Introduction
Hello, my name is Miguel, and on behalf of Team BirdFeedr, I would like to welcome you to our presentation on the different scenarios of pair programming.

We would greatly appreciate if you could hold all questions till the end of the presentation. However, there will be some periods of audience interaction, in which we would appreciate participation.

Now before I hand the floor over to Thomas, let me provide a bit of definition and personal experience to you.

Pair programming: The agile software development technique where two programmers work at one terminal. One is the driver, who writes the code, and the other is the navigator, who reviews the code and is the one doing most the thinking and it is important that they switch roles throughout the session.

During this presentation, we will lead you on an exploration of 3 different pair programming scenarios: Expert-Expert, Expert-Novice, and Novice-Novice.

(Throughout our project we used pair programming. Helpful. Some of us thought of different things. Shared experience.)
We found Pair programming beneficial because it was helpful to have several minds thinking in parallel and having different perspectives.

Now sit back and enjoy your journey through different variations of pair programming.

Transition
Thank you Miguel for that introduction.

For our journey into the different styles of pair programming, you will be joining me as we step into 2 different shops were pairs are at work coding the Fibonacci sequence for their bosses.

Let us begin with Expert-Expert pairs.

Expert-Expert
Aww, I see our first pair, Di and Wei. Let's observe them at work...

(Di and Wei start)

Hold on, something about this does not seem right. Can anyone take a stab at what is wrong?
(Handle audience's response)

Continuing on our journey, I can see that Josh and Nate are having a discussion, let's listen in...

(Josh and Nate start)

Now that seems more like how pair programing should be. Both individuals are working to increase one another's productivity.

We don't want to linger too long, so let us move on to observing Expert-Novice pairs.

Expert-Novice
Once again it looks like Di and Wei are working away...

(Di and Wei start)

Looks like Di is having a rough day at work today. Does anyone care to comment on what they feel is wrong with this environment?
(Handle audience's response)

Let us see if Josh and Nate are having a better day...

(Josh and Nate start)

That seemed better for both Josh and Nate. It is always good to see pairs helping each other out and not berating a novice due to lack of experience.

Our journey is almost at its end, but before then, let us make a final stop and see some Novice-Novice pairs.

Novice-Novice
It would appear that Di and Wei are just starting off. Let's see how they are doing...

j(Di and Wei start)

Wow. At that rate, it looks like Di and Wei will not be getting their program written.

Let us see if Josh and Nate are having any better luck.

(Josh and Nate start)

That seems much better. Does anyone care to comment on what was done right?
(Handle audience's response)

I hope your enjoyed your journey with me through different styles of pair programming, and I will now hand the floor back to Miguel.

Conclusion
Thank you Thomas.

As you have seen, each pair programming scenario has its advantages and disadvantages.
The Expert-Expert scenario showed us that two experts can produce high quality of work though they might show little insight.
The Expert-Novice scenario showed us that a novice can bring new insight to an expert if the novice participates
And The Novice-Novice scenario showed us that two novices can gain valuable experience by working together, making sure they are confident to ask questions and not working on high risk project.
Knowing these different types of pair programming can be beneficial to you so that you make the right choice in determining who in your project is going to work together.

We hope you gained value from this presentation and encourage you to try pair programming within your organization.

(Thank you, are there any questions?)