PAIR PROGRAMMING

A PRIMER

WHAT IS IT?

- 2 DEVELOPERS, WRITING CODE AS A TEAM
- SOLVING PROBLEMS WITH COMMUNICATION

WHAT IS IT NOT?

- LOOKING AT CODE TOGETHER
- LOOKING OVER SOMEONES SHOULDER AND TELLING THEM WHAT TO DO
- ONE DEVELOPER PROGRAMMING, ONE CHECKING SLACK, PHONE, TWITTER

HOW

THE PAIRING STATION

- 2 CHAIRS
- 2 KEYBOARDS
- 2 MICE
- 2 MONITORS
- 1 COMPUTER

THE SOFTWARE

- AN IDE BOTH DEVELOPERS ARE COMFORTABLE WITH
  - SCREENHERO IF ONE IS REMOTE

TECHNIQUES

- POMODORO
- PING PONG
- NATURAL FLOW

THE KEY IS TO BE DELIBERATE - ESPECIALLY WHEN STARTING. TRY DIFFERENT STYLES AND GIVE FEEDBACK ABOUT WHICH STYLES WORKED BEST. IT SHOULD FEEL LIKE A CONVERSATION.

ROTATION

- WE'VE SETTLED AT MAVENLINK ON ROTATING EVERY DAY
  - FIND A CADENCE THAT WORKS FOR YOU

LET THE JR DRIVE

THIS CAN BE HARD

- SIT ON HANDS
- TURN KEYBOARD AROUND
- UNPLUG IT EVEN!

REMEMBER: "GO SLOW TO MOVE FAST" - LEVELING UP YOUR PAIR HELPS EVERYBODY MOVE FAST IN THE END.

DO IT OFTEN

START CONSISTENTLY

- WE SET UP A A PAIRING ROTATION EACH DAY AT OUT DAILY STANDUP
- PAIRING STARTS PROMPTLY AT 9:15 AFTER STANDUP

MINIMIZE DISTRACTIONS

HAVE A PLAN

- "TODAY I'D LIKE TO FOCUS ON WRITING GREAT TESTS"
- "I WANT TO UNDERSTAND EVERY LINE OF CODE WE WRITE TODAY"
- "I WANT TO UNDERSTAND HOW JRUBY MAKES ANYBODIES LIFE EASIER"

GIVE FEEDBACK AND RECEIVE FEEDBACK

- IF YOU ENJOYED THE PAIRING SESSION, LET YOUR PAIR KNOW WHAT WENT WELL E.G. "THAT WAS A REALLY FUN REFACTOR"
- IF IT COULD HAVE BEEN BETTER, EXPLORE WITH YOUR PAIR WHAT YOU THINK COULD HAVE BEEN IMPROVED E.G. "I COULD HAVE COMMUNICATED MY IDEAS BETTER BEFORE JUMPING AHEAD AND I THINK I LOST YOU"

TAKE FREQUENT BREAKS

- DRINK A LOT OF WATER, IT WILL REMIND YOU TO TAKE A BREAK
- TAKE TIME FOR LUNCH
- DESIGNATE TIME TO CHECK EMAIL/SLACK

PAIRING SCHEDULE

- ROTATE TO BREAK DOWN SILOS
- "CLAMOR FOR WORK"
- IF THERE IS AN AREA OF CODE YOU HAVEN'T SEEN YET BE PROACTIVE ABOUT GETTING CONTEXT ON IT

A NOTE ON CODE OWNERSHIP

- CODE OWNED BY AN INDIVIDUAL DOES NOT GET THE BENEFIT OF BEING REVIEWED MULTIPLE EYES AND HARDENED BY MULTIPLE IDEAS
- BUILDS A DEPENDENCY ON CODE OWNER, WHICH IS NOT SUSTAINABLE

WHY

CREATIVE SOLUTIONS

SOLVING PROBLEMS THROUGH COMMUNICATION

- RUBBER DUCKING (BUT WITH A PERSON)
- OUR ABILITY TO SOLVE THE RIGHT PROBLEM IS CONSTRAINED BY OUR UNDERSTANDING OF THAT PROBLEM (THE CODE IS THE EASY PART)
- TWO HEADS ON THE SAME PROBLEM WILL NOT ONLY LEAD TO BETTER CODE BUT ALSO BETTER UNDERSTANDING OF THE REASON WHY WE ARE WRITING THAT CODE

INCREASE FOCUS

- HOW MANY TIMES A DAY ARE YOU CHIMING IN ON VARIOUS SLACK THREADS? MAKES CHECKING NOTIFICATIONS DELIBERATE AND TAKE TIME OUT OF DEVELOPMENT.
- OFTEN KEEPS THE PROBLEM SET SMALL

SHARE IDEAS

- YOU'VE BEEN AROUND THE BLOCK AND HAVE A CERTAIN WAY OF SOLVING A PARTICULAR PROBLEM.
- HAVING A SECOND SET OF EYES, HELPS YOU APPROACH THE PROBLEM DIFFERENTLY

CODE QUALITY

BETTER CODE/LESS DEFECTS/LESS BUGS

- TWO SETS OF EYES, FREQUENT ROTATION ON PROJECTS
- MORE LIKELY THAT ONE OF THE PAIR CAN SPOT A BUG BEFORE IT MAKES IT TO PRODUCTION

BUILT IN CODE REVIEW

- ROTATION FORCES REVIEW OF CODE EARLY AND OFTEN
- CONSTANTLY JUSTIFYING THE CODE YOU'RE WRITING TO YOUR PAIR

SHARE AND BE ACCOUNTABLE TO YOUR DEVELOPMENT PRACTICES AND STANDARDS

- PROPAGATE BEST PRACTICES
- NEW IDEAS CAN BE SHARED
- A PAIR CAN HELP MAINTAIN DISCIPLINE E.G. "LET'S FINISH THIS REFACTOR BEFORE WE START A NEW ONE"

KNOWLEGE DISTRIBUTION

CONTEXT SWITCHING

KNOWLEDGE TRANSFER / ELIMINATE SILOS

- WORKING ON A FEATURE THAT ONLY JAMES KNOWS ABOUT? HAVE HIM PAIR WITH YOU AND EXPLAIN ITS INNER WORKINGS.
- IF YOU KNOW SOMETHING THAT HASN'T BEEN SOCIALIZED PAIR ON IT TO SHARE CONTEXT WITH OTHERS

TEACHING/PERSONAL DEVELOPMENT

- START TRAINING A NEW DEVELOPER ON DAY 1 BY BUILDING FEATURES WITH THEM
- SINCE YOU'RE PAIRED, YOU DON'T NEED TO RUSH TO GET THEM CREDENTIALS AND TEACH THEM YOUR PROCESS BEFORE THEY CAN BE "PRODUCTIVE". JUST START WORKING.
- WE DON'T GIVE OUR ENGINEERS THEIR PERSONAL LAPTOPS UNTIL THE END OF THE FIRST WEEK.

"MOVE SLOW TO MOVE FAST"

- TAKE THE TIME TO INVEST IN YOUR CAPACITY TO BUILD SOFTWARE

"ALWAYS LEARNING"

- THE ROLE OF TEACHER AND LEARNER IS FLUID WITHIN A PAIRING SESSION
- ONE MINUTE TEACHING A NEW PATTERN OR BEST PRACTICE THE NEXT LEARNING A NEW SHORTCUT OR TECHNIQUE
  = YOU MAY BE SURPRISED WHAT YOU CAN LEARN FROM SOMEONE WITH FAR LESS EXPERIENCE
- BE READY TO TEACH - BUT ALSO BE READY TO LEARN
- IT'S IMPORTANT TO APPROACH THE PAIRING RELATIONSHIP AS

HOW DO I START?

- MAKE A PAIRING WORKSTATION
- FIND A PROBLEM
- FIND A PAIR
- MAKE A PLAN
- WRITE SOME CODE

QUESTIONS?