On Knowledge Transfer Skill in Pair Programming

Franz Zieris  
franz.zieris@fu-berlin.de

AG Software Engineering  
http://www.inf.fu-berlin.de/w/SE/WebHome
Pair Programming
Pair Programming Skill

• Good Pair Programming requires **skill** that goes beyond pure programming skills.

• What is this “skill”?  
  – We *don’t know* yet.  
  – We have to **understand** “pair programming” first.
Pair Programming Research

- PP research is usually **quantitative**.
  - Such methods are suited to **quantify known effects** or to **test a hypothesis**.
  - But: which “known effects”?  
  - Also: that doesn’t provide **explanations** for certain phenomena.

→ Gain understanding first, by applying **qualitative methods** available since 2013-12-06
Why consider **Knowledge Transfer**?

- **Typical scenarios for pair programming**
  1. Introducing new employees
  2. Handing over of software modules
  3. Working in critical regions in the source code

- **Knowledge Transfer:**
  1. Expert → Novice
  2. Expert* → Novice* (* concerning the module)
  3. Developer ↔ Developer
### Why consider Knowledge Transfer?

<table>
<thead>
<tr>
<th>Category</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Transfer</td>
<td>ca. 35%</td>
</tr>
<tr>
<td>Decision-Making</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

- **Session gross length:** ca. 2 hours
- **Categories**
  - **Knowledge Transfer:** information exchange, clarification, etc.
  - **Decision-Making:** What to do next? How to do it? ...?
  - **Other:** e.g. direct computer-interaction
- **Knowledge Transfer:** ca. 35% of the time
Knowledge Transfer in PP

• Knowledge Transfer definitely is an important aspect of Pair Programming

• Research Question:

What mechanisms underlie knowledge transfer during pair programming and which of these work well or not so well?
First example

• Let’s meet **Alice** and **Bob**:

  – **Alice** worked on a module that fetches and processes news files (audio snippets).
  – **Bob** sees it for the very first time and wants to understand the system.

• Alice and Bob **work together regularly** and we consider them to be **a very good pair**.

• This is the beginning of their session.
Ich kann dir ja so lange schon mal sagen, was dieses Plugin im großen Ganzen tut.

\[ 1\, \text{Minute} \]

Es wird angefangen zu prüfen, wie die sich in ihrer Größe sich noch verändert. Das heißt, es wird so lange geguckt, bis die Datei nicht mehr größer wird, dann ist sie wohl fertig. Und dann wird sie abgeholt und zur Transkodierung gegeben.

In was für nem Zeitfenster wird dann geguckt?
Ich fange an zu gucken, um zwei Minuten nach der vol- len Stunde, weil da garantiert ist, dass Nachrichtendateien vorliegen **wenn** welche vorliegen.

Und monitore diese Datei dann eben so lange bis sie fertig ist. Das kann bis zu sieben Minuten dauern, je nach Welle.

Ja genau, das ist, äh, Zeitfenster für die Veränderung ist variabel, je nachdem wie die Nachrichten gehen. Das weiß ich ja nicht. Es ist so, dass, die legen immer ne neue Datei an. Wenn die Nachrichten zu Ende sind, wird wieder ne Datei angelegt. Das heißt, ich hab quasi nie mehr als die Nachrichten.
Ja gut, bis maximal fünf vor der neuen Stunde. Also, ich warte wirklich lange.

Ja, ne, ich mein jetzt nur weil du sagst, du guckst halt so lange, äh, bis die Größe aufhört sich zu ändern, ja? Dann musst du ja nen gewisses Zeitfenster noch einplanen, in der immer noch eine Veränderung stattfinden könnte.

Ne ich mein tatsächlich die Größe jetzt, die Größe des Zeitfensters, also (..) du wartest 10 Sekunden, dann nach 10 Sekunden entscheidest du, in den 10 Sekunden hat sich jetzt nichts mehr verändert, dann ist die Datei wohl fertig.

Achso, das meinst du, ne 30 Sekunden.

30 Sekunden, das wollt ich wissen.

Lessons learned

• Knowledge transfer can be quite difficult.  
  – Even for a pair as good as Alice and Bob.
• But eventually they managed to clarify the issue:
  – Bob kept asking until he finally got the information he wanted.
  – Alice answered each of these questions as best as she could.

• We, as researchers, are looking for recurring patterns in such behavior.
Central Concept: Topic
Mode of Knowledge Transfer

Push Mode

Pull Mode

Propellor

Propellor
Episode: Same Topic & Mode

Push Episode →

the example above was a Pull Episode
Let’s revisit the example

- **Topic:**
  Size of polling interval

- **Alice:**
  Supplier

- **Bob:**
  Customer, Propellor

- **Mode:**
  Pull

- Bob kept asking, but *changed the way* he asked.
Let’s revisit the example

• 1st form of asking: Direct Question
  
  In was für nem Zeitfenster wird dann geguckt?  
  Hm genau, aber mh, also das Zeitfenster für die Veränderung?

• 2nd form of asking: Stating Known Facts
  
  Ja, ne, ich mein jetzt nur weil du sagst, du guckst halt so lange, äh, bis die Größe aufhört sich zu ändern, ja? Dann musst du ja nen gewisses Zeitfenster noch einplanen, in der immer noch eine Veränderung stattfinden könnte.

• 3rd form of asking: Simple Step
Let's revisit the example

• 4\textsuperscript{th} form of asking: Proposition

• other Episodes contain a 0\textsuperscript{th} form: Finding

Ne ich mein tatsächlich die Größe jetzt, die Größe des Zeitfensters, also (.) du wartest 10 Sekunden, dann nach 10 Sekunden entscheidest du, in den 10 Sekunden hat sich jetzt nichts mehr verändert, dann ist die Datei wohl fertig.
All five forms of “asking”

0. Finding
   - Locating a general area of interest

1. Direct Question
   - Prototypical form

2. Stating Known Facts
   - Narrowing down the area of the partner’s attention

3. Simple Step
   - Lead the partner’s thinking towards a particular spot.

4. Proposition
   - Reduce possibility to give irrelevant information to zero
The “Clarification Cascade”

0. Finding

1. Direct Question

2. Stating Known Facts

3. Simple Step

4. Proposition

- Locating a general area of interest
- Prototypical form
- Narrowing down the area of the partner’s attention
- Lead the partner’s thinking towards a particular spot.
- Reduce possibility to give irrelevant information to zero
Let’s re-revisit the example

• **Topic:**
  Size of polling interval

• **Problem:**
  The Topic was *unclear.*
What are the difficulties?

• Fundamental problem: Lack of Mental Awareness

• Knowledge transfer requires skillful handling of one’s own and the partner’s mental state.
  – The mental state comprises
    • knowledge (which might be uncertain)
    • an understanding of the partner’s mental state (which might be uncertain)
    • an understanding of these uncertainties

Huh?
Alice's perspective

Alice's uncertain knowledge about Bob's mental state (including his knowledge)
Bob’s perspective

Bob's uncertain knowledge about Alice's uncertain knowledge about his own knowledge

Bob's uncertain knowledge about Alice's mental state

Bob might ask for or describe or ...
Researcher’s perspective
Researcher’s perspective

Context
What are the difficulties?

• Fundamental problem: Lack of **Mental Awareness**
  – What does my partner know?
    • What does s/he expect me to know?

**Ah!**

• Why “fundamental”?  
  – This problem exists all the time, for every utterance.
What are the difficulties?

• More *specific* problems:
  – **Topic & Propellorship**
    • What topic should be addressed next? And by whom?
  – **Topic complexity**
    • Need to recognize complicated/complex Topics and find a way to clarify them anyway.
  – **Focusing**
    • Actually *finish* the Topic.
    • Actually finish the *intended* Topic.
Problem 1: Topic complexity

Positive Example

- Bob needs to explain **seven facts** (1) – (7); Alice understood (1)
- Bob starts with (7) and (6), then **becomes aware** of the complexity: “It’s more complicated than you’d think!”
- Bob **focuses** and eventually explains: (2), (6 again), (5), (3), (4), (5 again)
- Alice understands.

Negative Example

- Two Topics:
  (1) Constraint and its rationale
  (2) Current state of work
- Carl wants to explain both to Dave, and does so **in parallel**
- Dave understands (2), but not (1); it takes Carl 10 minutes to explain both
- Carl was probably **not aware** that Dave was **not aware** of (1).
Problem 2: Focusing

Positive Example

• Before the long example above, Alice started explaining the role of her module.
• Right after the Pull episode (75s in length) Bob gets back on track by summarizing his understanding of the plugin’s role.

Negative Example

• Bob asks Alice for the meaning of an external method’s “return null”.
• They end up discussing coding styles (return “null” vs. Exception).
• In the end, Bob’s semantic question is not answered, instead they agree on using Exceptions in the future.
Problem 3: Propellorship

Positive Example

- Eve and Fynn start their suggestions regarding the next steps simultaneously.
- Eve stops immediately, Fynn proceeds and asks Eve questions.
- Fynn cuts off each of Eve’s responses as soon as he got the information he wanted.
- They proceed with their actual work fluently (and Eve isn’t even mad).

Negative Example

- Dave interrupts Carl’s explanations multiple times by proposing new designs.
- Carl remains polite and reacts on Dave’s proposals, but still tries to round off his explanations.
- Finally Dave starts to pursue his design ideas, while shutting himself off of Carl’s explanations.
- Hardly pair programming
Again: What are the difficulties?

• More *specific* problems:
  – **Topic & Propellorship**
    • What topic should be addressed next? And by whom?
  – **Topic complexity**
    • Need to recognize complicated/complex Topics and find a way to clarify them anyway.
  – **Focusing**
    • Actually *finish* the Topic.
    • Actually finish the *intended* Topic.
Summary

• Rough sketch of problem solving for knowledge transfer challenges:
  – When both developers perceive a knowledge need, they must **not pursue both needs at once**.
  – The Propellor needs to **recognize complicated Topics** that should to be split up.
  – When the **Topic** itself is **difficult to communicate**: lead first oneself and then the partner to a better understanding of the Topic (**Clarification Cascade**).
  – Do **not lose sight of the Topic** until it’s resolved (or there is a good reason to give up).

• Sounds simple?
  – Apparently this is difficult enough to make some pair much more efficient that others → part of “the PP skill”
Research Question?

What mechanisms underlie knowledge transfer during pair programming and which of these work well or not so well?

• **Introduced terminology**
  *Topic, Propellor, Push/Pull Mode, Episode, ...*

• **Identified some patterns/mechanisms**
  *Coping with complex Topics, e.g. by splitting; coping with unclear Topics, e.g. using the Clarification Cascade; struggling for Propellorship*

• **Open:**
  – other mechanisms yet to be seen in different situations
  – relationship of Episodes (sub-topics, split-off topics, merging, ...)
Enough with Push ...
... time for Pull!
Thank you!

https://bitbucket.org/spooning
Used Images


http://en.wikipedia.org/wiki/File:Pair_programming_1.jpg

http://covers.dummies.com/