



Pair Programming

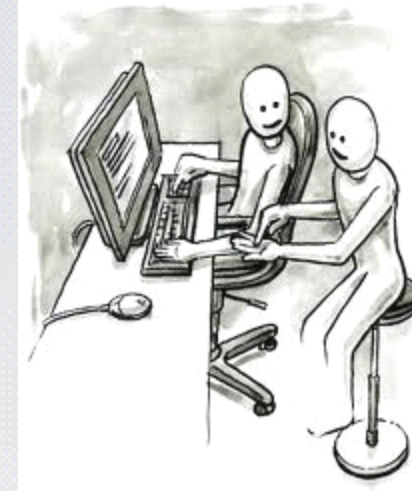
By Hanchao Wu

Outline

- **What is Pair Programming**
- History
- Motivation
- Techniques
- Why it works
- Problems
- Challenges
- When it is not working
- Conclusion

What is Pair Programming

- Two programmers code together at one workstation
- Driver types codes, and observer review and critiques it.
- Two switch roles periodically
- Note pair programming is not mentoring, even if one is significantly more experienced



Outline

- What is Pair Programming
- **History**
- Motivation
- Techniques
- Why it works
- Problems
- Challenges
- When it is not working
- Conclusion

History 1

- People have started pair programming long before it is called so.
- Fred Brooks announced "Fellow graduate student Bill Wright and I first tried pair programming when I was a grad student (1953-1956). We produced 1500 lines of defect-free code; it ran correctly first try."
- In the early 1980s, Larry Constantine reported observing "Dynamic Duos", producing code faster and more bug free.

History 2

- In 1998, Temple University professor John Nosek was the first to run an empirical study on the efficacy of pair programmers.
- In the late 1990s/early 2000s, pair programming practice is brought to forefront

Outline

- What is Pair Programming
- History
- **Motivation**
- Techniques
- Why it works
- Problems
- Challenges
- When it is not working
- Conclusion

Why Pair Programming (1)

- Back up each other

- Agile will put one's sick or leave into consideration
- Pair programming is a good solution to people's sick or leave
- It could reduce the risk of project failure due to one person's leave
- When one is sick or out of office, the other guy could continue working. It will reduce that influence to project.

Why Pair Programming(2)

- Improve the code quality

- It is one kind of code review; maybe we could call it continuous review.
- When one feel tired, the other guy could drive. The it is also one kind of work-relax cycle. It could give someone a breath.
- Two people could argue and find the best algorithm for one problem, or function.
- During paring, observer could challenge the coder, and find some small errors online.

Why Pair Programming (3)

■ Knowledge Transfer

- It is an good option to use pair programming to reduce learning curve for new people.
- With some different skill set and experience, people could learn from each other.
- Everyone could become expert with shuffling pairs.

Why Pair Programming (4)

- Share responsibilities and Share Pressure
 - Pairing Programming will allow pressure to be shared by partners, instead of just one.
 - They could also encourage each other during pairing
- Enhance Trust of Team Members
 - Successfully pairing could help team members believe each other
 - Partners will also learn each other's skill set

Outline

- What is Pair Programming
- History
- Motivation
- **Techniques**
- Why it works
- Problems
- Challenges
- When it is not working
- Conclusion

Techniques: Ping-Pong Pair Programming

- A writes a test, and make sure it fails.
- B works on the productions codes, and make sure it will pass the tests.
- B write another test, and see it fails.
- A starts working on the codes, and it will pass the test.

Techniques: Cross-Functional Pairing

- Only for embeded system development
- Instead of two software engineer, one software engineer works with one hardware engineer.
- Include more time to work alone
- If the platform is well known, it may not apply to the Cross Functional Pairing, because my problems have been resolved.
- If the platform is new and unused previously, that is the ideal scenario where we apply the Cross Functional Pairing.

Techniques: Distributed Pairing

- Only work when teams are geographically distributed, e.g someone works from home
- A good network is essential
 - if the lag time is over 1 second, what the observer says does not make sense.
- More tiring than traditional pair programming
 - do not plan to work 8 hours per day
- Tools:
 - Yuuguu, Mikogo, Trellis, ICICLE, gIBIS,...

Techniques: Others

- Selective Pairing
- Keyboard n Mice Pairing
- ...

Outline

- What is Pair Programming
- History
- Motivation
- Techniques
- **Why it works**
- Problems
- Challenges
- When it is not working
- Conclusion

Why Pair Programming Works

- Continuous Code Review
- Fewer blockages
- Masking distractions
- Guaranteed focus
- Multiple points of view
- Reduced training cost and time



Continuous Code Review

- Code Review is good, so continuous code review is great.
- Code reviewer could surface the code, and continuous code review could help observer understand why it is coded so, and review code line by line.
- Feedback is online and quicker.

Fewer Blockages

- Blind spots are rarely shared.
 - Blind of one person may not be a problem to the other guy.
- People with different skill set and experiences could learn from each other, so more experts will come out from a company.
- Pairing could put up with a better solution than one person does.

Masking distractions

- "White noise" could keep your brain from background noise.
 - when you pair programming, you will find out the noise from other cube is not a problem any more
- The conversation will keep you focus on what you are coding.

Guaranteed focus

- Pairing programming could let people really focus on productive coding, instead of just surfing the web, checking the email, or reading blogs.

Multiple points of view

- Explaining codes to someone could help people find potential bugs.
- Explaining could help people really really understand the codes, and avoid inconsistency between thoughts and codes.
- Explaining could also avoid some stupid errors in the first place.

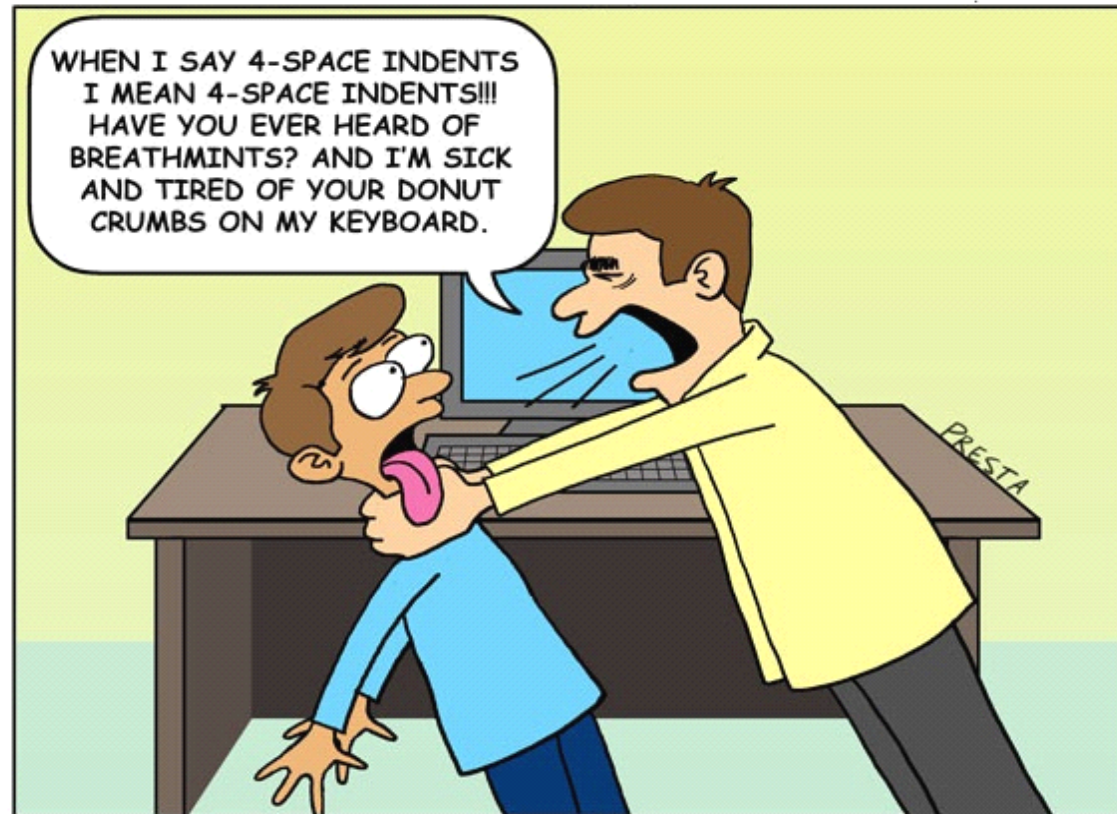
Reduced training cost and time

- Developers are also trainers.
- Instead of learning in class, people could learn from job.
- Developing time is also the training time.
- Better than just 5-day training in class, and it also increases the productivity.

Outline

- What is Pair Programming
- History
- Motivation
- Techniques
- Why it works
- **Problems**
- Challenges
- When it is not working
- Conclusion

Problems



The dark side of pair programming.

Problems

- Disagreements
- Scheduling Conflicts between partners
- Absence of partners
- Rushing
- Over-confidence
- ...

Outline

- What is Pair Programming
- History
- Motivation
- Techniques
- Why it works
- Problems
- **Challenges**
- When it is not working
- Conclusion

Challenges (1)

- It is hard to convince managements to believe in pair programming.
 - It is easy for managers to think that pairing programming will do nothing but simply to reduce or even to halve the productivity; therefore, it is hard to convince them.
 - Management teams needs some agile training to believe that pairing programming will work.

Challenges (2)

- If the skills and experiences between partners are huge different, it may reduce productivity.
 - If the background and experiences of two partners are very different, it is possible that they could not understand each other.
 - It may takes too much time for one guy to explain codes to the other.
 - Even worse, when one drives, the other one may fall asleep.

Outline

- What is Pair Programming
- History
- Motivation
- Techniques
- Why it works
- Problems
- Challenges
- **When it is not working**
- Conclusion

When Pairing is NOT Working (1)

- Research work
 - Pairing should not start until coding.
 - If still in research part, it is preferred to be separated.
- Both have no idea how to implement project
 - Two inexperienced people, working together, may reduce the productive
 - An inexperienced people need to pair with an experienced people.

When Pairing is NOT Working (2)

- Trivial work

- We use pairing to increase productivity. If the work itself is trivial, pairing is a waste of resource.

- People hate each other

- If partners hate each other, pairing will become a disaster.
- We want pairing to build relationship, not to hate each other more.

- One person is not around

- When one person is sick, or have to deal with some person affairs, the other guy may need to work alone.
- We do not need a temporary pairing.

Conclusions

- Pair Programming could increase productivity
- It could also reduce bugs, and give feedback more quickly.
- It is one kind of continuous code review.
- It does not apply to every situation. For some scenario, pair programming is not necessary.

References

- Distributed Pair Programming: Empirical Studies and Supporting Environments
- Pair Programming
 - - Laurie Williams
- The costs and Benefits of Pair Programming
 - -Laurie Williams

Resources

- <http://collaboration.csc.ncsu.edu/laurie/publications.html>
- http://en.wikipedia.org/wiki/Pair_programming
- <http://c2.com/cgi/wiki?PairProgramming>
- http://www.youtube.com/watch?v=rG_U12uqRhE