

Qualitative Forschung zu Paar-Programmierung – Teil 2

1. Zusammenfassung
2. Zielstellung
3. Beispiele / Codes / Abstraktion
4. Weitere Überlegungen und Ausblick

Zusammenfassung Vortrag 11.01.2024

Was ist Paar-Programmierung?

"Pair programming (PP) is the practice of two developers working closely together on one computer to solve a technical task." – Franz Zieris [2]

Was ist Methodologien für Entwickler User?

What do you do when you get stuck

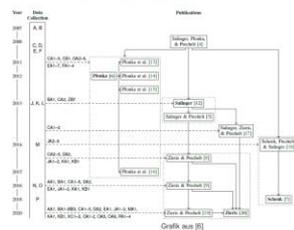


Warum diese Dissonanz???
 → PP als Practice vs. PP als Workmode
 → „Ach, das ist schon Pair Programming!“

Beiträge zum Software Engineering - 11.01.2024

PPind: Datensatz der Arbeitsgruppe [6]

- Sammlung von PP-Sitzungen aus der Praxis seit 2007
- Insgesamt 67 Sitzungen mit ca. 100 Stunden Videomaterial
- Aus 13 Firmen mit 57 Entwicklern
- Diverse Technologien, Zielsetzungen und Paar-Konstellationen
- Klassische (vor-Ort) PP & Distributed (virtuelle) PP



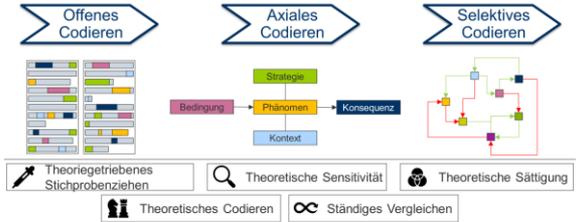
Beiträge zum Software Engineering - 11.01.2024

Qualitative Feldforschung der AGSE

-  Betrachtung / Untersuchung der Softwareentwicklung in der Praxis.
-  Qualitative, explorative Forschung um positive / negative Phänomene zu identifizieren & zu erklären.
-  Formulieren von Theoriegebilden & Ableiten von Empfehlungen für die Softwareentwicklung in der Praxis.
-  Verwendung der Grounded Theory Methodology aus den Sozialwissenschaften zur Bildung fundierter Theorien.

Beiträge zum Software Engineering - 11.01.2024

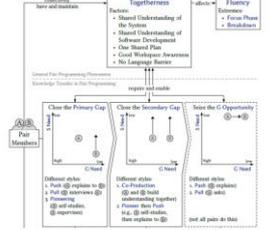
Grounded Theory Methodology*



*Hier sehr vereinfacht und verkürzt dargestellt

Wissenstransfer in der PP (F. Zieris) [2]*

-  Togetherness als fundamentales Konzept in der PP.
-  Hohe Togetherness führt zu Focus Phases, niedrige zu Breakdowns
-  Unterscheidung von S(pecific)- & G(eneral)-Knowledge
-  Wissenstransfer und -erwerb in unterschiedlichen Modi



Gratik aus [2] *364 seitige Dissertation: Ergebnisse in Auszügen und verkürzt

Beiträge zum Software Engineering - 11.01.2024

Missverständnis CA1

Bin mir noch nicht sicher, lass mal überlegen. Im Prinzip kann es sein, dass das leer ist. *Zeigt auf 1. Auswahlfeld "Nicht unter*", kann sein, dass das leer ist. *Zeigt auf 2. Auswahlfeld "Nicht über*", da kann sein das beide hier sind realisiert. Und dann wieder keine Mauszeigerabhängige Beschreibung, obwohl das aktiviert ist. (...) Wäre ich mir nicht sicher, ob es mit einem Button doch sinnvoller wäre. Der Button löscht einfach die Sachen raus.

Aber hier zum Beispiel: "Beschriftung skizzieren" da steht auch was drin, ne? Obwohl es deaktiviert ist.

Nurhäh. (...) Nihilhäh. (...)

Die Frage ist dann halt, inwiefern, wenn ich die Checkbox deaktiviere, dass ich dann, wenn ich es das nächste Mal öffne, Dingier eh leer sind. Dass könnten wir dann ja so machen. (...) Die Auswahlfelder dann leer, wenn man die GUI dann nochmal öffnet und die Funktion nicht aktiviert ist. (...)

Dann macht die Checkbox keinen Sinn, denn für eine Checkbox (...) Na ja ja ich weiß nicht. Nihilhäh. Hast du es dem Markus schon mal gezeigt? Hat er die schon mal gesehen die GUI?

Ne.

Weiß nicht, vielleicht wollen wir erstmal anfragen? Vielleicht sehen wir dann nachher beim Code, wie es am einfachsten gehen könnte.

- Existenz des Missverständnisses nicht klar geworden
- Keine Auflösung und Design-Schwäche wird nicht adressiert
- Unklare Ausdrucksweise C2
- Unterbrochener Denkprozess durch C1
- Ablehnung des Reparaturversuchs

Beiträge zum Software Engineering - 11.01.2024

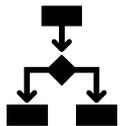
Zielstellung



Wie kann man bessere Entscheidungen in der PP treffen?



Patterns & Anti-Patterns im Entscheidungsprozess identifizieren.



Welche unterschiedlichen Schritte / Strategien gibt es auf dem Weg zu einer Entscheidung?



Charakterisiere Entscheidungen anhand ihres Verlaufs und vergleiche sie miteinander.

WEITERE ERGEBNISSE / ÜBERLEGUNGEN

CA5 – „Wie weiter“

„Wie weiter“ - Entscheidung

Durch Frage initiiert

Wie weitermachen?

Vorschlag direkt angenommen

Strategie

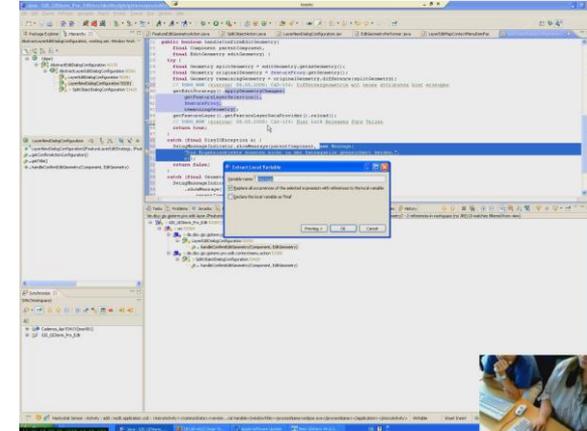
In lokale Variable auslagern - Entscheidung

Durch Vorschlag initiiert

Vorschlag durch Umsetzung

Vorschlag direkt angenommen

Design-Ergänzung



DA4 – Array statt List versuchen

Problemlösung unklar

Wie weitermachen?

Gemeinsames Brainstorming

Einziges Vorschlag

Vorschlag direkt angenommen

Nachgelagerte Erklärung

DA2 – Superklasse benennen

Problemlösung unklar

Wie weitermachen?

Gemeinsames Brainstorming

Vorschlag wird abgelehnt

Übergang zu anderer Entscheidung

OA5 – Methodennamen

Problemlösung unklar

Wie weitermachen?

Gemeinsames Brainstorming

Vorschlag wird abgelehnt

Vorschlag ergänzen

Widerstreitende Vorschläge

Gemeinsames Abwägen

AA1 – Micro zu Mini-Objekt

Durch Vorschlag initiiert

Vorschlag durch Umsetzung

Vorschlag wird abgelehnt

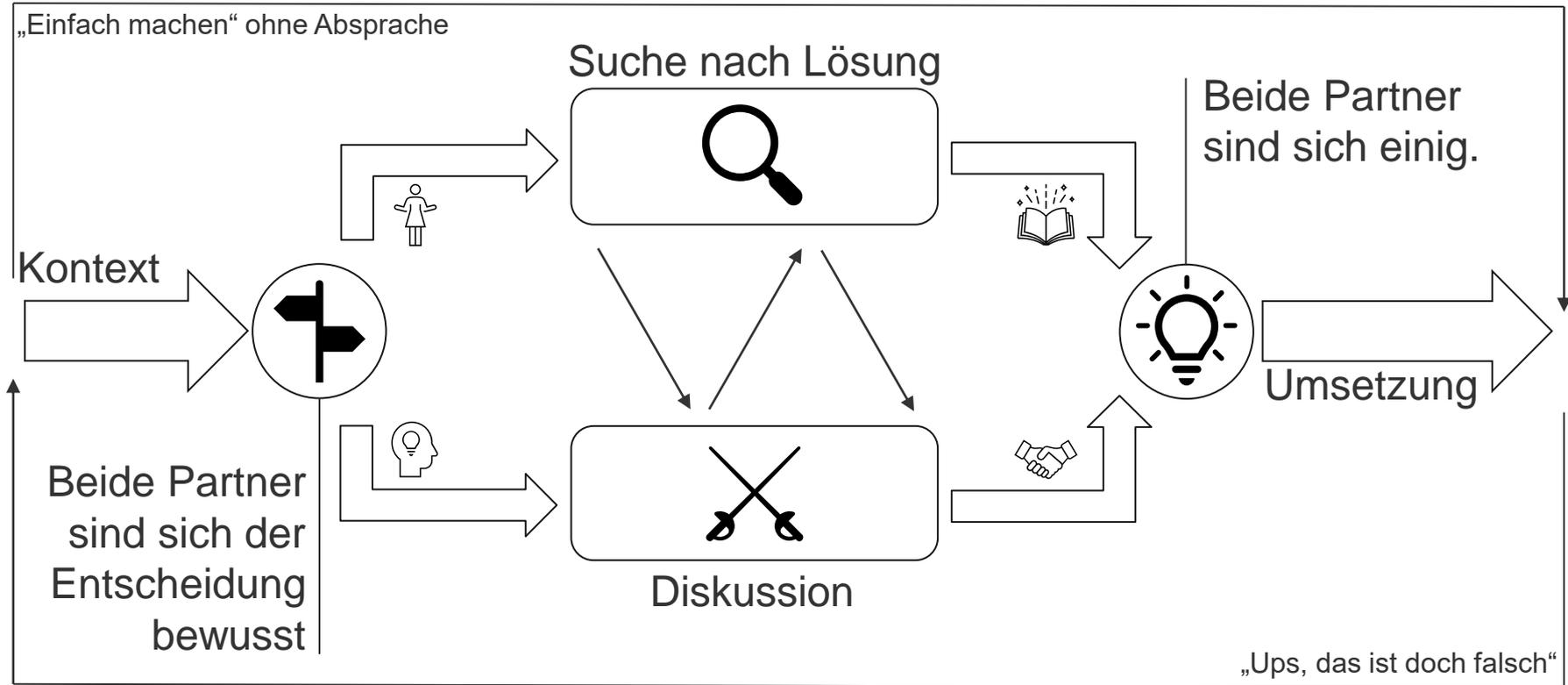
Intervention

Nachgelagerte Erklärung

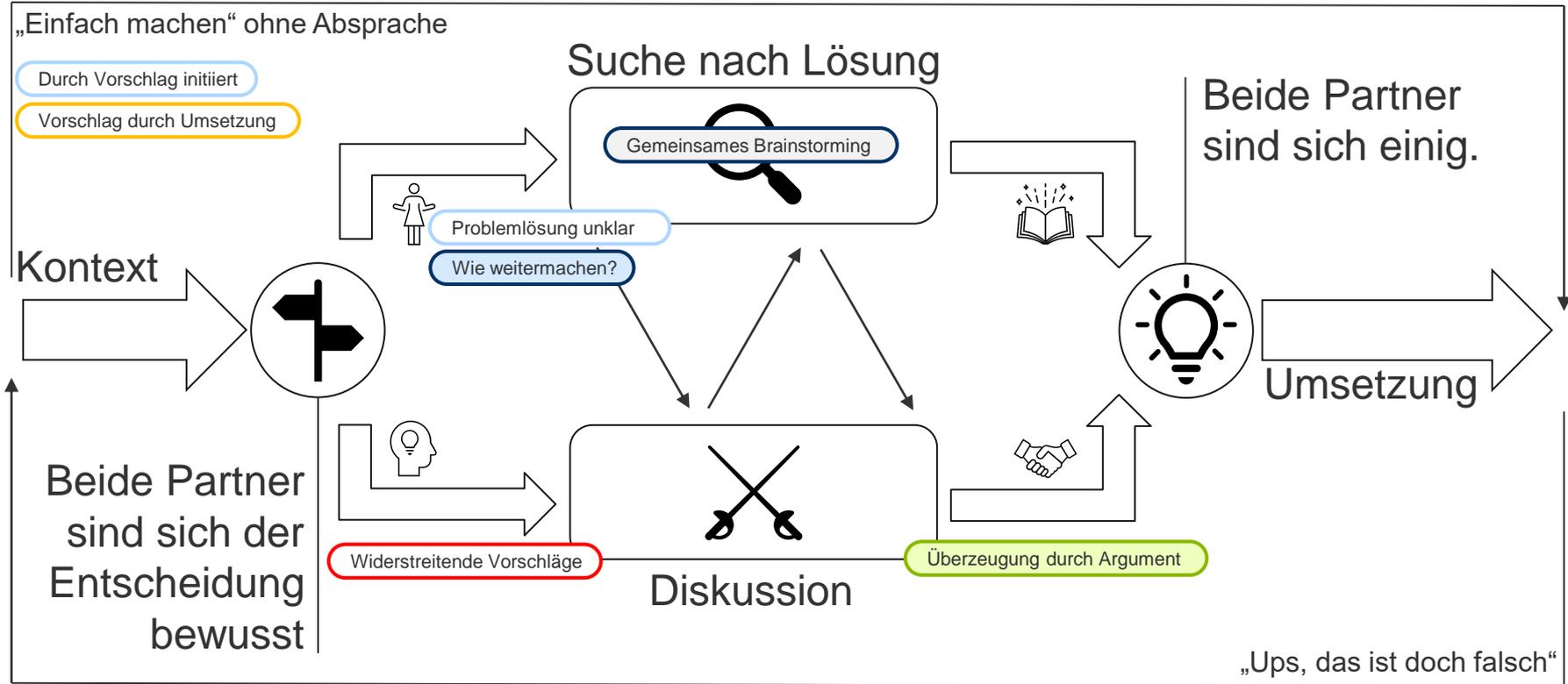
Widerstreitende Vorschläge

Überzeugung durch Argument

Theoretische Betrachtung von Entscheidungen



Theoretische Betrachtung von Entscheidungen



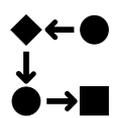
Theoretische Betrachtung von Entscheidungen



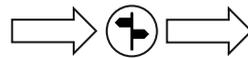
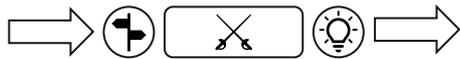
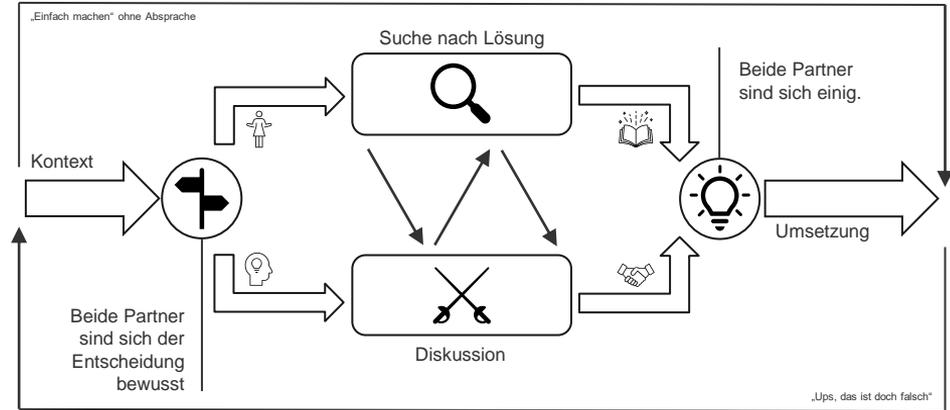
Framework zur Beschreibung von Entscheidungsepisoden.



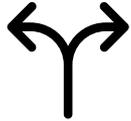
Zwei wohldefinierte Punkte zur Komplexitätsbestimmung.



Charakterisierung von Entscheidungen anhand ihres Verlaufs.



Entscheidungsepisoden?



Eine Paar-Programmierungssitzung ist i.d.R. geprägt durch mehrere kritische Entscheidungspunkte. (Strategie / Design)



Entscheidungen werden durch die Partner in Form eines abgeschlossenen Dialogs (oder Monologs) gefällt.



Entscheidungen sind miteinander verkettet. Manche Entscheidungen können erst getroffen werden, nachdem abhängige entschieden worden sind.



Vergangene Entscheidungen können sich als falsch herausstellen und revidiert werden.

Weitere Themenfelder



Passive Partner (Masterarbeit) & dominante Partner



KI vs. Paarprogrammierung



Wie werden Vorschläge gemacht? „Wir ...“, „Ich ...“, „Man ...“
Mit welcher Stärke wird diskutiert? „Ich glaube / Wir könnten / Wir müssen“

Vielen Dank!

Fragen / Anregungen / Kommentare?