Sprint Retrospective, Iteration #3

Context Project:	TSE
Group:	BlueTurtle

User Story	Task	Member responsible for the task	Task Assigned To	Estimated Effort per Task (poi	Actual effort per task (points)	Done?	Pull Request Number for the (finished)task	Notes
Visualizer: the component that is responsible for creating the visualizations.	Visualizer should be able to create a visualization of the results from different ASATs.	Tim	Boning, Tim	6	5	Yes	34	-
	Menu to enable and disable ASAT's and warning categories	Tim	Tim	7	8	Yes	34	-
	Visualization of class structure (inheritance and dependencies)	Boning	Boning, Tim	7	7	Yes	34	-
User Interface: the user interface of the system.	Basic user interface of the system	Clinton	Sunwei, Clinton	5	5	Yes	23	-
	Interface should have the option to let the user selects his/her project for analysis.	Clinton	Sunwei, Clinton	2	2	Yes	23	-
								-
Categorizing the warnings: same kinds of warnings should be put into the same category.	Write a class that can map a warning to the right category (using General Defect Classification).	Sunwei	Sunwei, Clinton	2	8	Yes	30	The plan was to write a single class, but when starting implement it, a single class is not enough, firstly, it is required to find a suitable parser for html, and we need to read the GDC information. After that, also construct example input file and try to parse it. It took some table about finding out how to parse the table as well. And it is also required to update all warning classes, with the classification derived from GDC, and update the parsers so they ouput the correct outputs.
Documentation: update documents based on changes or feedback.	Update the architecture design document if the architecture of the system has changed.	Michiel	Michiel	1				-
Analyzer: the component that is responsible for analyzing the project that is given as input.	Finish the implementation that the analyzer can run CheckStyle on the whole project instead of one file.	Michiel	Michiel	4	1	Yes	20	-
	Finish the implementation that the analyzer can run PMD on the whole project instead of one file.	Michiel	Michiel	4	1	Yes	20	-
	Implement the following feature: the analyzer should be able to run Cobertura and FindBugs.	Michiel	Michiel, Sunwei, Clinton	4	8	Yes	38	Cobertura runs but provides null output (for some reason). This is an issue but we have decided to ignore it for now as cobertura is not an important tool (it doesn't provide any data about bugs or defects) and we might end up scrapping it anyways.

Main problems encountered:

Problem 1	There were consequtively failing builds when I tested FindBugsParserTest, this is due to the slash on Microsoft Windows System is '\', but on UNIX systems the value of the slash is '/'. So when we replace the regex, some ways works locally in our Windows system, but when running Maven on travis with UNIX systems, the results are different, and to fix it, we cannot do it locally, so we have to keep test it by push on a branch and see the travis test. See pull request #35
Reaction	This problem is fixed (pull request #36)

Adjustments for the next sprint

Adjustment 1	Focus of the visualizer should be moved towards the comparison of ASATs. So in this case we will move the problem of the dependencies between packages/classes to later and focus more on the ASATs themselves.
Adjustment 2	Focus more on the treemap instead of the graph according to Moritz. This visualization provides a better overview of different ASATs.
Adjustment 3	We should take the scale of the projects that we will be analyzing into account. We only tested for small projects, but we should also test for bigger projects.

Team member	Total actual effort (points):
Boning Gong	10
Tim Buckers	10
Sunwei Wang	10
Clinton Cao	10
Michiel Doesburg	10