Workshop und Experiment
- GT
- Blickwinkel
- Vorgehen
- Ergebnisse

Pair Programming

- Gemeinsame Arbeit an einem Rechner
  - Nur eine Maus und Tastatur
- Viele Studien haben Vorteile gefunden
  - Schneller als Soloprogrammierer (Nosek, Cockburn u. Williams, Williams et al.)
  - Weniger Defekte (Cockburn u. Williams, Williams et al.)
  - Glücklichere Programmierer (Williams u. Kessler)
  - Wissenstransfer, voneinander Lernen (Williams u. Kessler)
- Allerdings auch mögliche Nachteile
  - Zu intensiv (Cockburn)
  - Stumpfes Programmieren wird lieber alleine gemacht (Williams et al.)
  - Unwohlsein, z.B. durch zu geringen sozialen Abstand, Körperhygiene, persönliche Gefühle (Beck)
  - Starre Rollenaufteilung (Nawrocki et al.)
- Hauptsächlich quantitative Untersuchungen
  - Prozesse weitgehend unerforscht, werden derzeit von Stephan Salinger untersucht

Side-by-Side Programming

- Eine von Alistair Cockburn in Crystal Clear vorgeschlagene Technik
  - Alistair Cockburn. Crystal Clear. Addison-Wesley, 2004.
    - Beschreibt Sicherheitsmaßnahmen um einen erfolgreichen Ausgang eines Projekts zu unterstützen

- 2 Programmierer arbeiten unmittelbar nebeneinander
  - Jeder arbeitet an seiner Aufgabe
  - Bildschirm des jeweils anderen leicht einsehbar
  - Zusammenkünfte möglich um Fehler zu suchen, gemeinsam ein Stück Code zu schreiben, etc.

- Weniger invasive Alternative zu Pair Programming
  - „traditionelle“ Arbeitsweise
    - Jeder Entwickler arbeitet für sich am eigenen Computer
  - Trotzdem „Osmotische Kommunikation“ erwartet
    - Wissenstransfer durch z.B. zufälliges Mithören oder Mitbekommen von Tätigkeiten des Anderen

Stand der Forschung

- Bis auf ein Paper keinerlei wissenschaftliche Arbeiten darüber
- Jerzy R. Nawrocki, Michal Jasinski, Lukasz Olek, and Barbara Lange. Pair Programming vs. Side-by-Side Programming. Lecture Notes in Computer Science, 3792:28-38, 2005.
- Quantitativer Vergleich von Soloprogrammierung, Pair Programming und Side-By-Side Programming
  - Side-By-Side Programming schneller als Solo- und Pair Programming
    - Gleiche Aufgabe in kürzerer Zeit erledigt (Solo 100%, PP 74%, SBS 61%)
    - Somit nur 22% Overhead beim SBSP hingegen 48% beim PP im Vergleich zu Solo Programmierung
  - Allerdings vermutlich weniger Kenntnis vom gesamten Quellcode als beim Pair Programming
    - Umsetzung eines Change Requests dauerte länger als beim Pair und Solo Programming
    - Schwache Hypothese

Motivation und Ziele

- Beitrag zum Wissen über Side-by-Side Programming
  - Bisher nur eine Studie, die das Thema untersucht
  - Erforschung der Prozesse und Abläufe beim Side-by-Side Programming
- Entwicklung einer Theorie über die Vorgänge beim SbS-Programming
  - Blickwinkel auf die Umstände, die zu Trennung / Zusammenkunft führen
- Durch Auswertung von Audio- und Videoaufzeichnungen von Programmierern
  - Müssen in einem Experiment gewonnen werden
- Mittels Grounded Theory
  - Völlig unbekanntes Terrain, daher explorative Methode erforderlich

Grounded Theory

- Entwicklung von in Daten begründeten Theorien
- Entwickelt von Strauss und Glaser Mitte 1960
- 3 Schritte
  - Kodieren der Daten
    - Vergabe von Codes für die beobachteten Konzepte
    - Dabei ständige Überprüfung mit bereits kodierten Phänomenen
  - Identifizierung von Beziehungen zwischen den Konzepten
  - Formulierung einer Theorie aus den Konzepten und ihren Beziehungen zueinander
- Systematische und nachvollziehbare Forschungsmethode
  - Für qualitative Auswertung
  - Explorativ
    - Keine vorherige Annahmen über das Ergebnis
      - Allerdings Einschränkung des Forschungsgegenstandes mittels geeigneter Fragestellung

Vorgehen

- Workshop
- Experiment
- Auswertung/Kodieren
- Theorienbildung

Workshop

- Workshop zum Thema Webentwicklung mit Java
  - 4 Tage
  - 10 Teilnehmer aus höheren Semestern
    - Komplexer Stoff
  - 5 Paare
- Einführung in die Frameworks Hibernate, Spring und Tapestry
  - Einführungsvortrag, dann Übungen
- Entwicklung einer dreischichtigen webbasierten Bibliotheksanwendung
- Probanden praktizierten schon während des Workshops Side-By-Side Programming ohne es explizit gesagt zu bekommen
  - Enge Zeitvorgaben so dass sich Trennung und parallele Bearbeitung anboten
  - Jedes Paar hatte 2 Rechner nebeneinander zur Verfügung

Experiment

- Am letzten Tag des Workshops
- Erweiterung der Bibliotheksanwendung
  - Aufgabe war umfangreich und mit guten Möglichkeiten zur parallelen Bearbeitung
    - Bot Gelegenheit sich zu trennen und wieder zusammen zu kommen
- Arbeit wieder im Paar
  - 2 Rechner direkt nebeneinander
  - Blick auf den Bildschirm des anderen leicht möglich
- Aufzeichnung von Audio und Video der Probanden sowie des Bildschirminhalts
- Nach dem Experiment: Beantwortung von 2 Fragebögen

Blickwinkel

- Blickwinkel auf Trennung bzw. Zusammenkunft der Probanden
- Was führt zu Trennungen / Zusammenkünften
  - Arten der Zusammenkunft
- Aufgabe lässt sich in zu erledigende Schritte einteilen
  - Versuch diese Schritte in der Kodierung wieder zu finden
  - Evtl. hängt das Trennungsverhalten mit diesen Schritten zusammen
- Eigenschaften der Personen könnten Einfluss auf Trennungsverhalten haben
  - Verhältnis zueinander
  - Kenntnisse

Blickwinkel (2)

- Kodierung anhand des Blickwinkels
  - Kodiere nur, was im Blickwinkel liegt
    - Also Trennung / Zusammenkunft und was davor und danach geschieht
  - Aufbauend auf das Kodierschema von Stephan
    - Untersucht Pair Programming
    - Pair Programming ähnlich einer Zusammenkunft beim Side-by-Side Programming

Paarzusammensetzung

- 3 Paare

- 1 Paar mit leichtem Unterschied beim Kenntnisstand
  - Empfand die Zusammenarbeit als hilfreich und harmonisch
  - Hat vorher nicht zusammengearbeitet

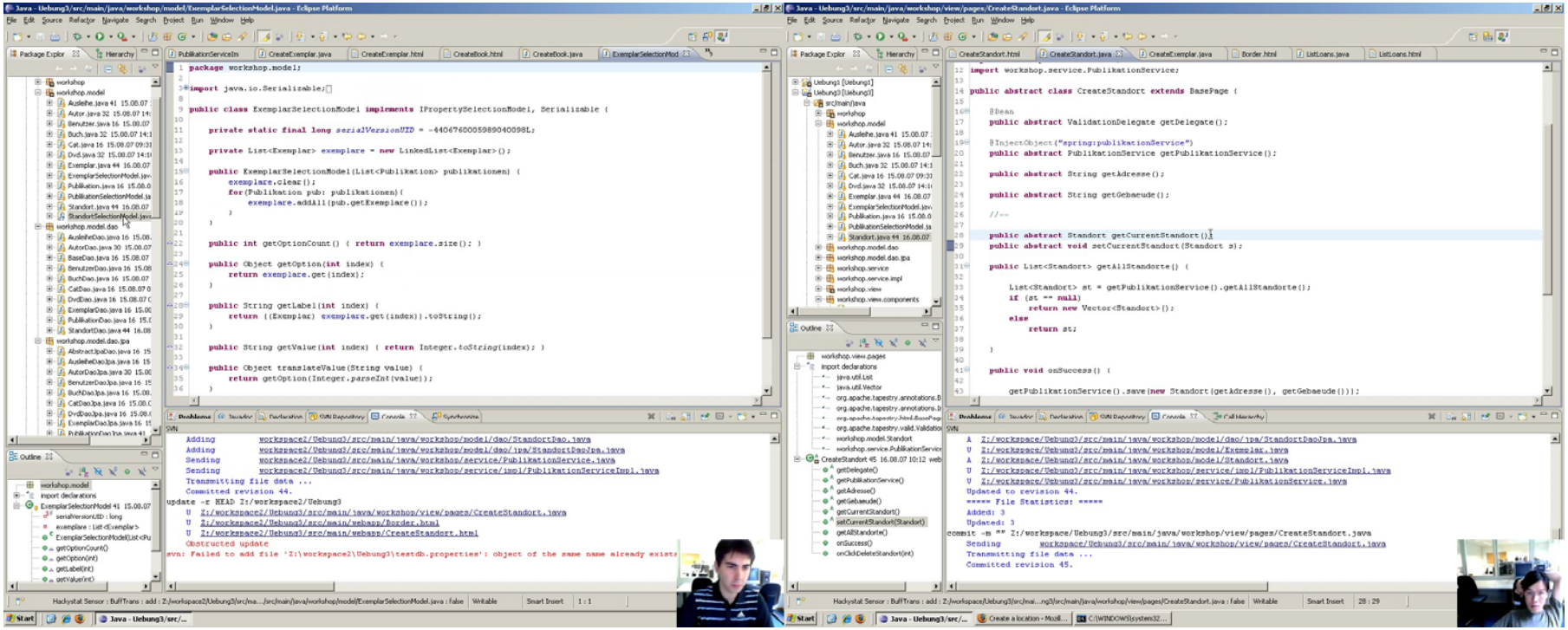
- 1 Paare mit etwa gleichem Kenntnisstand
  - Hat schon einmal zusammengearbeitet
  - Empfand die Zusammenarbeit als hilfreich und harmonisch

- 1 Paar mit großem Unterschied im Kenntnisstand
  - Bewertete die Zusammenarbeit als nicht so harmonisch wie die anderen Paare
  - Hat vorher noch nicht zusammengearbeitet

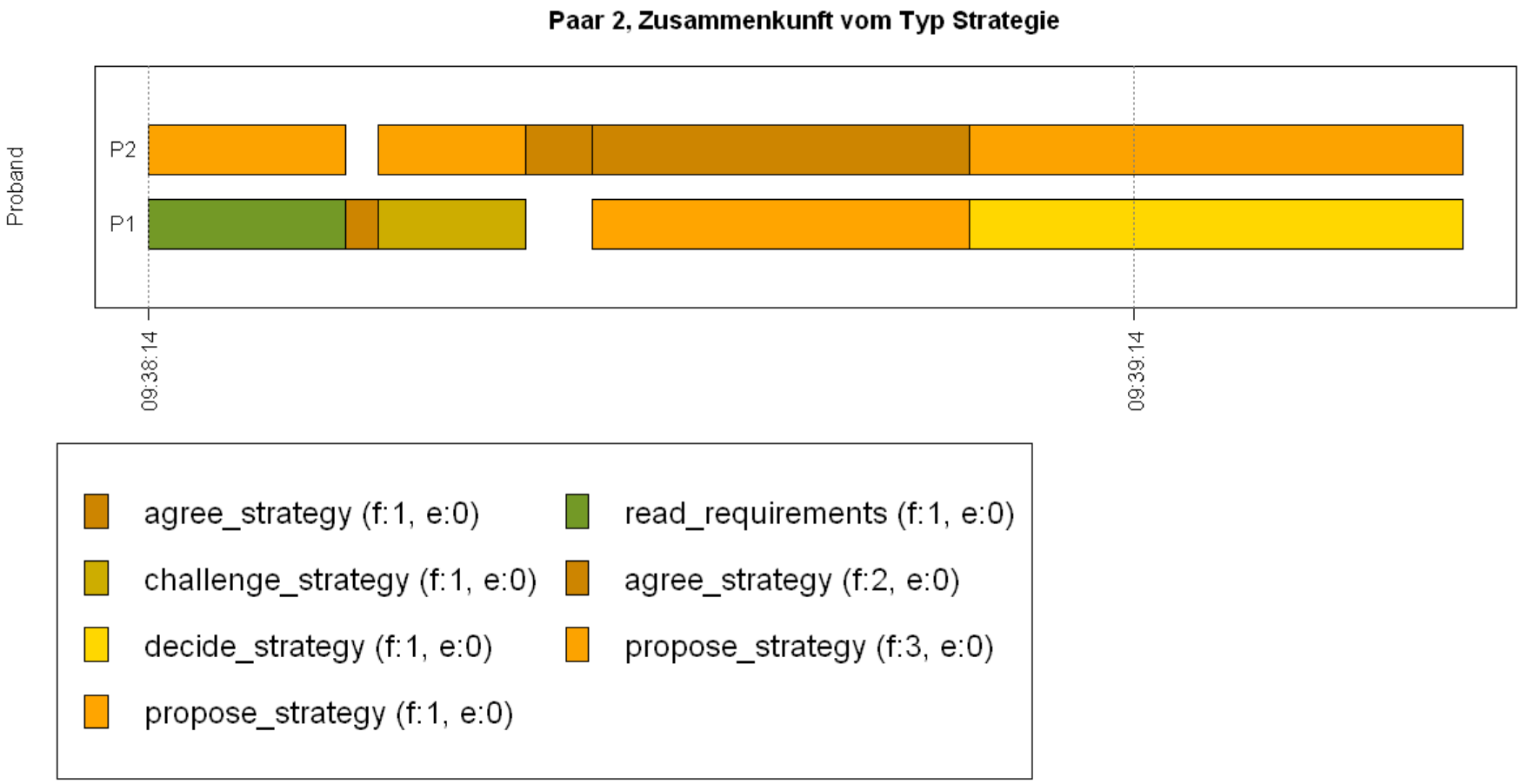
Vorgehen bei der Auswertung

- Zuerst alle Zusammenkünfte kodiert
  - Insgesamt 91
- Dann feingranular alle Phänomene einer Zusammenkunft kodiert
  - Und 30 Sekunden vor der Zusammenkunft
  - Phänomen ist eine bestimmte Aktion, eine Aussage, ein Ereignis, etc.
- Zusammenfassung der Zusammenkunft geschrieben
  - Was passiert, von wem geht es aus, usw.
  - Weiterer Blick auf die Daten, zusätzlich zu den Kodierungen
- Anhand der verwendeten Kodes Typen entwickelt
  - Wo nötig mit Zusammenfassung abgeglichen
  - Mit Paarzusammensetzung verknüpft

Videoscreenshot



Beispielkodierung



Ergebnisse

- 6 Arten der Zusammenkunft und Spezialfall Bemerkung
- physisch und nicht-physisch
- Projektwissen
  - überwiegend nicht-physisch
  - Trat 43 mal über die gesamte Sitzung verteilt auf
  - Austausch von Wissen über das Projekt
    - Mitteilung über den Bearbeitungsstand
      - Dann häufig Integration oder Strategie oder Planung im Anschluss
    - Fragen zur Implementierung des anderen
  - Weniger häufig bei Paaren mit gemeinsamen Hintergrund
  - Beim Paar mit großem Unterschied oft zur Einleitung von Fehlersuche genutzt

Ergebnisse (2)

- Wissen
  - Transfer von Wissen bspw. über den korrekten Einsatz einer Technologie
  - treten am seltensten auf und alle physisch
  - 6 von 7 beim Paar mit großem Unterschied

- Strategie
  - Planung des weiteren Vorgehens (mehr als der nächste Schritt)
  - nicht-physisch
  - treten am Anfang der Programmiersitzung und am Ende von Integration oder Projektwissen auf
  - 9 mal beobachtet

Ergebnisse (3)

- Planung
  - Planung des nächsten konkreten Schritte oder Ausprägung eines Freiheitsgrades
  - Kommen überall während der Sitzung vor
  - Mit 27 Auftreten zweithäufigster Typ
  - Paarzusammensetzung hat keinen Einfluss auf Häufigkeit, Dauer und Themen

- Integration
  - Test des Zusammenspiels der Komponenten
  - Lang und physisch
  - Oft davor: Mitteilung über den Stand, danach: Strategie oder Planung
  - Keine Zusammenkunft beim Paar mit großem Unterschied

Ergebnisse (4)

- Fehlersuche
  - Gemeinsame Behebung eines auftretenden Fehlers
  - Meist physisch
  - Bei den gleichstarken Paaren immer zwischen 2 Integrationen
  - Beim unterschiedlichen Paar isoliert und in nur 2 von 9 Fällen alleiniges Thema der Zusammenkunft
    - Insgesamt nur 13

- Bemerkung
  - Spezialfall
  - Kommentierung einer Aktion des Anderen
  - Von Cockburn als „gut“ bezeichnet
  - Nur beim Paar mit gemeinsamer Vorgeschichte

Probleme

- Völliges Neuland
  - Bisher keinerlei Wissen über Side-by-Side Programming
  - Völlig andere Abschlussarbeit als gewöhnlich
    - Normalerweise Programmierung und dazu ein bisschen Empirie
    - Hier: qualitative, explorative Empirie
- Probleme beim Workshop
  - Unzuverlässigkeit der Probanden
    - Einer erschien nicht zum Experiment
    - Ein Paar praktizierte absichtlich nur Pair Programming
      - Hätte man auf Side-by-Side Programming bestehen sollen?
  - Problem bei der Aufzeichnung
    - Ein Video ging verloren, für ein Paar liegt also nur ein Video vor
  - Somit nur drei verwertbare Videos

Probleme (2)

- Zweites Experiment möglich, aber
  - Genügend Zusammenkünfte in den Videos
  - Zeitlich schwer machbar gewesen
- Beobachtungen während des Workshops nicht möglich
  - Zu sehr in die Durchführung eingebunden
- Technische Probleme
  - Videokonvertierung
  - Sehr große Dateien
  - Synchronität
- Datenflut
  - Jedes Phänomen im Video zu kodieren viel zu umfangreich
  - Entwicklung eines Blickwinkels um die Kodierarbeit zu leiten

Fazit

- Erste qualitative Untersuchung von Side-by-Side Programming
  - Arten der Zusammenkunft, Gründe dafür und Einfluss der Paare darauf ermittelt
- Kein Hinweis darauf, ob SbS-Programming besser/schlechter als PP ist
  - Fokus lag auf Verständnis der Vorgängen nicht auf einem Vergleich der Methoden
  - Allerdings wurde Zusammenarbeit als harmonisch und hilfreich empfunden
- Nur drei Paare
  - Folgestudie könnte Ergebnisse überprüfen
  - Wenn wieder Experiment dann mehrschichtige Anwendung
    - Begünstigt Arbeitsaufteilung

Fazit (2)

- Workshop o.ä. ratsam
  - Gleicher Wissensstand der Probanden
  - Beobachtungen schon während des Workshops
    - Außerhalb einer Experiment-Umgebung
    - Funktionierte hier nicht

- Auf geeignete Aufzeichnungstechnik achten
  - Xvid und MP3 zur Reduzierung der Dateigröße einsetzen
  - Synchronität
    - Nimmt mit der Aufnahmedauer ab
    - Evtl. separate Aufzeichnungsrechner nutzen
    - Lässt sich nur mühsam wieder herstellen

Vielen Dank!