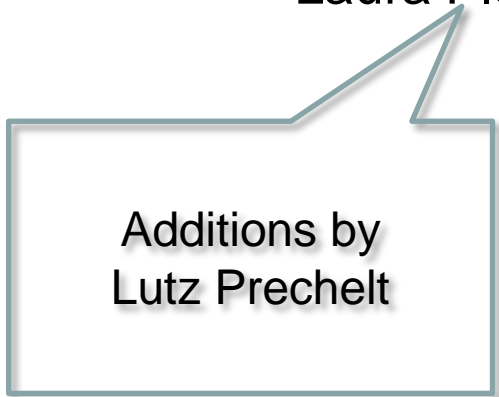


# Disengagement in Pair Programming: Does It Matter?

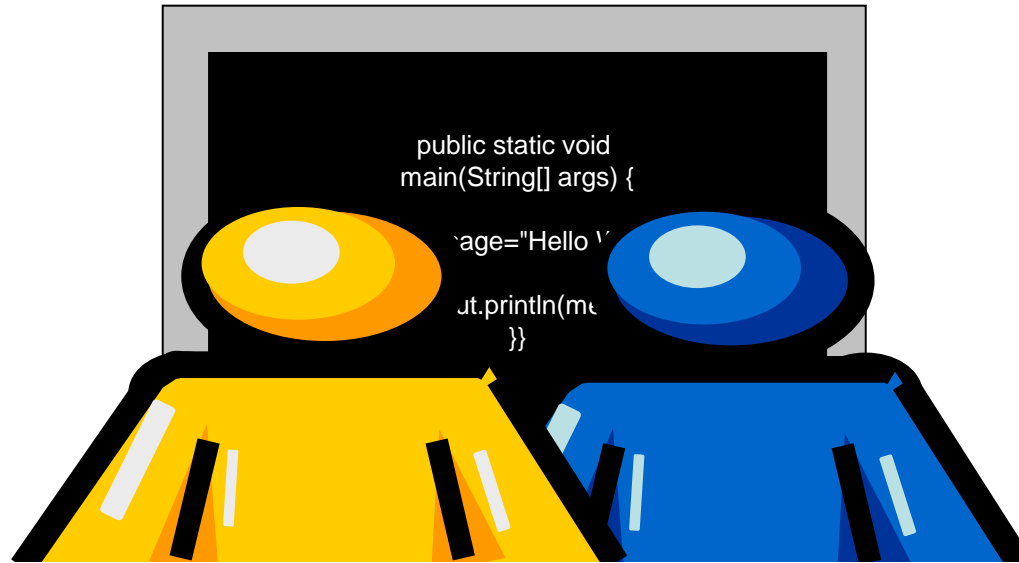
Laura Plonka, Helen Sharp and Janet van der Linden

Department of Computing,  
Open University, UK



Additions by  
Lutz Prechelt

# Pair Programming



- Two developers
- One computer
- One task
- Shared responsibility

# Why Pair Programming?

**Quality**



**Knowledge**



**Communication**



# How does it work?

*“Pair programming works when the pairs are tightly integrated, interacting and working closely.” (Williams and Kessler, 2002)*

**Quality**



**Knowledge**



**Communication**



## Developers' interactions



- Collaboration
- Engagement

**Quality**



**Knowledge**

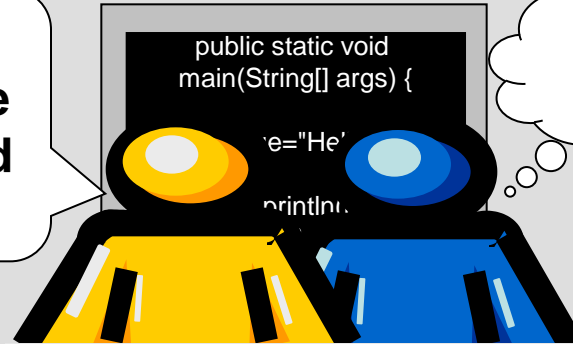


**Communication**



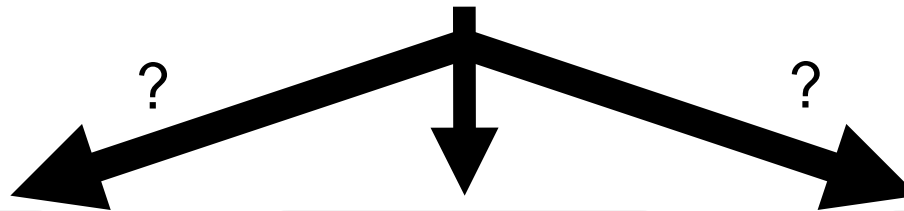
# Disengagement

“Let's test the method ...”



Lalala  
lalala

Not focus on the task,  
allow the partner to work for herself



## Quality



## Knowledge

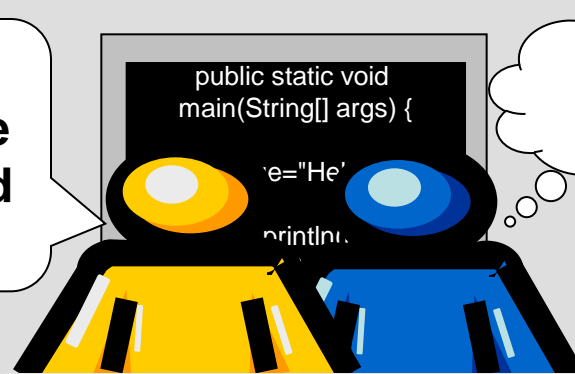


## Communication



# Disengagement

“Let's  
test the  
method  
...”



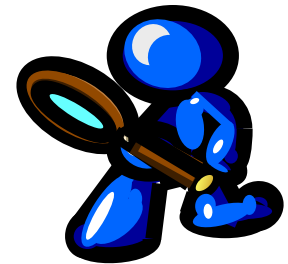
Lalala  
lalala

# Research focus

1. When do developers disengage in PP sessions?
2. How can disengagement be avoided?
3. Does disengagement matter?

# Research approach

In depth qualitative analysis of developers' interactions





# Data Collection

**Questionnaires**



**Audio and video  
recordings of PP sessions**



# Video of PP sessions

The image displays a screenshot of an IDE (likely Eclipse) with a Java project. The Package Explorer on the left shows a hierarchy of files, including `UndoManagement.java`, `VetoableVariableProxy.java`, `ProjectProxy.java`, and `VariableProxy.java`. The main editor window shows the code for `UndoManagement.java`, which includes methods like `addAndNotify`, `add`, `remove`, `setVariable`, and `setVariable` with an `exclude` parameter. The code is as follows:

```
25 public void addAndNotify(VariableProxyListener<T> listener) {
26     nonVetoables.add(listener);
27     listener.setVariable(variable);
28 }
29
30 public void add(VariableProxyListener<T> listener) {
31     nonVetoables.add(listener);
32 }
33
34 public void remove(VariableProxyListener<T> listener) {
35     nonVetoables.remove(listener);
36 }
37
38 public boolean setVariable(T variable) {
39     return setVariable(variable, null);
40 }
41
42 public boolean setVariable(T variable, VariableProxyListener<T> exclude) {
43     this.variable = variable;
44     for (VariableProxyListener<T> vpl : nonVetoables) {
45         if (vpl == exclude) {
46             continue;
47         }
48         vpl.setVariable(variable);
49     }
50     return true;
51 }
52
53 public T getVariable() {
54     return variable;
55 }
```

The Problems window at the bottom shows "No consoles to display at this time." An inset photograph in the bottom right corner shows two men, one in a blue shirt and one in a red shirt, looking at a computer screen together. The man in the blue shirt is pointing at the screen, and the man in the red shirt is holding a smartphone.

# Data Collection

**Questionnaires**



**Audio and video recordings of PP sessions**



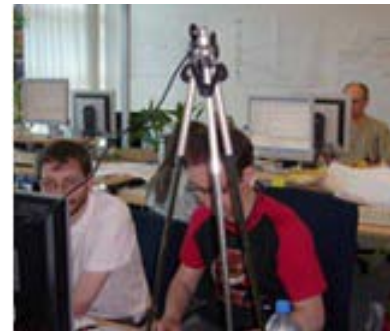
**Interviews with pairs of developers**



# Data Background

- Industrial settings
- Day to day task
- 21 recorded pair programming sessions (each session between 1 and 3 hours)
- 31 developers
- 4 companies

various application domains



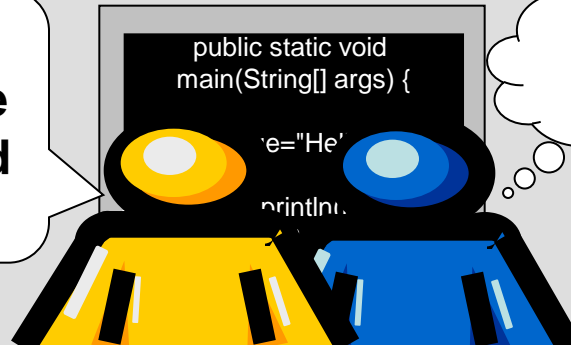
# Analysis Steps

1. Identify episodes of disengagement in PP sessions
2. Investigate circumstances leading to disengagement
3. Explore how disengagement in PP sessions can be avoided

- Feedback
- Mirroring
- Question, response
- Contribution
- Intervention

## Disengagement

“Let's test the method ...”



Lalala  
lalala

**Indicators for engagement**

**Absence of engagement**

# Analysis Step 1

Identify episodes of disengagement in PP sessions

**Audio and video recordings of PP sessions**



**Interviews with pairs of developers**



# Analysis Step 2

Investigate circumstances leading to disengagement

**Audio and video recordings of PP sessions**



**Interviews with pairs of developers**





# Analysis Step 3

Explore how disengagement in PP sessions can be avoided

**Audio and video recordings of PP sessions**



compare similar sessions with and without disengagement

# When do developers disengage?



Interruptions



Division of work  
according to expertise



Simple tasks



Social pressure



Time pressure



**Pete** replies to the co-worker: *“I have no idea what you are talking about.”*

**Co-worker** explains what he means and asks: *“[...]When do you think it will be ready?”*

**Pete** is pointing to the screen: *“I’ll take care of it when we’re done here.”*

**Co-worker:** *“Ok.”* Then he leaves.

**Sandra** is still typing: *“We do it like this, alright?”*

No reaction from Pete, Sandra continues typing in silence. After 40 seconds

**Sandra** stops driving and asks: *“Can you continue?”*

**Pete** is taking the mouse and keyboard without starting to type: *“I’m completely lost at the moment. Why does he need to bother me with this right now?”*



Division of work according to expertise

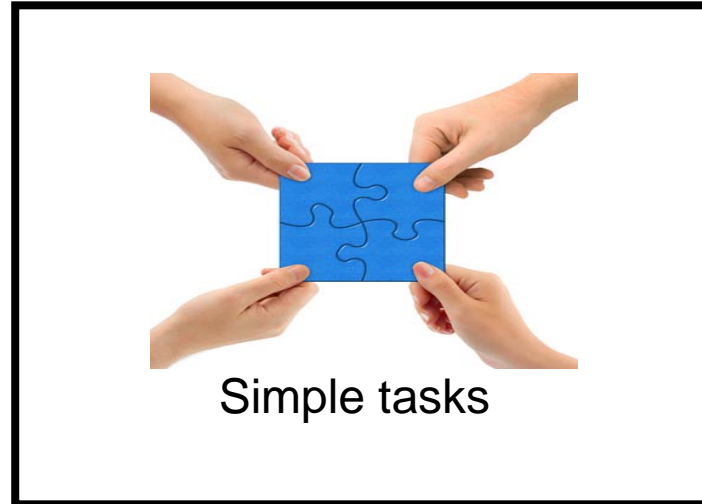
**David:** *“And that is called in Java. So let’s debug it from the Java side because everything else does not make sense.”*

**Alex:** *“But I do not know the Java part.”*

**David is taking the keyboard:** *“That does not matter. So I take the lead now.”*

David starts driving and an episode of disengagement of Alex starts. Later in the session, they face a C++ task while David is still driving. He shifts the keyboard to Alex and says: *“Ok, I think I let you do that! Ok?”*

from the interview:



*“[...] the part where the refactoring started. You could have done that without me. I was just sitting next to you in this situation. There was nothing to think about. [...] We did not build anything new [in this situation]. That was just about executing the task.”*

# When do developers disengage?



Interruptions



Division of work  
according to expertise



Simple tasks

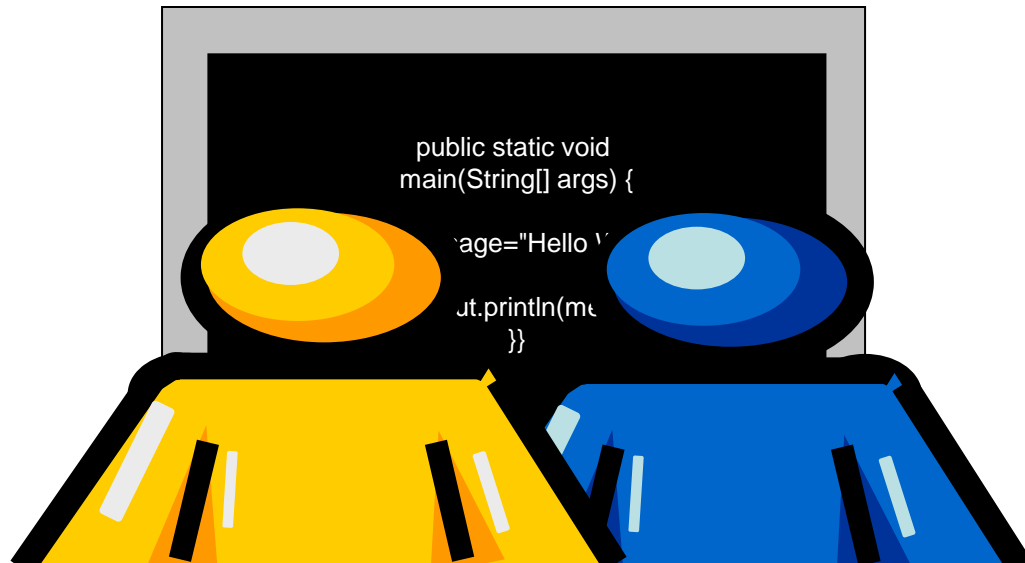


Social pressure

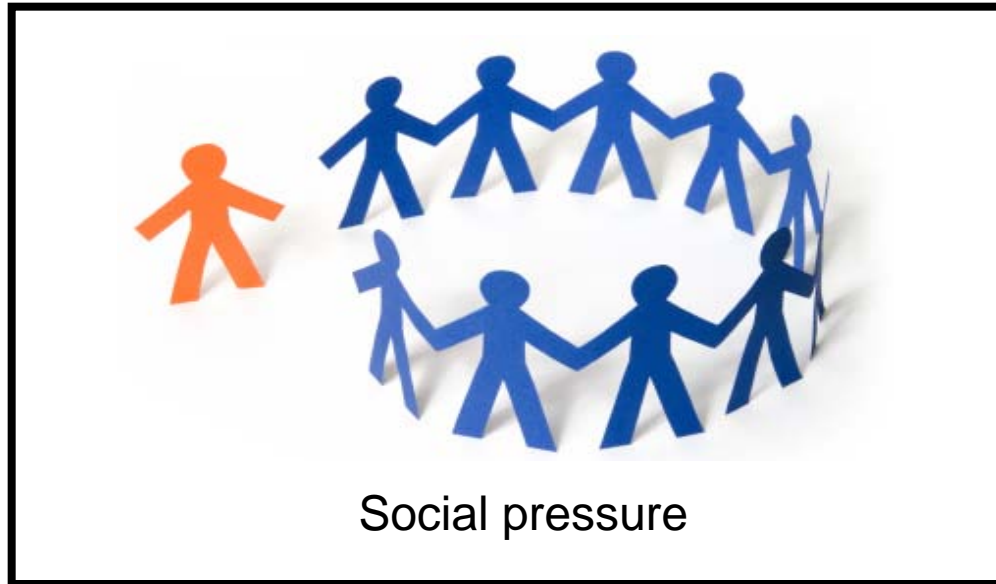


Time pressure

# Expert-Novice Constellations



- **Pair Programming experience**
- **Knowledge transfer**



**Tom (novice):** *“I feel like I have to understand it [explanations by his partner] as quickly as possible. So, that we can move on to solving the task and that we get started and”.*

**Max (expert) interrupts:** *“In order to not look stupid?”*

**Tom (novice):** *“Yes.”*



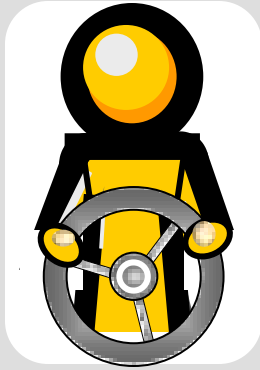


Time pressure

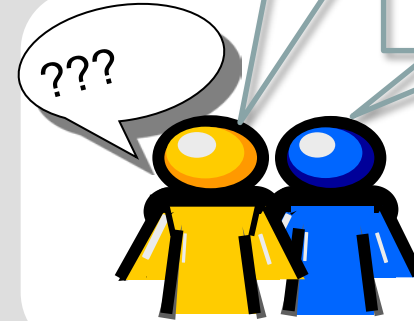
### **Expert statement**

*“If we would have had more time and not that much deadline pressure I would have explained the topic to him properly. [...] but we haven’t had the time for that. “*

# How can disengagement be avoided?



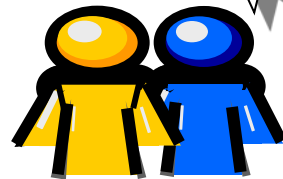
Encouraging novice to drive



Asking for clarification

Novice

Expert



Do you agree?

Verbalisation, explanation and feedback questions

# Does disengagement matter?

**Acceptable  
disengagement**



**Harmful  
disengagement  
(conflicting goals)**



**Visible vs invisible disengagement**



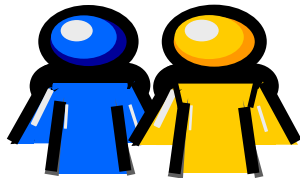
## Awareness

State of engagement

Motivation for session

Minimizing the  
risk of harmful  
disengagement

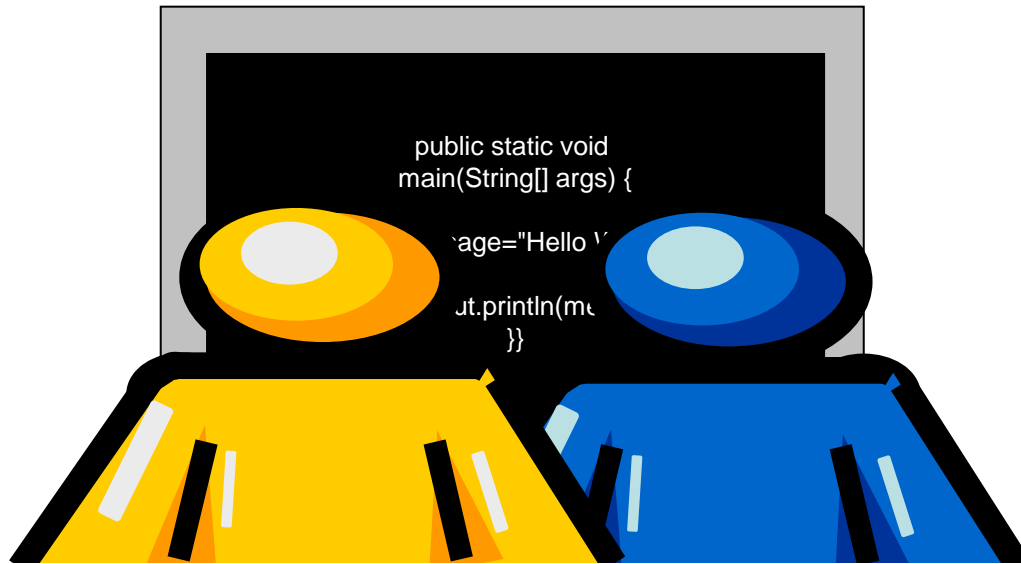
Developers' interactions



Session planning



# Thank you!



## Any questions?