

Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

Video Analysis of Pair Programming

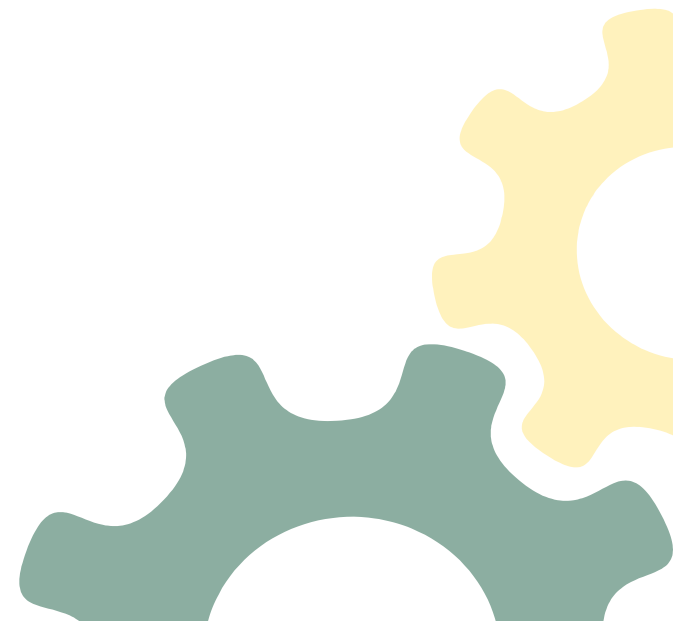
Andreas Höfer · APSO08

10 May 2008



Fakultät für **Informatik**

Lehrstuhl für Programmiersysteme





We have a lot...

- descriptions of pair programming in books for XP practitioners
 - Beck 2000
 - Jeffries et al. 2001
 - Wake 2002
 - Williams & Kessler 2002, ...
- studies about the efficiency of pair programming compared to other methods (though the results are mixed).
 - Dybå et al. '07



But we only have a few...

- studies examining pair compatibility based on personality traits
 - Katira 2004
 - Domino et al. 2003
 - Chao & Atli 2006
 - Sfetsos 2006
- studies observing and analyzing pair interaction
 - Bryant 2004
 - Bryant et al. 2006
 - Chong & Hurlbutt 2007



Role Attributes

Driver

- programming
- using mouse and keyboard
- writing down design
- explaining actions to the navigator

Navigator

- thinking strategically
- thinking of alternative solution/design
- reviewing driver
- listening to the driver's explanations

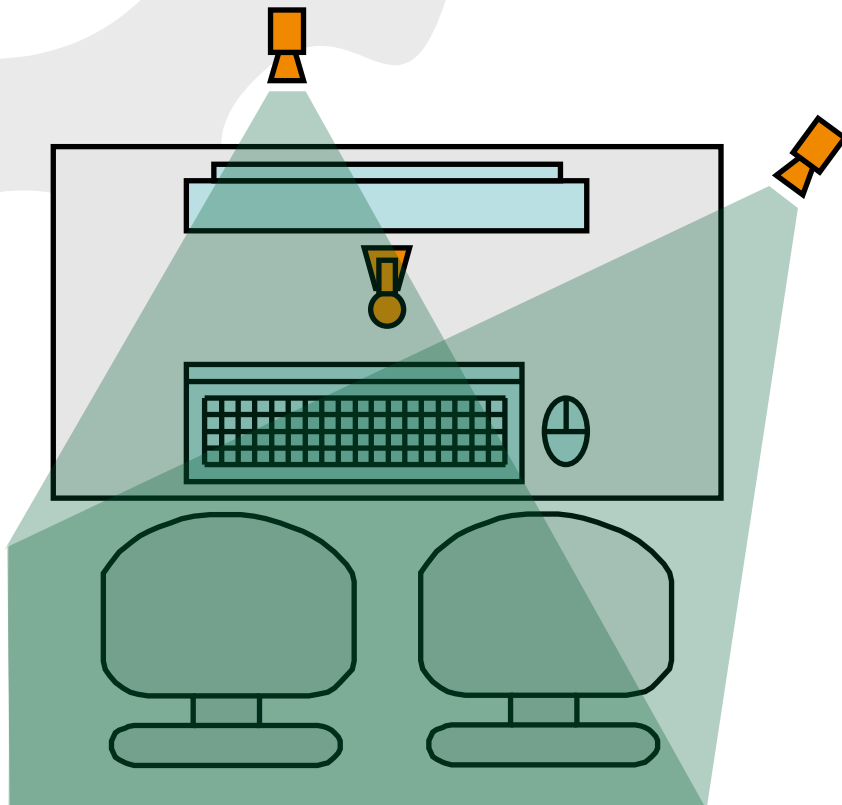


Participants

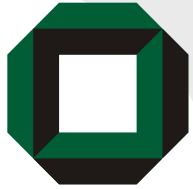
- 18 Computer Science Students
- enrolled in an XP-Course
- 3.4th year of study
- 4.8 years programming experience, including 2.0 years experience with Java



Workplace and Task



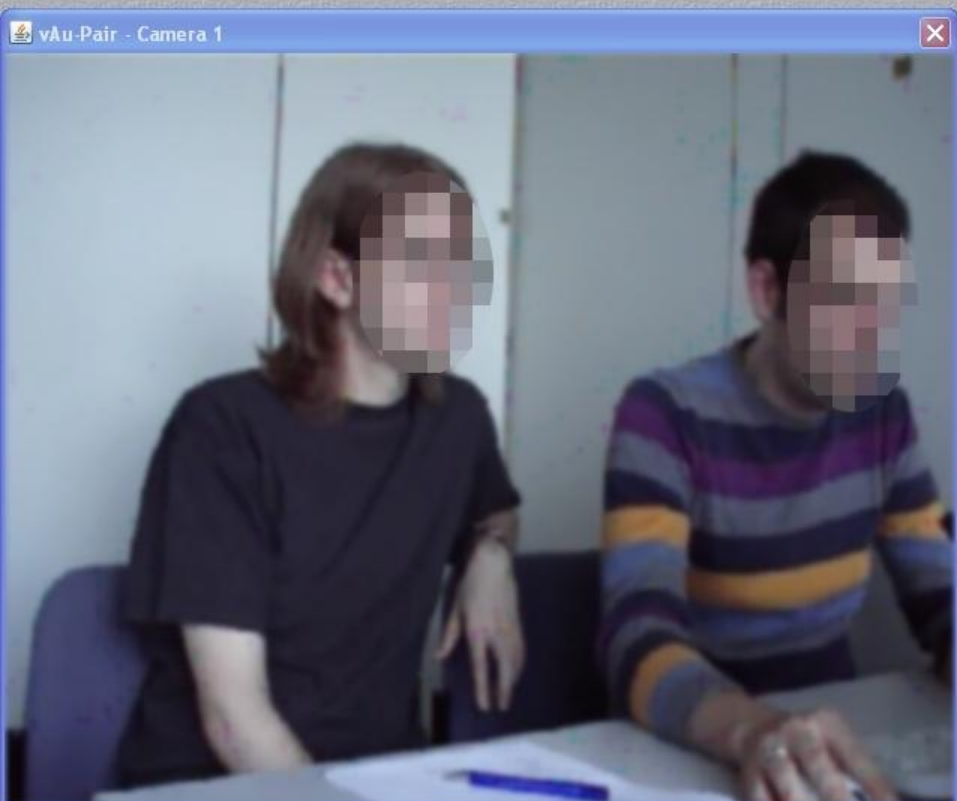
- Implement “open” state of an elevator control
 - Java
 - JUnit 3.8.1
 - mock-object



Obtaining the results

Transcribe
videos





vAu-Pair - Screen capture

IDE interface showing Java code for an Elevator simulation.

```

class Elevator {
    ...
    public void testOpenBottom() throws Exception {
        initElevator(0, 0, Elevator.RUNNING_DOWN, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        Elevator.doWait();
        assertEquals(Elevator.RUNNING_DOWN, elevator.getState());
        assertEquals(0, elevator.getFlooringTime());
        checkIsEmptyInVerify();
    }

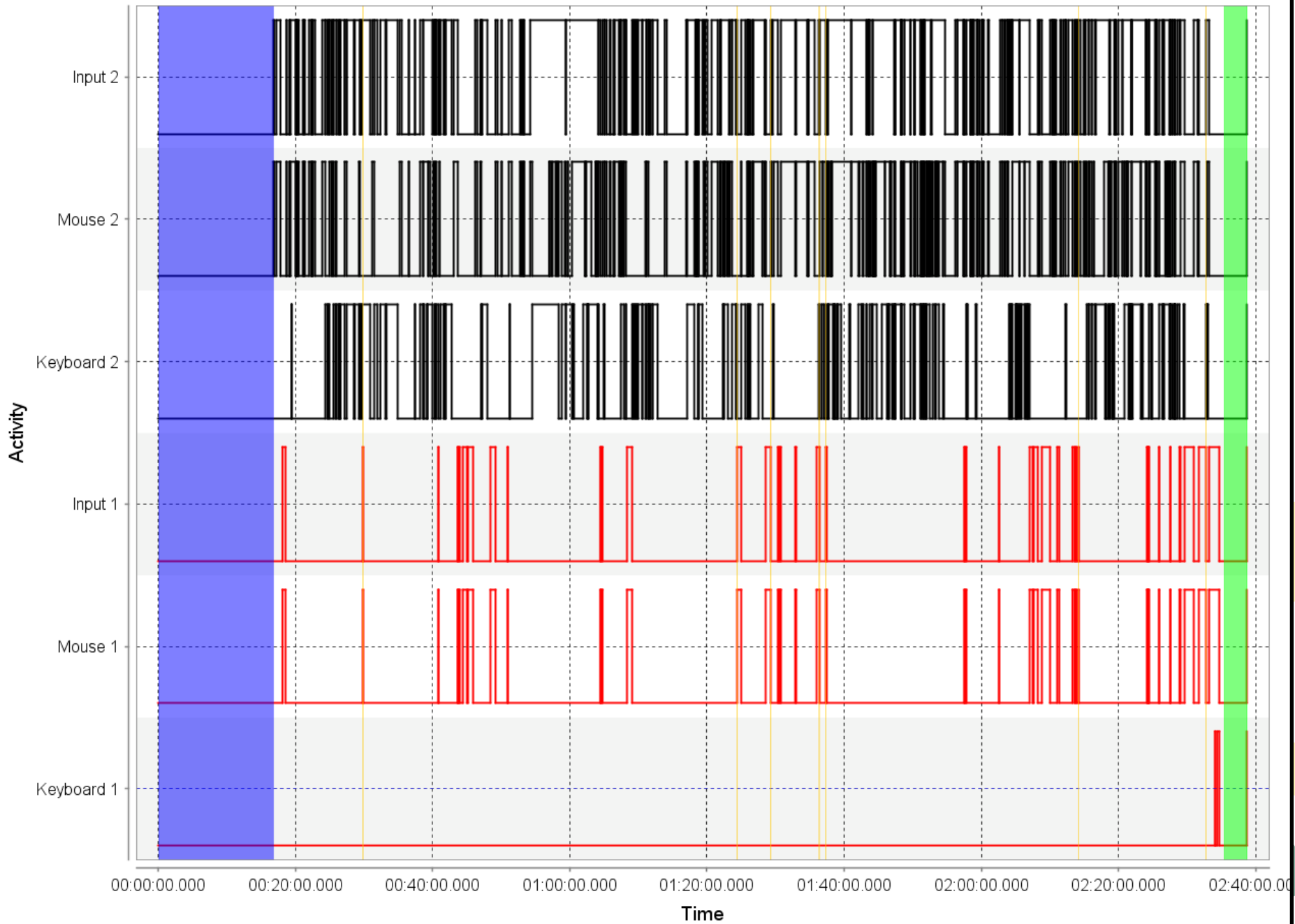
    public void testOpenBottom() throws Exception {
        initElevator(0, 0, Elevator.RUNNING_DOWN, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        Elevator.doWait();
        assertEquals(Elevator.RUNNING, elevator.getState());
    }

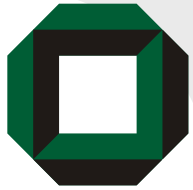
    public void testOpenBottom() throws Exception {
        initElevator(0, 0, Elevator.RUNNING_DOWN, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        checkIsEmptyInInitFlooringResponse(0, 1, 0);
        Elevator.doWait();
        assertEquals(0, elevator.getState());
        assertEquals(0, elevator.getState());
        assertEquals(0, elevator.getState());
        assertEquals(0, elevator.getState());
        assertEquals(0, elevator.getState());
    }
}
    
```

vAu-Pair - Pair events

Timestamp	Level	Category	Type	Participant	Comment
02:18:43.265	2	Programmer uses mouse.	BEGIN	TWO	
02:18:46.193	4	Programmer uses keyboard.	BEGIN	TWO	
02:18:59.860	2	Programmer uses mouse.	END	TWO	
02:19:00.919	2	Programmer uses mouse.	BEGIN	TWO	
02:19:06.213	4	Programmer uses keyboard.	END	TWO	
02:19:28.112	2	Programmer uses mouse.	BEGIN	TWO	
02:19:28.722	2	Programmer uses mouse.	END	TWO	
02:19:36.066	2	Programmer uses mouse.	BEGIN	TWO	
02:19:43.480	2	Programmer uses mouse.	END	TWO	
02:19:54.576	2	Programmer uses mouse.	BEGIN	TWO	
02:20:32.916	2	Programmer uses mouse.	END	TWO	
02:20:38.665	2	Programmer uses mouse.	BEGIN	TWO	
02:20:52.427	4	Programmer uses keyboard.	BEGIN	TWO	
02:20:53.662	2	Programmer uses mouse.	END	TWO	

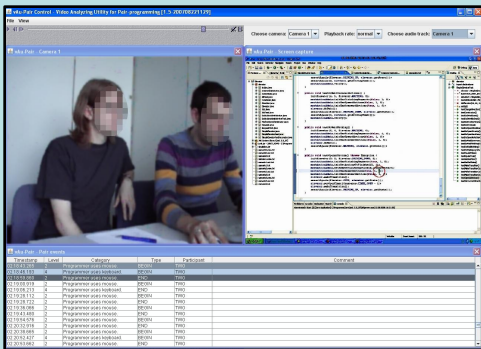
Input Activity per Programmer



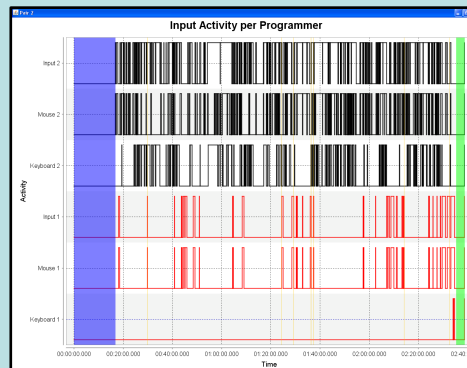


Obtaining the results

Transcribe
videos



Preprocess
transcripts

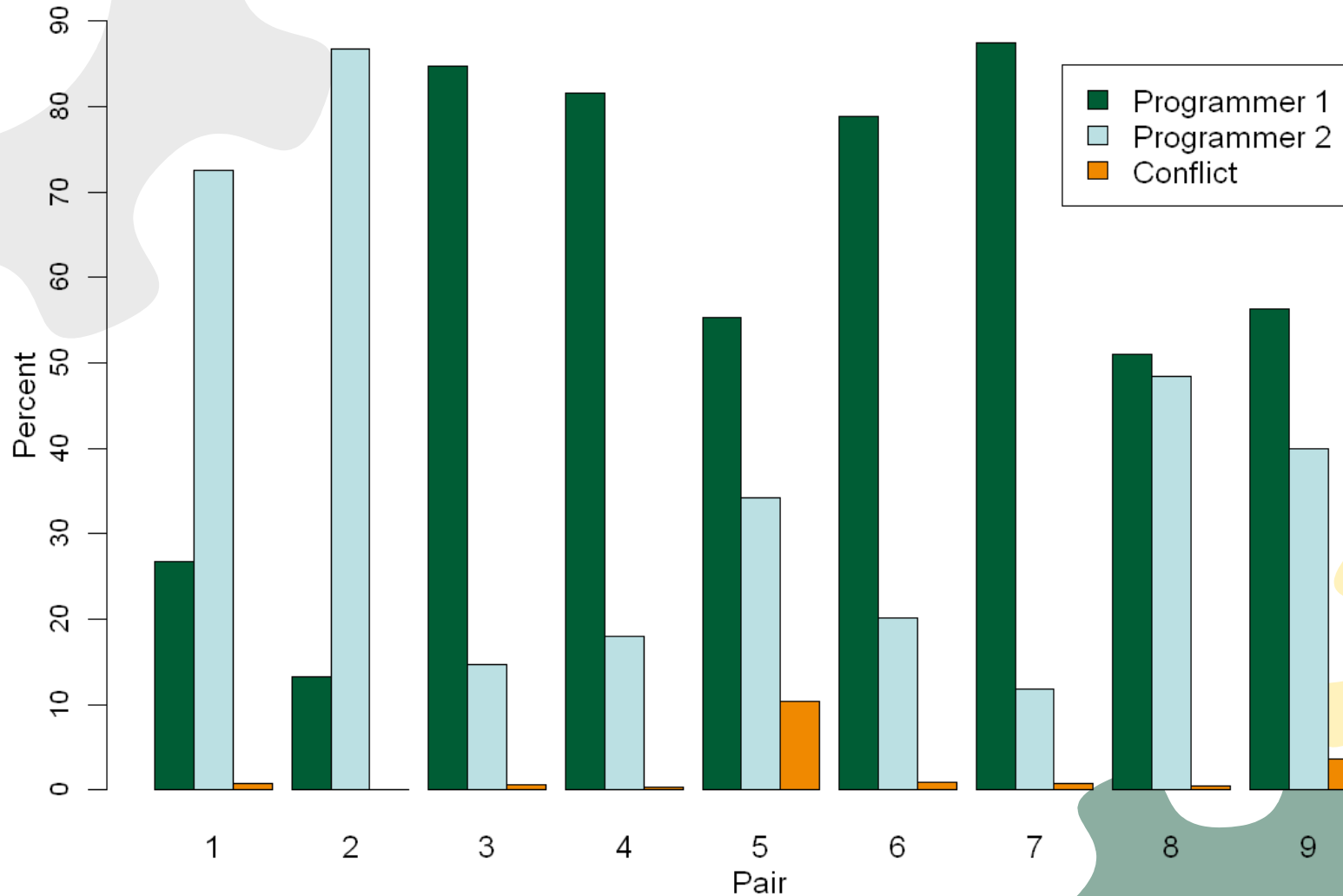


Analyze
transcripts





Input Device Control

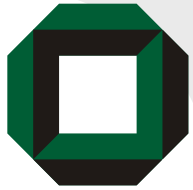




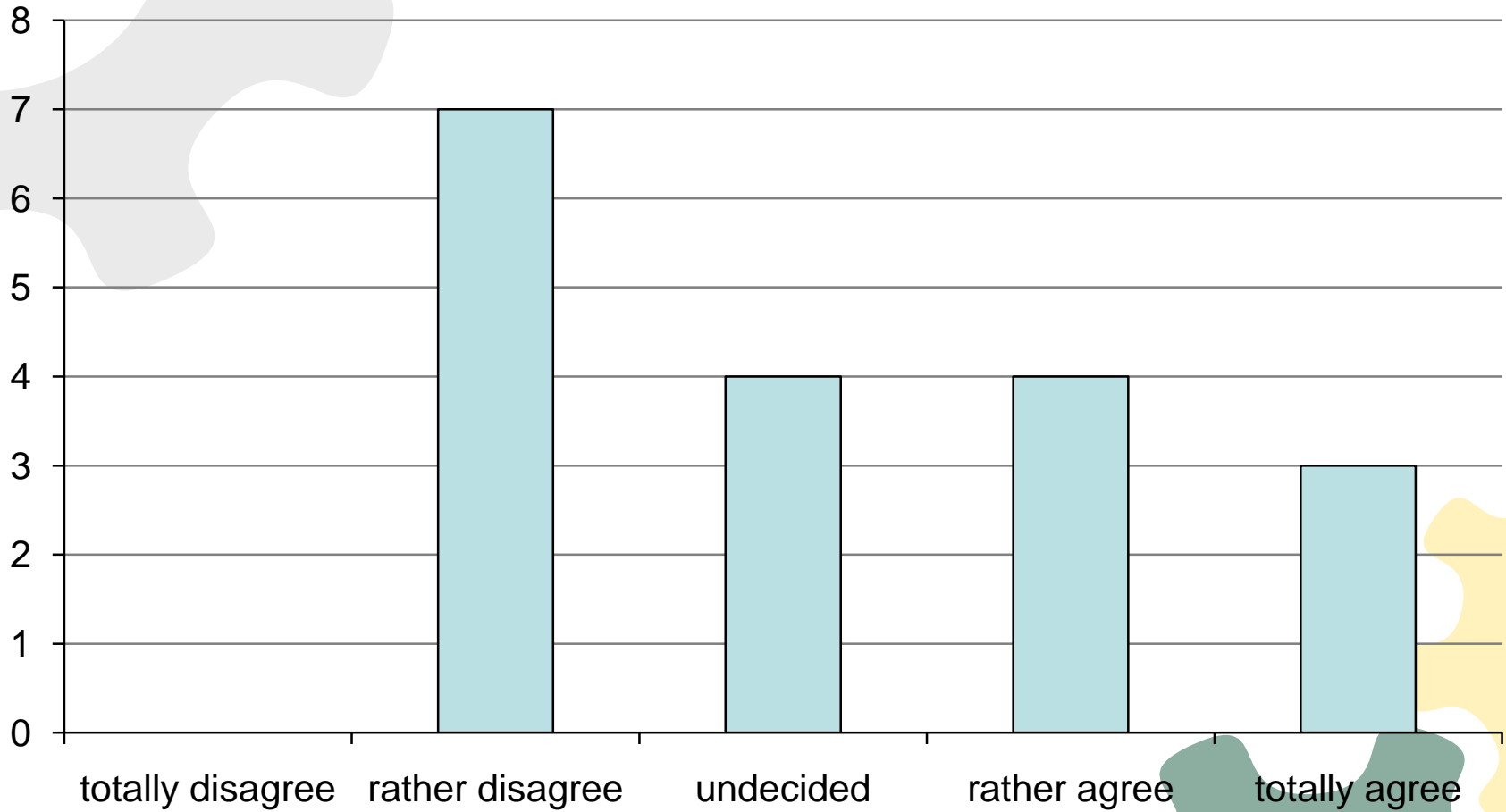
Time for task completion pair balance and input device

$$b = \frac{\min(t_1, t_2) + \frac{1}{2}t_c}{\max(t_1, t_2) + \frac{1}{2}t_c}$$

Pair	Time (h:mm:ss)	Balance	Conflict (%)
1	3:07:40	0.37	0.76
2	2:17:55	0.15	0.10
3	2:34:45	0.18	0.57
4	6:52:07	0.22	0.33
5	4:03:01	0.65	10.42
6	4:28:57	0.26	0.94
7	4:20:05	0.14	0.75
8	2:42:13	0.95	0.54
9	2:26:46	0.72	3.60
Mean	3:39:17	0.40	2.00

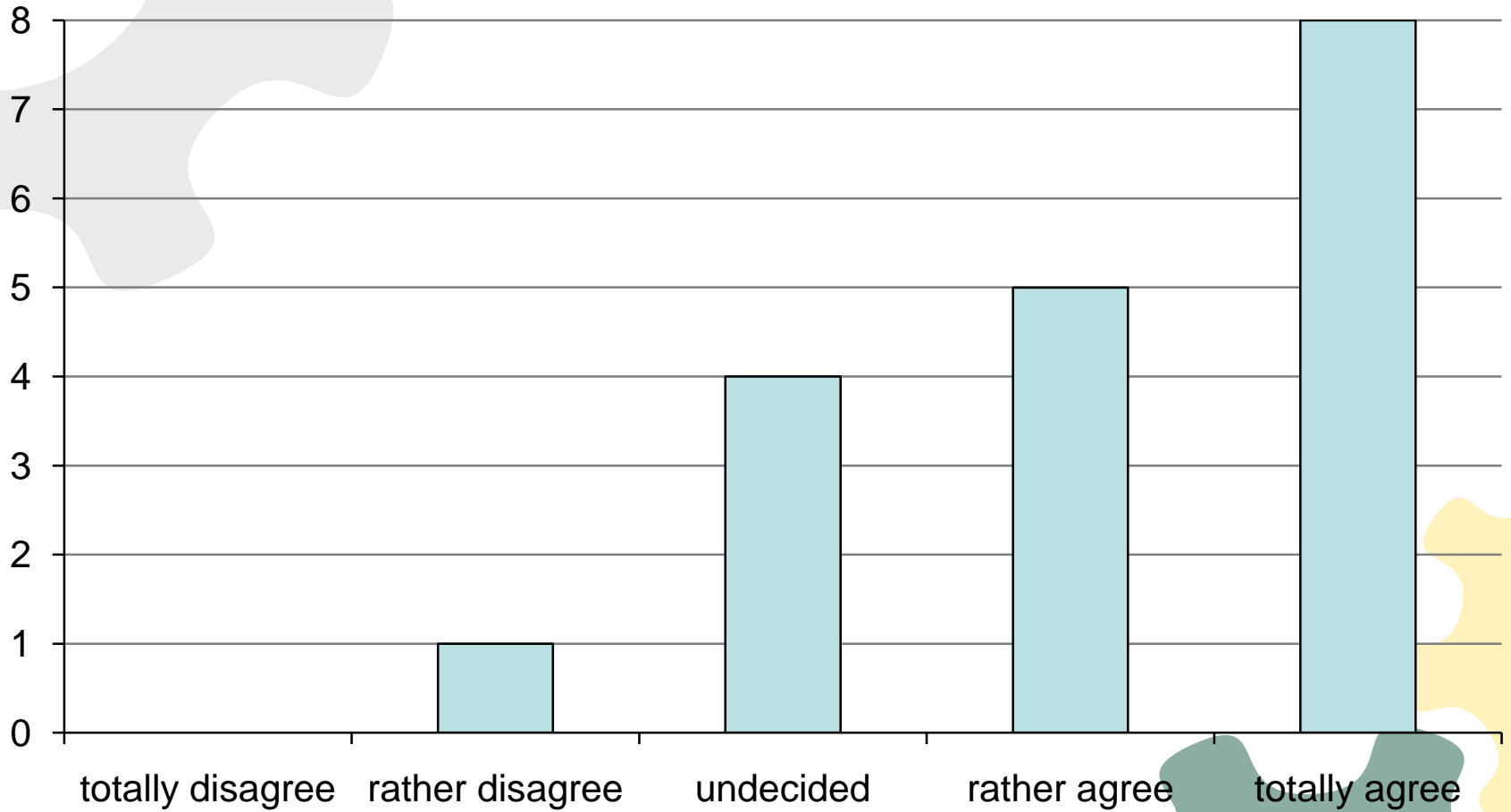


Our activity on the keyboard was balanced.





I would work with my partner again.





Driving Times

Pair	Min.	Max.	Med.	Mean	Std. Dev.
1	0:00:02	0:13:02	0:00:43	0:01:40	0:02:28
2	0:00:01	0:22:12	0:00:46	0:02:23	0:04:36
3	0:00:18	1:42:57	0:03:03	0:11:54	0:27:45
4	0:00:01	0:45:31	0:01:17	0:04:02	0:06:56
5	0:00:03	0:22:55	0:01:19	0:02:19	0:03:12
6	0:00:02	0:33:33	0:01:07	0:02:37	0:04:27
7	0:00:04	0:31:28	0:01:00	0:03:46	0:06:29
8	0:00:07	0:25:33	0:01:17	0:03:07	0:04:49
9	0:00:04	0:15:16	0:01:03	0:02:02	0:02:34

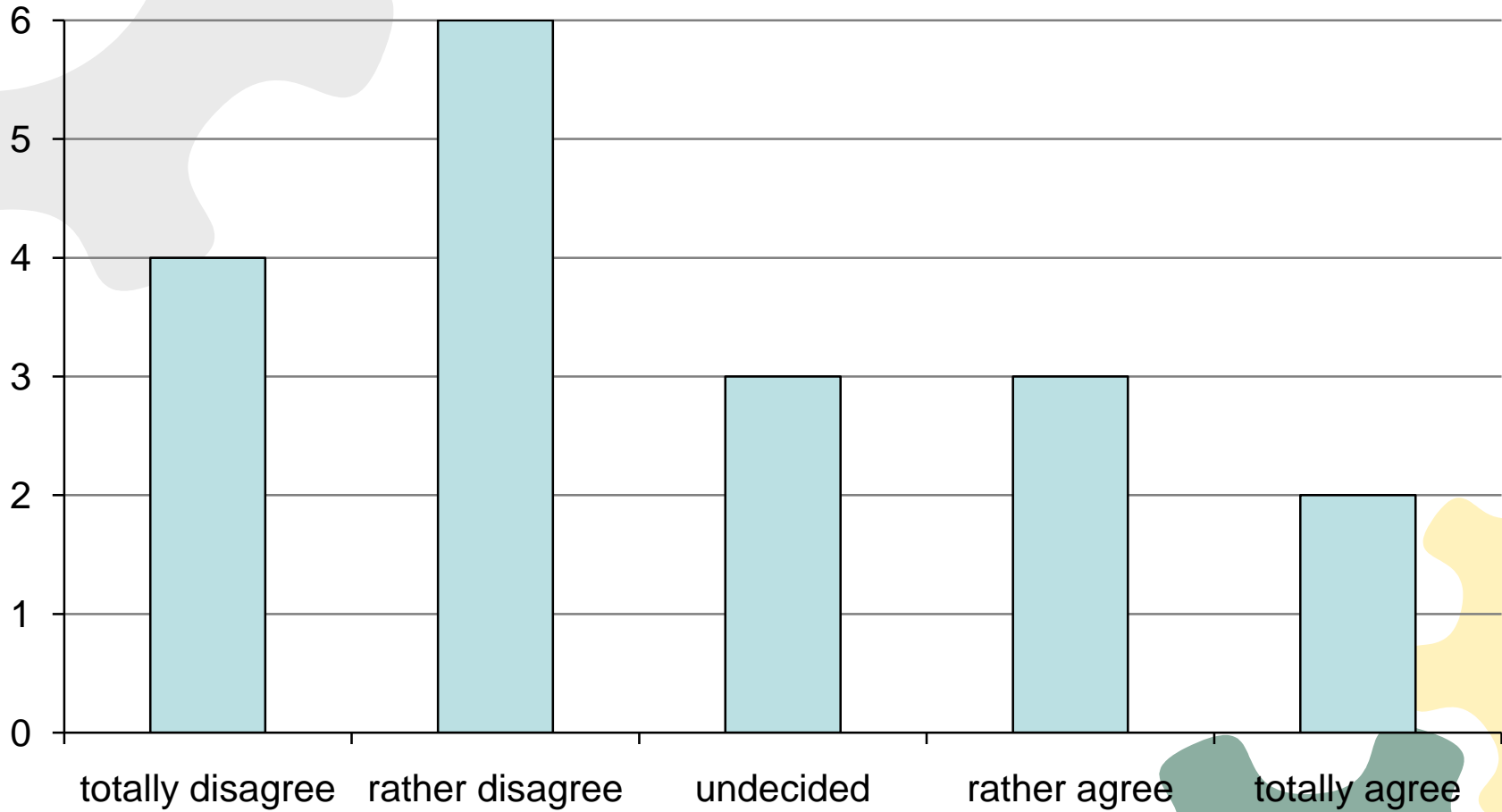


Limitations

- Few data points
- Participants rather inexperienced in XP
- Experimental setting and cameras



I felt disturbed and observed due to the cameras.





Next Steps

- Collect more data points (especially from experts)
- Analyze verbalizations
- Analyze conformance to TDD
- Further suggestions are welcome!



Discussion

Thank you
for your attention!