Pair Programming Canned Scenarios

Unstructured Pairing – Programmer and Wizard agree to work on a randomly chosen project together. Who types and when is not declared in the beginning, but naturally determined between the pair.

Driver-Navigator – Programmer and Wizard are determined beforehand with the Wizard specifically driving the Programmer with Wizard driving the Programmer's actions, listening to Programmer, and guiding the flow of the project development.

Backset Navigator – Programmer and Wizard determined beforehand. Programmer is the primary driver of project development. Wizard interjects as requested or when an obvious step should take place like creating a method or correcting an error.

Tour Guide – The Programmer gives the Wizard a "tour" of the project as it is worked on with the Wizard asking questions to provoke new ideas and thoughts on how to improve the project or correct issues.

Ping-Pong Pairing – the order of who starts the following isn't required. Either the Programmer or the Wizard create a failing test and the other writes a passing test. Then they write a failing test and the other writes a passing test. This continues back and forth like a game of ping-pong.

Strong Style – Where the Programmer only does what is directed by the Wizard and nothing else.

Ideas to consider:

- Can all of the above scenarios be implemented with our application?
- What features would be required to allow each of these scenarios to be used in our application?
- Are there are other types of scenarios that can be discovered using our application?
- What are the Programmer and Wizard's skill level?
- Can we determine this from the application output?
- Do the skill levels of the Programmer and Wizard influence the above type of pair programming scenario selected?
- How can we use the output of the application to determine if any of these scenarios are more successful than others?
- Is there a way to determine if user's skills are improved?
- Can we assess satisfaction with the application from the application output?
- Can the length of time of each pair programming session tell us anything insightful?
- Should there be time limits on certain scenarios?