Mentorship Mutual frustration - Junior drives, senior navigates "Junior is slowing me down" - Junior makes mistakes and - Senior drives, junior frantically corrects tries to follow or tunes out Large Junior asks questions - Junior is a mouse and keyboard - Senior asks *leading* questions to senior's commands Lots of learning - Poor output, hurt morale **Effectiveness** Gap watch the gap in effectiveness between the two individuals - Ego driven development - Collaboration - Competition and friction - Share driving and navigating - Battle of opinions Small - Discussion, planning, thinking - One drives, the other tunes out - Lots of learning or strong - Tension, animosity output - Poor output Small (or negative) High Benefit Pair programming success matrix → Pair programming) good things happen when there is understanding between people empathy Understanding comes from communication Culture self confidence comes from the top it's not what they say, it's how they act Conclusions

```
brought by the extreme programming movement

√ you meet wonderful people

"it depends" whether it's worth it
hope that the business(and the individuals) will
benefit more than if the two worked independently.
                                                                      understand and respect each other
                                                                     communicate often
                                                    good chemistry /
                                 work out better
m with small effectivness gap
                                due to
                                                                      help each other learn
                                                    devs have more in common
                   more beneficial than senior pairs
                           more empathy to one another
                           devs have weak opinions
② junior pairs / why?
                          egos are smaller
                           devs care more about learning than being "right'
                          new kids in school effect
                  Primary goal: Learn
                   harder to get a good match
                               devs respect each other
                               talk a lot / discuss
                                             plan
                               devs are different
                               devs have different opinions
senior pairs
                               more experience, stronger opinions
                               more experience, more ego
                                                                     being insecure
                              if one dev has a large ego, the pair
                               may wield poor decisions due to
                                                                      "tiptoeing" to not hurt the other person's ego
                                                              they argue, not debate \boldsymbol{a} \boldsymbol{lot}
                               Both may have large egos ∫
                                                            can become tense, with animosity
                                                             no communication
                  Primary goal: Produce
                          requires a different approach
                                           new devs join the company
                                          devs switch to a new development context
                                           leadership must understand that the "junior" will be ramped up as fast as possibl
                                                                            other devs
                                          every one must understand this
                          expectations
                                                                            especially the senior in the pair
                                          no pressure to deliver
                                            really hard to do as humans since
                                            we tend to do the opposite
                                                          listen
                                                          answer questions
                                                          observe
                                                          ask leading questions
                                                          be patient
                                                           empathy
                                             🢡 helpful

  juniors with seniors

                                                          let them make the mistake you see coming __ and let them feel some of the consequences
                                                          let them think about why that was bad
                                                          let them correct it
                                                          let them ask you questions along the way
                             the senior
                                                          try to answer with questions
                            must mentor
                                                          if they get stuck __lead them out with questions
                                                              DO NOT tell them what to type
                                                             -{ DO NOT tell them what to do
                                                             DO NOT take over and show them
                                                     we want to build the junior up, not show him what you know
                                            Goals
                                                     Mentoring and Learning
                                                        everyone is busy trying to get stuff done
                                                        don't have human skills to do it well
                                                                 Do they share knowledge?
                                                                 Do they help juniors often?
                                            Questions you may
                                            ask yourself
                                                                 Do they get defensive about their?
                                                                 Are they comfortable with sacrificing for the good of the team
     programming is as much about people __as_it is about technology
                                          in colaboration with other people
      we are people writing software
                          great tech skills
     great programmer
                         being a great person
```