The Agile Modules
CSC3045 & CSC3052

Pair Programming

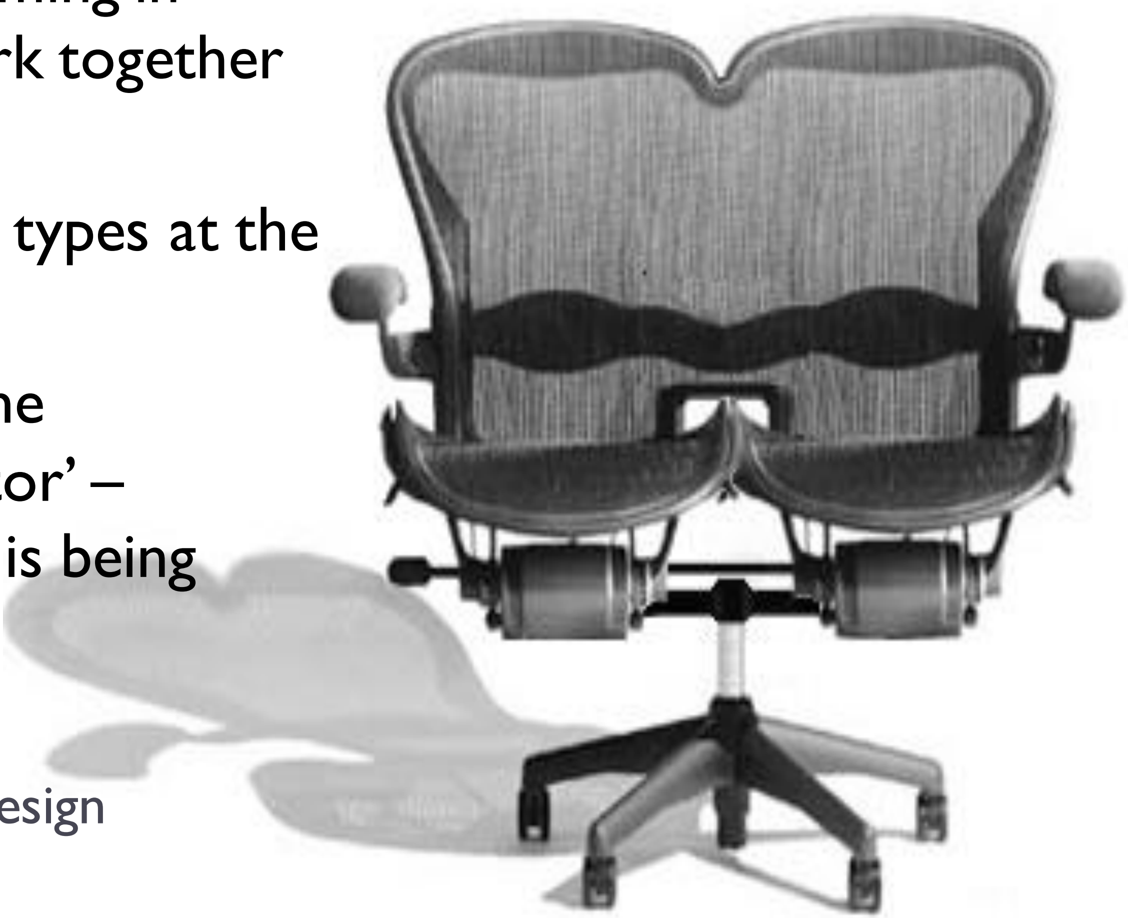
School of Electronics, Electrical Engineering & Computer Science

Queen's University, Belfast

Dr Darryl Stewart

What is Pair Programming?

- A method of programming in which two people work together at one keyboard
- One person ‘Drives’ – types at the keyboard
- The other person is the ‘Observer’ or ‘Navigator’ – reviews each line as it is being typed
  - Checks for errors
  - Thinks about overall design



What reasons would you have for not wanting to do this?

Your Reasons for Avoiding Pair Programming

- Distracting
  - Chit chat
  - Lose train of thought
  - Sexual tension
- Pressure from the observer
- Inferiority complex
- Arguments
- Half as efficient
- Hate the other person
- Smells

What are the benefits?

- Expect to get better code
  - Fewer bugs – (less expense)
  - Simpler design
  - More maintainable
- Learn more in less time
  - Share whole project knowledge throughout a team
  - Specific and general knowledge/techniques shared

- Expect more code
  - **Sometimes** more LOC from pair programming hour than for 2 individual programming hours
- Better time management
  - More focused work than when done alone
- More enjoyment
  - Social, supportive, rewarding activity

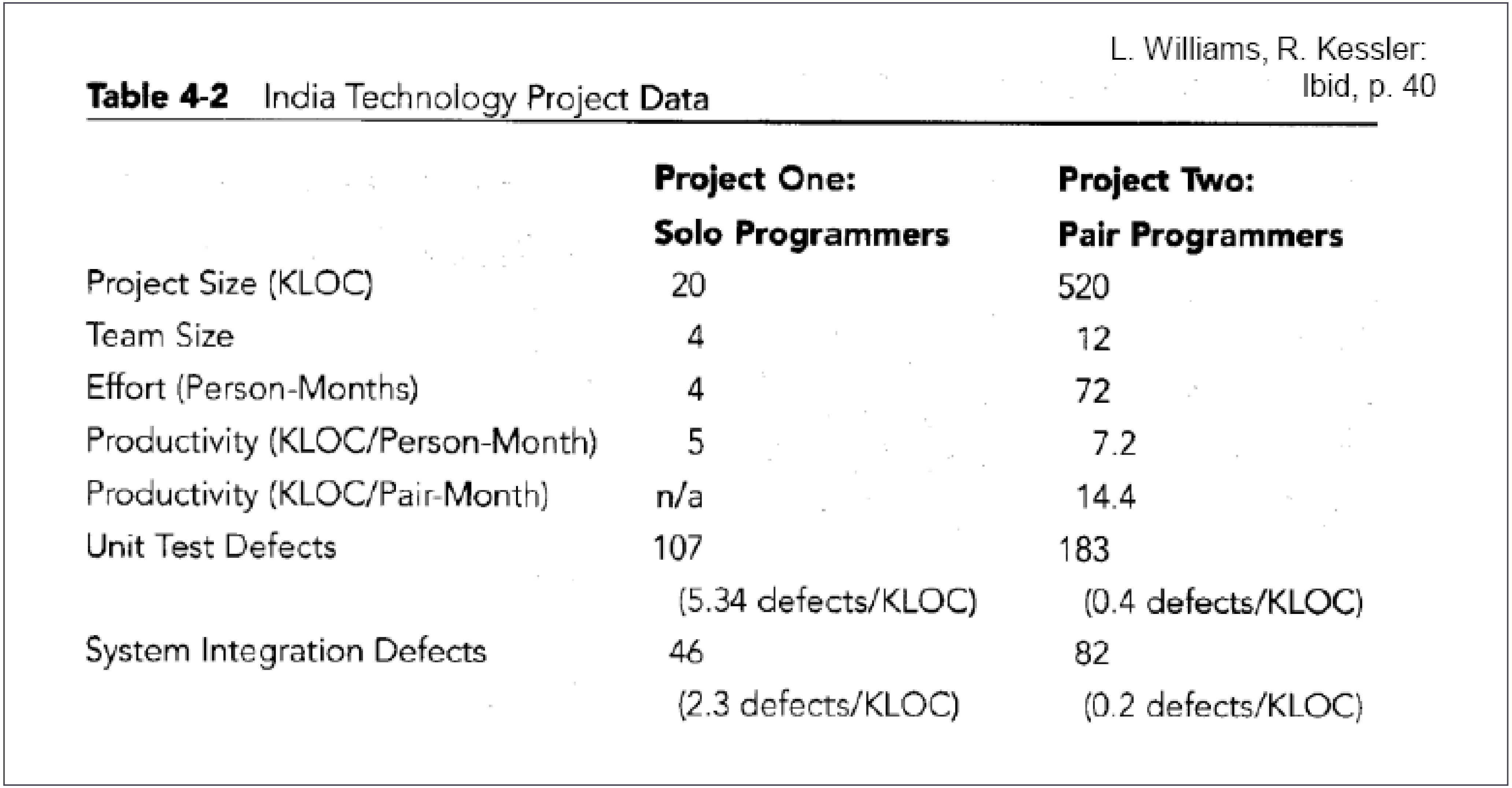
What are the benefits?

Research Results

- *Williams, Laurie, Kessler, Robert R., Cunningham, Ward, and Jeffries, Ron, "Strengthening the Case for Pair-Programming", IEEE Software, July/Aug 2000*

University study with 41 students

- Higher quality code
  - Test cases passed individuals: 73.4% - 78.1%
  - Test cases passed pairs: 86.4% - 94.4%
- Pairs completed assignments 40-50% faster (average 15% higher costs)
- Pair programming preferred by students (85%)

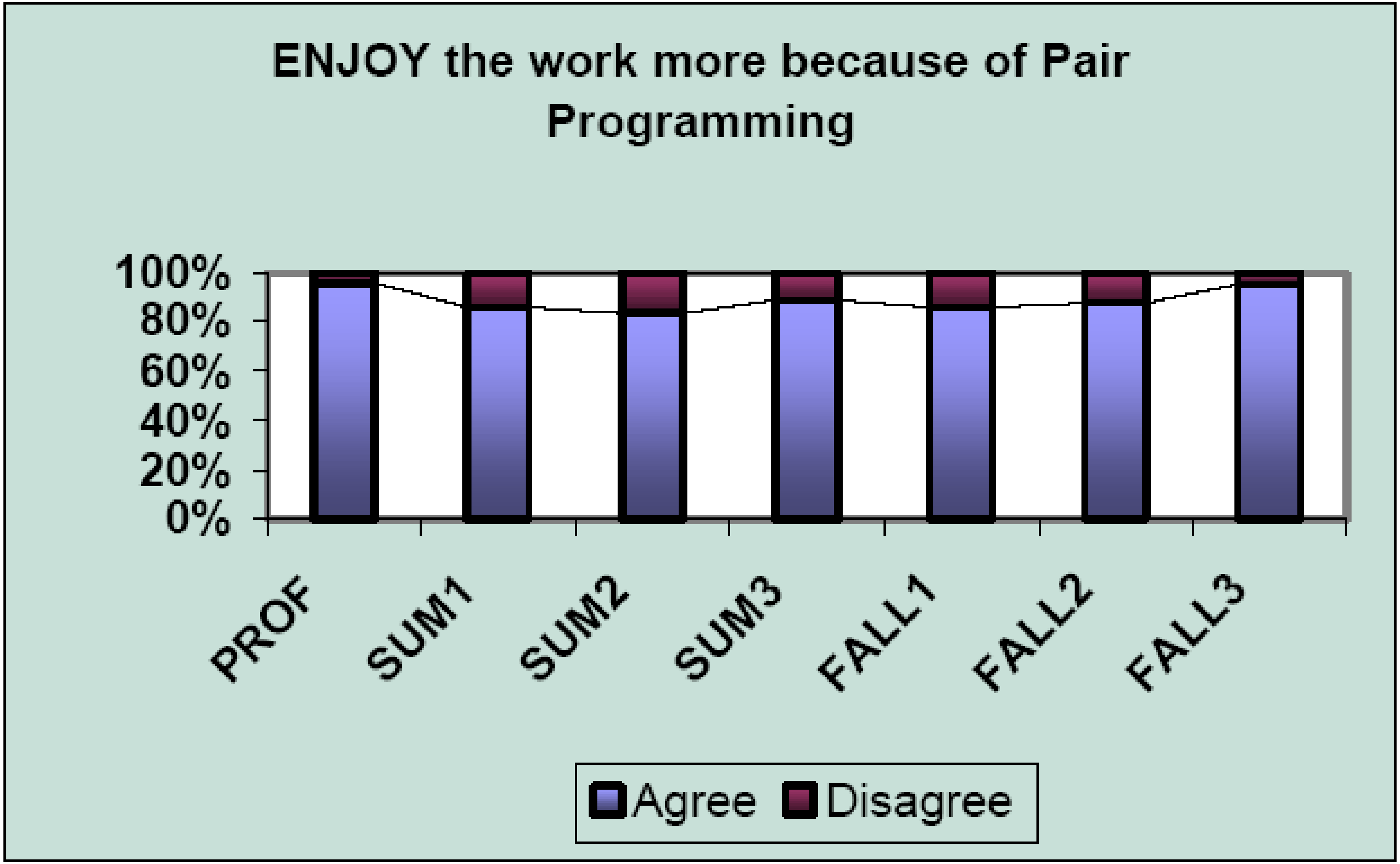


L. Williams, R. Kessler:
Ibid, p. 40

Table 4-2 India Technology Project Data

	Project One: Solo Programmers	Project Two: Pair Programmers
Project Size (KLOC)	20	520
Team Size	4	12
Effort (Person-Months)	4	72
Productivity (KLOC/Person-Month)	5	7.2
Productivity (KLOC/Pair-Month)	n/a	14.4
Unit Test Defects	107 (5.34 defects/KLOC)	183 (0.4 defects/KLOC)
System Integration Defects	46 (2.3 defects/KLOC)	82 (0.2 defects/KLOC)

Cockburn, Alistair & Williams, Laurie (2000), "The Costs and Benefits of Pair Programming", Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2000)



Guidelines for Pair Programming

- Start with a specific achievable task to complete in this pairing session (maybe 2 hours)
- Agree on a small starting task to work on

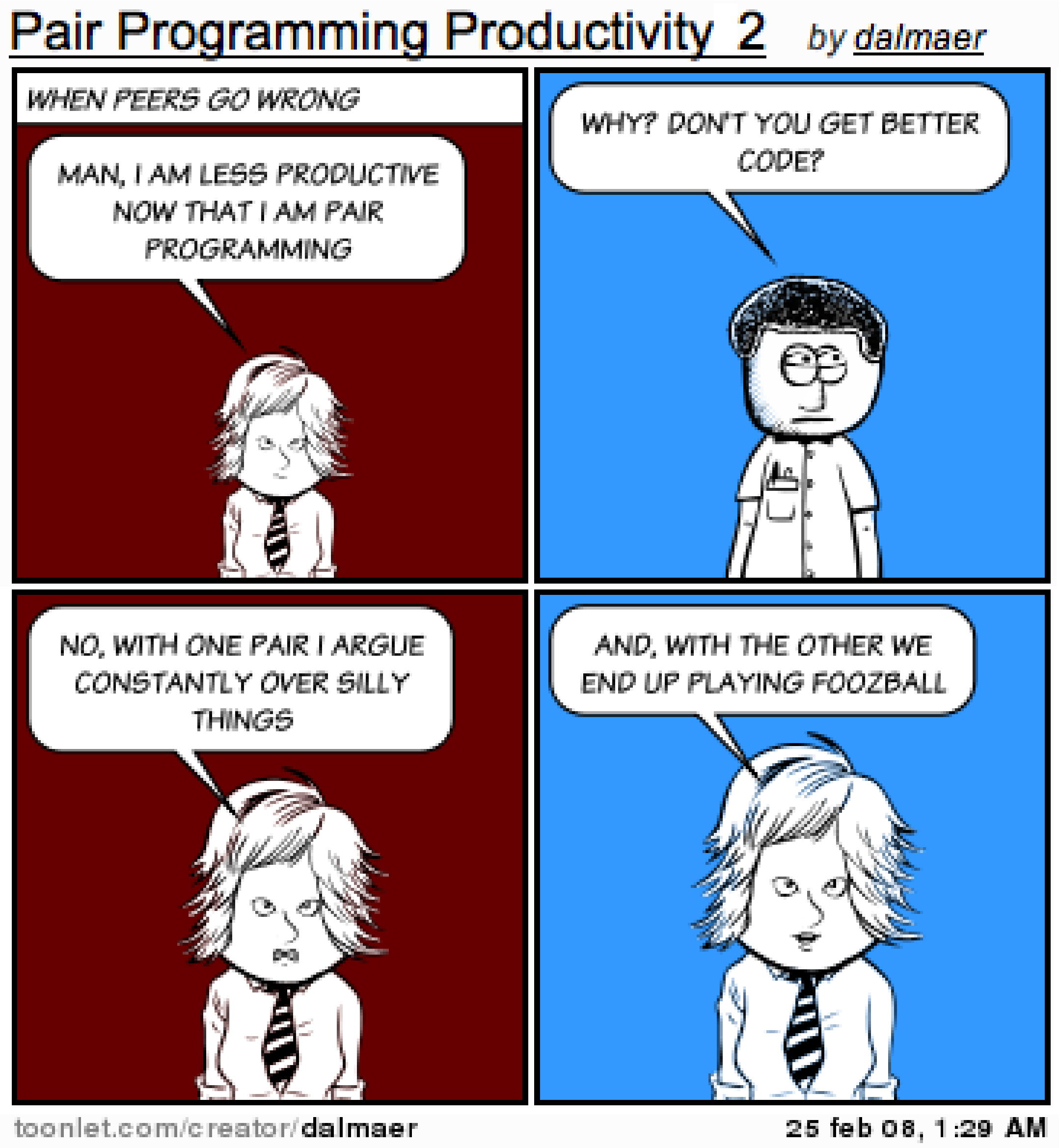
As the Driver

- Complete the current small task as quickly as possible. Don't worry about large design issues initially – the Observer will think about this
- **Talk a lot as you do the work** – ask for implementation ideas, ask if they know a better way to do this or if this looks correct to them etc.

Guidelines for Pair Programming

As the Driver

- Try to **say what you are going to do** almost before you do it
- **Ask your partner** "Will we write the …. now?"
- It is OK to pass the keyboard to your partner and say "**Show me**" if you don't understand their suggestion



Pair Programming Productivity 2 by dalmaer
WHEN PEERS GO WRONG
MAN, I AM LESS PRODUCTIVE NOW THAT I AM PAIR PROGRAMMING
WHY? DON'T YOU GET BETTER CODE?
NO, WITH ONE PAIR I ARGUE CONSTANTLY OVER SILLY THINGS
AND, WITH THE OTHER WE END UP PLAYING FOOZBALL
toonlet.com/creator/dalmaer
25 feb 08, 1:29 AM

Guidelines for Pair Programming

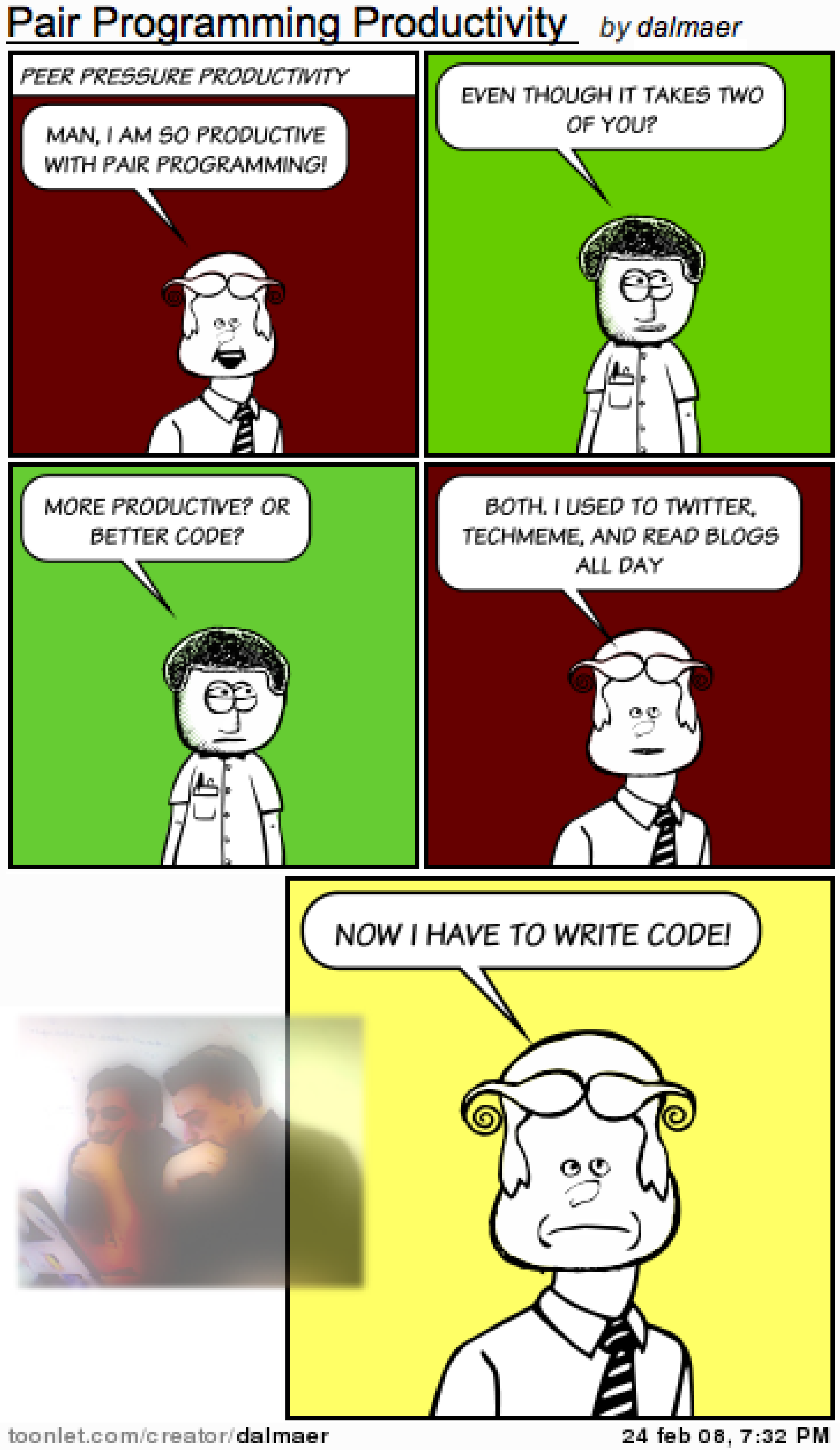
As the Observer

- **Read the code** as the Driver types it
- Spot possible bugs, typos, or unreadable parts of the code etc. – bring these up **when the current line of code is finished**
- Think about **ways to simplify** what is being done – wait until the small task is complete before raising these issues
  - **Keep notes** if they would help about things that need addressed

Guidelines for Pair Programming

As the Observer

- When asked a question **respond quickly**
- It is OK to say "**Can I drive?**" when it is easier to type what you mean than verbally explain it
- **Keep in sync** – if you become unsure about what is being done then ask questions straight away - DO NOT DELAY
- **PAY ATTENTION to everything – 100% on the task in hand**

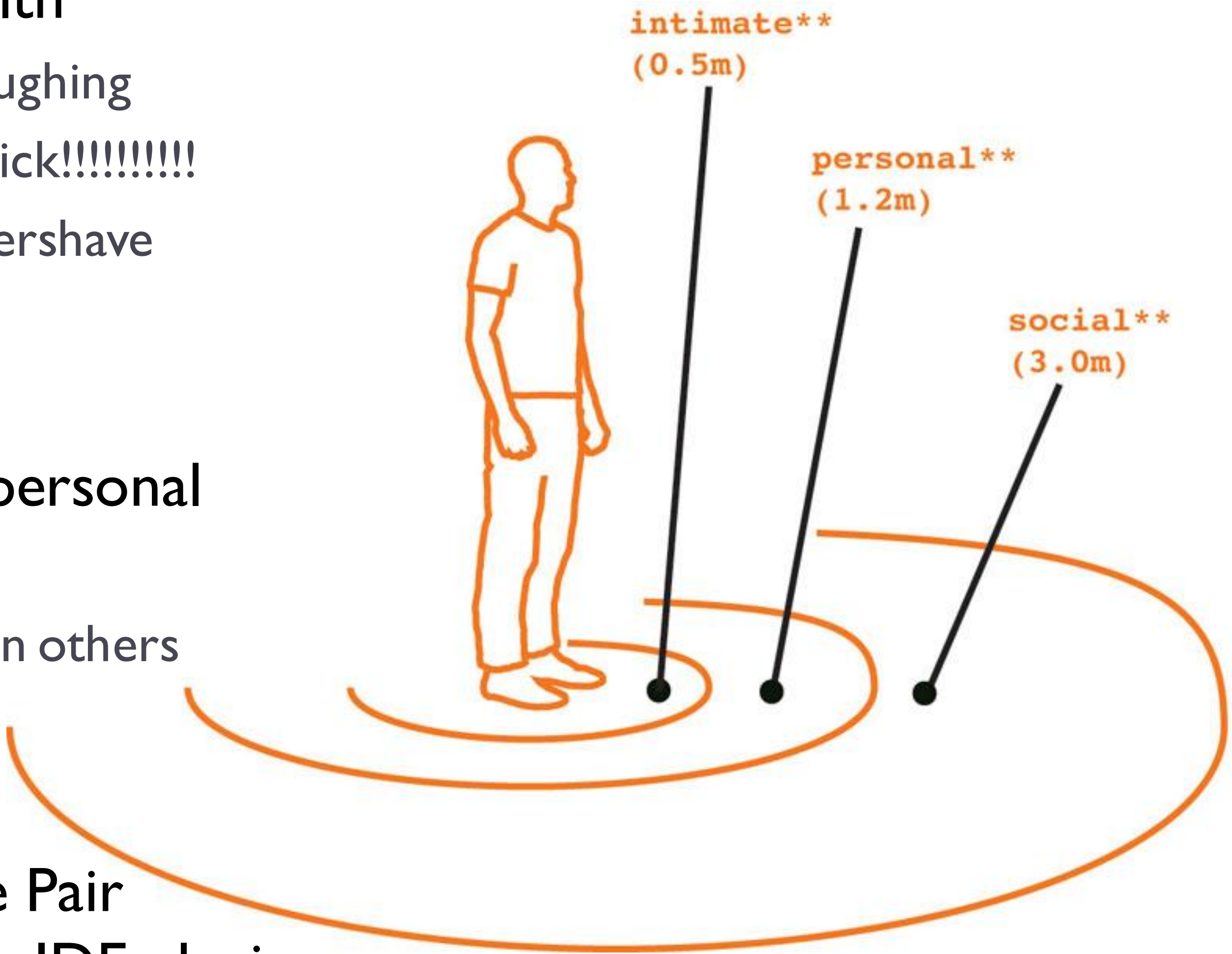

Pair Programming Productivity by dalmaer
PEER PRESSURE PRODUCTIVITY
MAN, I AM SO PRODUCTIVE WITH PAIR PROGRAMMING!
EVEN THOUGH IT TAKES TWO OF YOU?
MORE PRODUCTIVE? OR BETTER CODE?
BOTH. I USED TO TWITTER, TECHMEME, AND READ BLOGS ALL DAY
NOW I HAVE TO WRITE CODE!
toonlet.com/creator/dalmaer
24 feb 08, 7:32 PM

Guidelines for Pair Programming

- Enjoy any small successes along the way
- Switch roles often
  - At least every half hour (or one **Pomodoro**!)
- Be especially courteous
  - Say "Thankyou" (or similar) when an error is highlighted
  - Be *gentle* when highlighting an error – avoids offending egos
- The least experienced person should perhaps do more of the driving

Guidelines for Pair Programming

- Personal Hygiene and Health
  - Cover your mouth when coughing
  - Don’t come to work when sick!!!!!!!!!!
  - Avoid strong perfume or aftershave
- Be aware of the need for personal space
  - Some people want more than others
- It is possible to do remote Pair Programming using various IDE plugins
  - Useful for distributed teams



Take home messages

- Pair programming involves two developers working at the same computer together
- The same work may take slightly longer than if done by individuals but studies have shown that it is not twice as long
- The results produced are likely to be of higher quality and hence the practice will provide better value in the long run
- The developers take the Driver and Navigator roles
- There are guidelines on how to carry out these roles effectively
- Published results have shown that student developers enjoy this practice and perform better than when working alone
- This has been demonstrated by the students on this course

I would encourage you all to consider doing this – and vary your pairings