SPRINT RETROSPECTIVE

**By Team 1**

**Software Agility**

As any other agile software team, we too held a sprint retrospective meeting at the end of the sprint 2 after the sprint review. This meeting consisted of the Product owner, scrum master and the scrum team and no stakeholders. In this informal meeting, we have inspected and adapted the process of building the product.

**What went well?**

1. There was a positive team spirit in the team, and everyone had an effective communication among the members.
2. The sprint velocity was high since the team members worked on it with full determination and were successfully able to achieve all the requirements.
3. In our code, we have used some of the design patterns specially Observer Pattern very efficiently which has improved the performance of the product drastically.
4. The team has implemented the Model-View-Controller approach which has enabled us to achieve several software engineering principles such as high cohesion, loose coupling, parallel development of the code and separation of concerns.
5. The team has successfully handled the user-constraint which does not allow the user to make invalid connections i.e. connecting from an input dot of one shape to output dot of another.
6. Performance improved by implementing JButtons for the left panel icons and dots and adding the scroll bars in the workspace whenever there is an overflow of icons in the current workspace.

In short, it was a well-organized team with good collaboration.

**What could we have done better?**

In spite,of successfully completing the project on time and within budget and with all the features, we all concluded that some features in our project could have been improved which we planned to do in the next sprint.

1. We could have further improved the code by incorporating some more design patterns like delegate pattern.
2. There is a 300ms lag on dropping the shape from left panel to right panel which we need to minimize.
3. Instead of manual testing, we could have executed test plans and automated those test plans using a testing tool like JUnit.