What Pair Wins?

Keith Callis
Principle Consultant, Strategic Data Systems
Keith.Callis@sds.io

https://www.linkedin.com/in/keithcallis

About Me

- ✓ Principal Consultant at SDS, Inc., based in South East Ohio.
  - ✓ Software developer.
  - ✓ Scrum Master (Scrum Alliance).
- ✓ Recent agile projects (2015 to present):
  - ✓ (DevOps) SQL Server, SSIS.
  - ✓ (Pivotal Tracker) Java and C# web application, SQL Server/H2.
  - ✓ (Rally) Hadoop, Oracle, MongoDB, Java, Scala.
- ✓ When not writing or breaking software, he is amateur photographer or chef in his kitchen.

https://www.linkedin.com/in/keithcallis

Strategic Data Systems – What we do:

- Application Development and Application Renovations
  - Architecture, Development, and Architectural Design Reviews
  - .NET, Java
- Agility - Agile Coaching and Transformations
- Training - Agile, .NET
- Our place or yours!
- We are looking for .NET/Java developers, Scrum Masters and Product Owners.

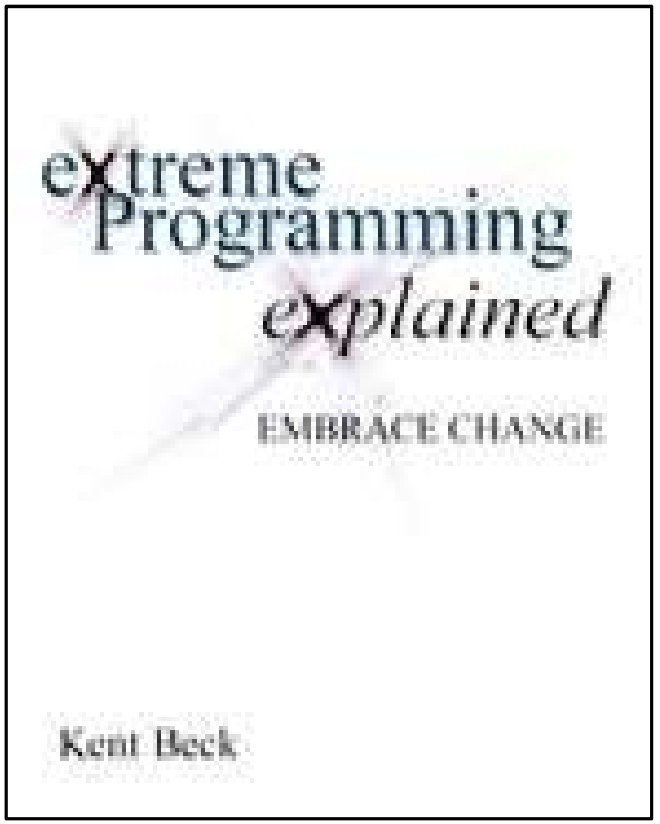
https://sds.io/

Pair Programming

- What is it?
- When did it start?
- “Pair Programming” status?
- Experiences and suggestions.

Extreme Programming History

- 1996 Chrysler Comprehensive Compensation System (C3)
- People involved
  - Kent Beck
  - Ward Cunningham
  - Ron Jeffries
  - Martin Fowler
  - Others



- https://en.wikipedia.org/wiki/Extreme_programming

XP: Planning

- User stories
- Release planning
- Frequent small releases
- Iteration planning

XP: Managing

- Team open work space
- Sustainable pace
- Standup meeting
- Measure velocity
- Move people around
- Fix XP via retros

XP: Designing

- Simplicity
- System metaphor
- CRC cards for design sessions
- Use spikes as needed to reduce risks
- Keep functionality to minimum (MVP)
- Refactor as needed

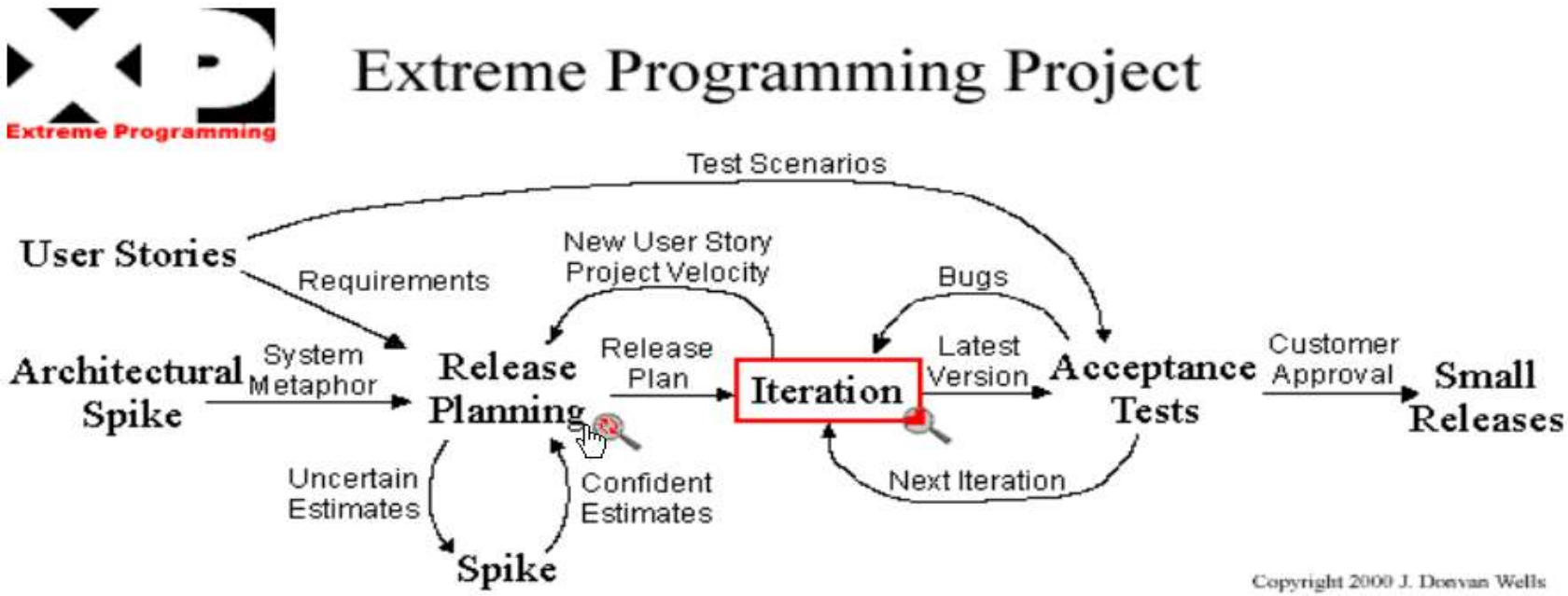
XP: Coding

- Customer is available
- Code standards
- Unit test first
- Pair programming
- Integrate code per pair
- Integrate often
- Dedicated integration computer
- Collective ownership

XP: Testing

- Code must have unit tests
- Code must pass all unit tests before release
- If bug found, create test for bug, fix code, run tests
- Acceptance tests usually by user/customer

Extreme Programming Project



Manifesto for Agile Software Development

- 2001 meeting
  - Kent Beck, Ward Cunningham, Dave Thomas
  - Jeff Sutherland, Ken Schwaber, Jim Highsmith
  - Alistair Cockburn, Robert C. Martin, Mike Beedle
  - Arie van Bennekum, Martin Fowler, James Grenning
  - Andrew Hunt, Ron Jeffries, Jon Kern
  - Brian Marick, Steve Mellor

https://en.wikipedia.org/wiki/Agile_software_development

Manifesto for Agile Software Development

from https://agilemanifesto.org/

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile Methodologies

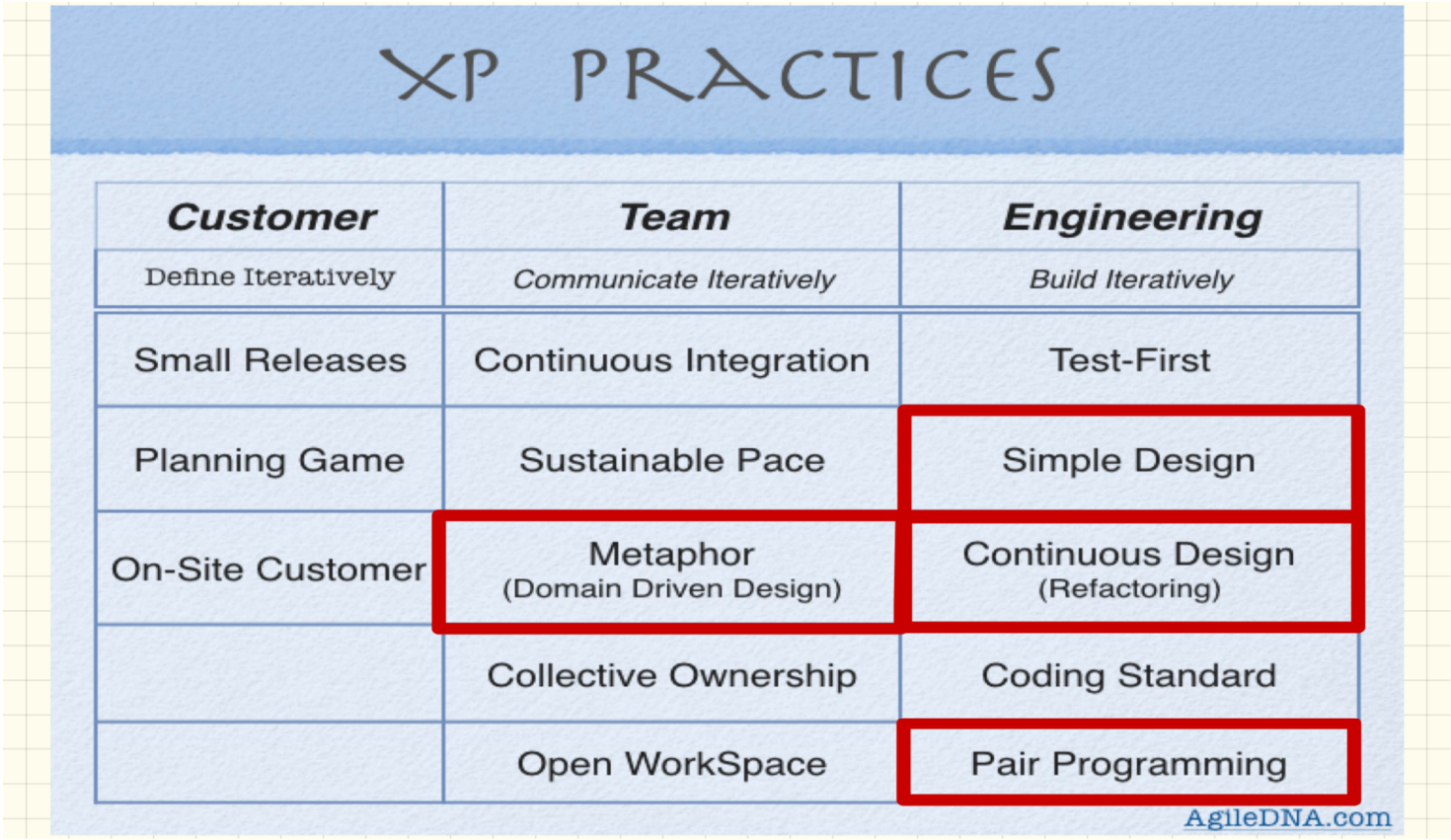
- XP
- Scrum
- Kanban
- Lean
- …

- Pair programming can be used for all

Extreme Programming Status

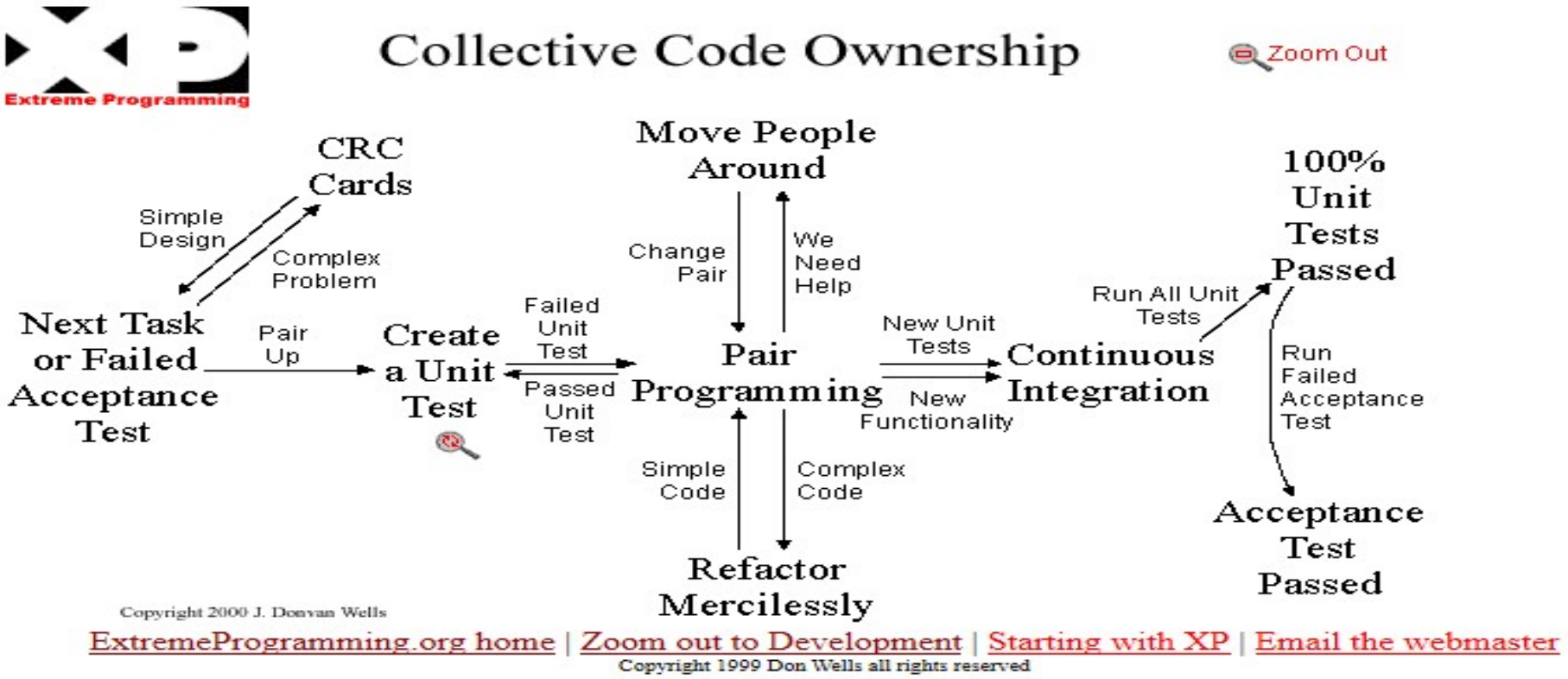
From Mark Windholtz's presentation at Cincy Deliver conference 2019:

Rediscovering XP: Extreme Programming. Retro? or Resurgent?



https://www.linkedin.com/in/keithcallis

Extreme Programming Coding



Pair Programming for Development?

- ❑How many Companies have tried Pair Programming?
- ❑How many Teams have tried Pair Programming?
- ❑How many Developers have done Pair Programming?

One Developer

One Computer

- Work on own
- Member of a team
- Work story on your own
- Ask team members for assistance
- Code review with team member?
- Testing may not be complete?

My Experience

- 2013
- Picked up story
- Team mate said to do “ABC” for story
- Original test **failed**, original then **passed**, then 22 tests **failed**
- Team mate reviewed and said that I should have done “XYZ”
- Reverted code, original test **failed**, then original test passed
- All other tests **passed** (~1,800)

Two Developers Share at Same Time

One Computer

- Work as a team
- Share keyboard/mouse
  - Slide keyboard
  - Move mouse
  - Maybe adjust monitor
- Work story as a pair
- Ask pairing partner for assistance
- Code review with each other
- Testing should be complete

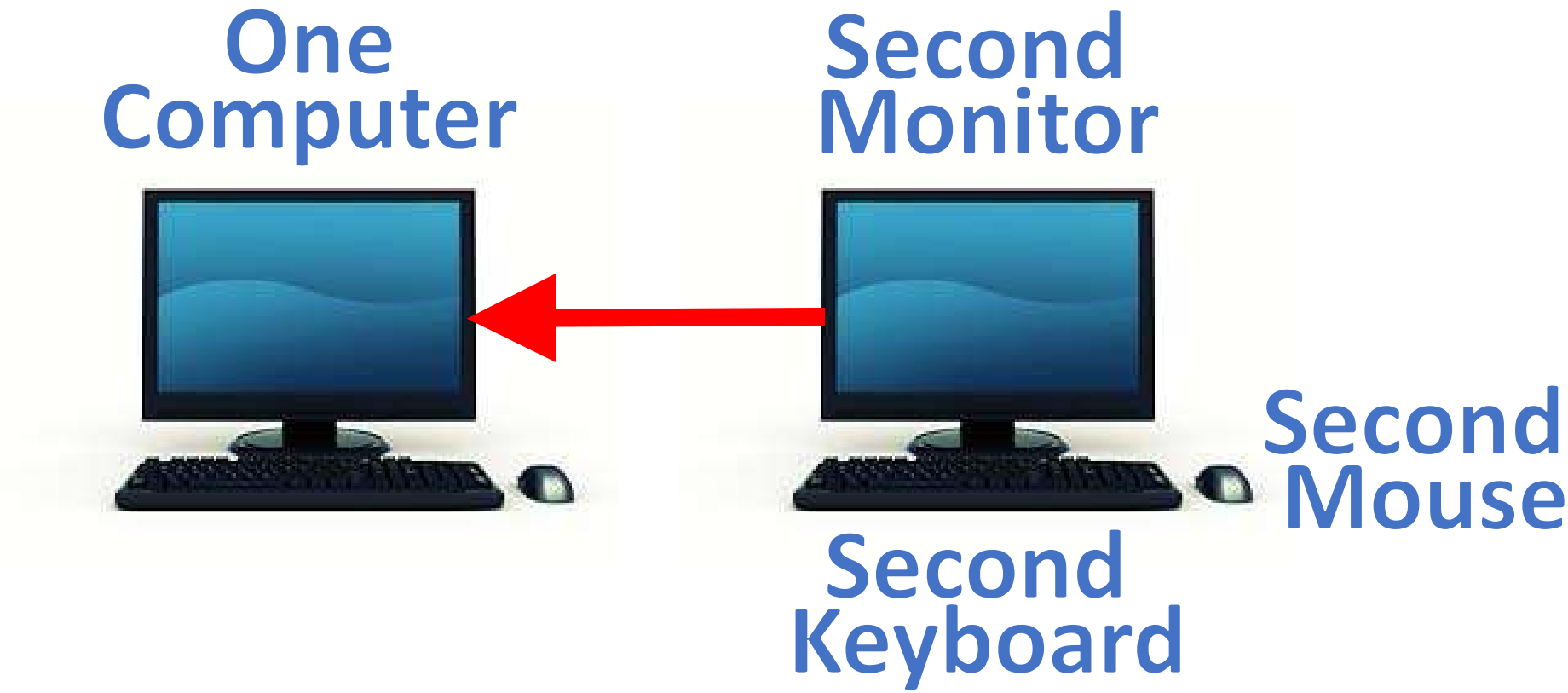
My Experience

- 2009, September
- SQL Server Stored Procedure 8+ hours to run
- Asked manager if we could pair to improve performance (Thu morn)
- 1 computer, 1 keyboard, 1 mouse, in a cube
- Switched around Thu after lunch
- Fri afternoon performance 20-30 minutes
- Did not know that manager wrote original SQL procedure

My Experience (poor)

- 2017
- Scala code with Hadoop data
- Tried to pair with person sitting next to me
  - My computer/monitor
  - Person wasn't interested in participating
- Tried to pair with their computer
  - Probably thought they were being reviewed
  - Afraid of showing not enough knowledge of program

Two Developers Share at Same Time



Two Developers Share at Same Time

- Work as a team
- Share computer more frequently, e.g.
  - Switch every 10 to 20 minutes
  - One writes test, then other writes code
  - More give/take ideas
- Work story as a pair
- Ask pairing partner for assistance
- Code review with each other
- Testing should be complete

My Experience

- 2019, April
- Switched pairing partner every day
- Stories small, usually 1 point
- PO sat among developers
- CI/CD shown on common screen (for deploys)
- Remote pairing worked with no loss of productivity

My Experience (cont)

- Breaks (Morning and Afternoon)

Pairing Costs/Benefits

- Two developers, twice man-hours?
  - Better quality of code
  - Less QA needed
- Offset Costs
  - More staff know application
  - Build team comradery

Pairing Costs/Benefits (cont)

- Better design
  - Two members contributes to single solution
- Communications
  - Improved between members and with team
- Less interruptions
  - Phone calls/Emails
  - Breaks

Pair Programming Success Items

- Switch driver often
  - Helps junior members learn code/testing
  - Helps inexperienced members code/testing
  - Helps experienced member to teach/share knowledge
- Avoid working alone when partner is away
  - Review tests/code

Pair Programming Success Items (cont)

- Take breaks away from computer
  - Morning break
  - Lunch same for everyone
  - Afternoon break
- Share knowledge of application
  - Banking, general ledger, manufacturing, ordering, shopping cart

Pair Programming Success Items (cont)

- Know when to listen and when to talk
  - Let each other have their say since they may have a different viewpoint
  - Keep the conversation going
- Be a student and/or teacher
  - Learn from your partner
  - Teach your partner

Pair Programming Success Items (cont)

- Share knowledge level of code
  - HTML, CSS, Javascript
  - C#, Java, Scala
  - Database SQL (SQL Server, Oracle, H2, no-SQL)
- Celebrate success
  - Fist bumps, high fives
  - Share at retros

Pair Programming Success Items (cont)

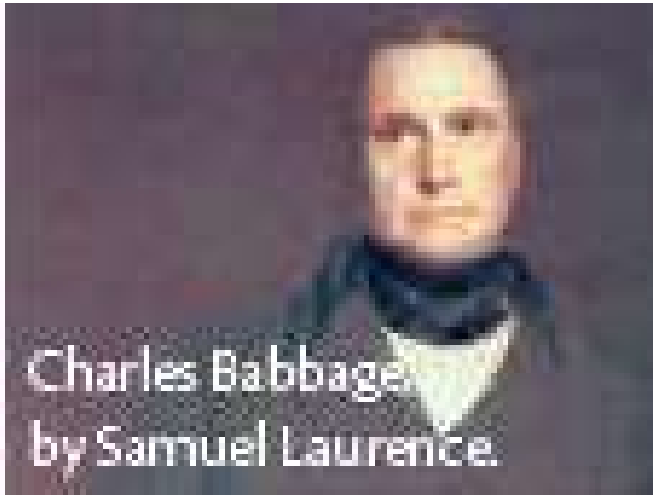
- Learn from troubles
  - Code issues
  - Database issues
  - Application issues
- Hold retro at end of day
  - Discuss good events from the day
  - Pass on what should be learned from the day
  - Be honest

Pair Programming Success Items (cont)

- Good Hygiene
  - Body
  - Breath
  - Clean hands, especially if sharing keyboard/mouse
- Successes
  - Pairing level
  - Team level
  - Company level

Maybe First Pairing?

Ada Lovelace



Charles Babbage

From: http://hucknallparishchurch.org.uk/wp-content/uploads/2015/11/dl_ada_2015_web_LR.pdf

Questions?

¿

Keith.Callis@sds.io
https://www.linkedin.com/in/keithcallis