Testing Approach:

Testing was done using a number of techniques including unit tests, manual validation and verification, and semi-formal inspections (using acceptance criteria checklist). Testing of a feature was done right after the implementation phase, before deploying it to the main branch, in order to find an error as soon as possible. Due to the fact that TP is an application mainly oriented to the human-user experience, most of the tests simulated user behaviour. Extensive communications were used as a measure to avoid both design, and requirements errors. Unite tests and System tests were mainly performed by developers, during the period of peer-review. The review usually was done in either form of a "Pair Programming", or by the "Walkthrough". Most of the time developers were using white-box type test, but black-box testing was also done, during key implementations, by product owner.

System Test Scenarios:

Sprint 1:

User Story: As a user I want to be able to create a new project so that I can organize my tasks into groups.

Scenario:
1. Start *Task Planner* app; select green plus sign icon; select "P" for project
     • Project Name = "Project A"
2. Press 'enter' key
3. Press 'Projects' on home page
4. "Project A" appears in list of projects

Sprint 2:

User Story: As a user I want to be able to add notes to each task so I can have a place to put other relevant information not provided by the preexisting list of attributes.

Scenario:
1. Start *Task Planner* app; select 'Dump' button; select desired Task from list
2. Locate 'Notes' icon and click adjacent to it
3. Enter notes
4. Press 'enter'
5. Notes will be saved

Sprint 3:

User Story: As a user I want to be able to "tag" my tasks so that later I can search and find all tasks with that tag.

Scenario:
1. Start *Task Planner* app; navigate to any Task list; select the Task
2. Push the drop-down icon
3. Select category

4. Push Enter
5. Tag is recorded and assists organization