# User Experience Cycle 1

The UX 1 plan and results discussed here.

**Research Questions:**

This section shows the list of sub research questions considered for each main research question. In case of first main research question i.e., with displaying of results for a single project from multiple static analysis tools, the following is the sub research question.

For this part, the solution ideas i.e., single list, separate list and tags are tested about their convenience in doing regular tasks from user perspective in seeing analysis results.

1. Does separate list or single list help the user to identify the common bug?
2. Will having tags help in scalability of bugs?
3. Does the given statistics screen help the user in understating the analysis results overview?

Next, in case of second main research question i.e., with feedback while bug fixing is on-going, the following are the sub research questions.

1. Will the animation (rotation) of icons for tools suffice the feedback required by the user?
2. Will stating the progress of analysis for each tool be better than animation provided as feedback to user?
3. Does having more textual information as a popup feedback is required by the user?
4. Do users require multiple feedbacks i.e., any combination of animated icons, progress bar or pending status popup?

Finally, in case of third main research question i.e., with carrying traceability of bug fixing, the following is the sub research question.

1. Whether the given UI i.e., previous commits in process of fixing a bug finding with numbers determining the adding or removing of other bugs be able to address the scenario from user perspective?

**User Study Process and Results:**

This section explains the user scenario and questionnaire used during user study.

**Results**

**Metrics Analyzed:**

**Quantitative measurements:**

* Task Success:

Whether the user able to complete the respective task or not?

**Qualitative measurement:**

1. Usability: Ask user to provide feedback through cognitive walkthrough process about problems faced while using the prototype and get insights about the solution idea.

2. For each research question, ask users to evaluate the solution idea in terms of perceived usability on a Uni-polar Likert scale of 0-10. Scale: 0 – worst, 10 – best, in comparison to alternative solution idea provided.

**Pre-test:**

By first, the user background is verified with pre-test questionnaire whether he/she is right candidate to consider for user study. The ideal choice is the user who has Computer Science studies background and programs software projects. Also, it is found whether the user has used any static analysis tools and if so, the relationship between their favorite tool and usability is found out.

**Questionnaire:**

Q. How often do you do software development (i.e., coding)?

Q. Have you used static analysis tools?

Q. What tools have you used?

Q. Is it IDE integrated tool or any other dedicated tools like FindBugs, PMD etc.

Q. Which is your favorite one?

Q. Why it is your favorite? Any correlation to its better user interface feature?

**Results:**

There are 5 users participated for this user study phase. Everyone has Computer Science background with bachelor’s degree and in addition 3 users have sound professional experience. All 5 users are pursuing master’s degree in computer science at the time of this user study.

Out of 5 users, 3 users code almost daily and 1 user code 4 days in a week and 1 other user interest domain is designing and so labels himself as designer, thereby no coding at the moment. So, the rest of pretest questionnaire hardly applies to the designer but main part of study i.e., evaluating solution ideas from usability point of view helps better. Among remaining 4 users, 2 users are familiar with static analysis tools such as Linters like ESLint, Codacy, Better Code Hub etc. and other 2 users use manual testing but heard of static analysis in their study course curriculum. One user stated Codacy is preferred to Better Code Hub for its better recommendations in solving bugs. All users have agreed upon the importance of user interface by saying it should be easy to work on, special attention is needed for UI development etc. Interestingly, one user has stated that it happened to use two static analysis tools for some applications to have maximum coverage. He states the reasoning as a benefit in integrating two tools is that those things which are not covered by Codacy are covered by other tool. This gives an additional hope for this existence of thesis work.

Once the pretest questionnaire is done, the user is asked to assume that they are working on a software project and want to find bugs in their codebase. There are two tools linked to their codebase for better coverage of vulnerabilities. Next, walkthrough three main research questions with respect to its user interface one by one with their sub research questions and evaluate their respective solution ideas.

The solution ideas are made into prototypes i.e., wireframes using Balsamiq software tool. They are presented to user one after other in random order. Cognitive walkthrough is carried out with Think Aloud process during user study.

Now let us walkthrough each research question.

**Research Question 1:** How to display the results of the same codebase from different analysis tools?

As part of this research question, there are three sub research questions considered. The solution ideas such as single list, separate list and tags are made into respective prototypes that also includes a home screen showing statistical overview in form a pie chart graph in every prototype. The solution ideas are evaluated with following pattern with User Scenario, Tasks and Follow up questionnaire.

**User Scenario:**

The following user scenario is presented to user, “Assume you as a Software Developer working on a project and about to see the analysis results in a given user interface.”

**Tasks:**

The user is asked to perform three tasks in a presented prototype. The three tasks are about finding a common bug with tools integrated, filter the analysis results and finally fixing a bug.

**Follow Up:**

1. How do you feel about the home screen i.e., with statistics and diagrams?

2. Among the solution ideas presented, which one do you feel convenient (easy to use) with for given task?

3. Why is it the mentioned solution idea is convenient?

4. How do you rate in terms of perceived usability ranging from O be low to 10 be high for provided solution designs in comparison?

5. Do you imagine anything better UI than these? (Yes / No)

6. If yes, what does it look like?

7. Here we have seen usage of 2 tools. If the number of tools increases, which UI do you think scales better?

Let us walkthrough the three solution ideas one by one and see how the users evaluated.

**Solution Idea: ( Single List )**

As we know each static analysis tool has respective name and an icon representing it. So, when we integrate multiple tools, we either must use tool names or icons to represent the respective analysis tool while displaying the bug finding results that indicates which bug was identified by what tool. In the current scenario, an idea of having a table view with a tool column that shows tool icons for a respective bug finding. This could be well understood looking at following figure or a complete prototype images added in Appendix.

**Evaluation:**

**Quantitative Results:**

There are five user study participants. Every user has managed to perform the all three tasks stated above on the provided prototype representing ‘single list’ solution idea. This shows the task success rate is 100 percent.

Among the 5 users, 2 users felt this is convenient in comparison to other solution ideas to be considered as final solution idea in showing the analysis results. In terms of usability on scale of 0 to 10, where 0 being worst and 10 being the best, 5 users rated as 9,8,8,8,7 which averages to 8.

**Qualitative Results:**

Let us now investigate the reasoning behind the user’s choice of solution idea and respective ratings. Users felt the union of bugs from multiple tools is good, it also gives satisfaction that there are these many bugs in a codebase, easiness in filtering the results and able to go through once getting familiar, easiness in finding a common bug and overall visually appealing. As improvisation for prototype design, one user suggested to add ‘hide all’ button to disable results from all tools and ‘show all’ button for displaying complete results.

**Solution Idea: ( Separate List )**

Similar to previous one, now the idea is showing separate table view for each tool. This could be well understood looking at following figure or a complete prototype images added in Appendix.

**Evaluation:**

**Quantitative Results:**

There are five user study participants. Every user has managed to perform the all three tasks stated above on the provided prototype representing ‘single list’ solution idea. This shows the task success rate is 100 percent.

Among the 5 users, 3 users felt this is convenient in comparison to other solution ideas to be considered as final solution idea in showing the analysis results. In terms of usability on scale of 0 to 10, where 0 being worst and 10 being the best, 5 users rated as 7,8,6,9,8 which averages to 7.6.

**Qualitative Results:**

Let us now investigate the reasoning behind the user’s choice of solution idea and respective ratings. Users felt the separate lists of bugs from each respective multiple tools is that it is more effective when using more tools, filters do not make sense in this scenario and there is a suggestion of adding trending, priority as improvisation. Here, trending indicates what are the new bug findings and priority indicates the ones which are important to get fixed.

**Solution Idea: ( Tags )**

As we have seen so far using icons for representing a analysis tool. Now, having a tag name for each tool and that could be used for bug finding results displayed in a complete list view. The present solution idea is inspired from StackOverflow user interface. This could be well understood looking at following figure or a complete prototype images added in Appendix.

**Evaluation:**

**Quantitative Results:**

There are five user study participants. Every user has managed to perform the all three tasks stated above on the provided prototype representing ‘single list’ solution idea. This shows the task success rate is 100 percent.

Among the 5 users, no user felt this is convenient in comparison to other solution ideas to be considered as final solution idea in showing the analysis results. In terms of usability on scale of 0 to 10, where 0 being worst and 10 being the best, 5 users rated as 5,6,4,6,7 which averages to 5.6.

**Qualitative Results:**

Let us now investigate the reasoning behind the user’s choice of solution idea and respective ratings. Users felt the solution idea of using tool names as tags while displaying complete list of bug findings is that being the user interface little bit confusing, could not understand how tags work at first impression, could be habituated once used again and again but confusing at for first time, the UI is modern but hazy, as a viewer it looks more presentable but being a developer wishes for a easy one and less number of steps. So, overall even the idea is modern and looks presentable, users did not consider as final choice.

Solution Idea: ( Summary Screen )

As the results from different analysis tools are combined in order to show on a common interface, it is felt ideal to show a summary of results in form of graphs. In present scenario, a solution idea with pi chart with sections resembling the interested grouping / categories found in the usability evaluation report done by Johannes. These categories are found out to be the overall sections considered by users going through static analysis results. This could be well understood looking at following figure or a complete prototype images added in Appendix.

Once the solution idea is presented, the formative feedback is considered.

Result:

Overall, users have agreed upon its importance in terms of scalability issue. This means there is a need to have such summary screen which gives overview of bug findings made by multiple tools and that results in huge numbers in some cases as codebase scales up.

Users have recommended to have a transition from summary screen to list of bugs as when a particular pi / section is clicked on graph then it’s respective list of bugs are shown or at least a separate button named as ‘view bugs’. One user has suggested to use Line chart as well which shows overall performance of application by having timestamp on X – axis and number of bugs on Y – axis. In addition other user recommends to have a vulnerability scale that determines whether the risk factor is high or low. Also, one user recommended to have a numerical representation of how many bugs are fixed and how many are left.

**[ Research Question 2]**

Q. What feedback works to know that the bug fixing is on-going?

As part of this main research question, we are going to evaluate the three feedback mechanisms proposed i.e., animated icons, progress bar and pending status popup in our designed MSAT-UI ( Multiple Static Analysis Tools – User Interface ).

The ‘animated icons’ solution idea demonstrates that when a bug finding has been selected and user attempted to fix it by submitting for re-analysis, a feedback is shown with respective tool icons being animated for that bug which is under analysis. This could be well understood looking at following figure or a complete prototype images added in Appendix.

Next, ‘progress bar’ solution idea demonstrates that when a bug finding is been worked on and submitted for analysis, a feedback is given to user about how far the respective analysis tools has finished analyzing by showing progress bars for one or many tools that found the bug. This is displayed when clicked on the bug which is under analysis. In addition, the respective icons are shown as blurred indicating they are under analysis in contrast to previous solution idea. This could be well understood looking at following figure or a complete prototype images added in Appendix.

And finally the third solution idea i.e., ‘ pending status popup’ demonstrates the textual feedback about what the analysis tools are scanning as an example, tool name, timestamp, respective filename with code under analysis. This is displayed on click on text link named pending in status column in given table view. This could be well understood looking at following figure or a complete prototype images added in Appendix.

The User Scenario, Tasks and Follow up questionnaire is similar to previous research question. However, the tasks are demonstrated by the designer in order to make the feedback mechanism ideas are understood by user. The reason for not doing so is the limitation with prototype builder i.e., Balsamiq not having dynamic nature in showing animation effects and thereby designer tries to mock the animation effect with multiple clicks which could be tedious if asked the user to do.

**Evaluation:**

**Quantitative Results:**

Since the designer has performed the tasks, there is no such as task success as seen in previous research question. Among the 5 users, no user felt any solution idea is not convenient in comparison to other solution ideas to be considered as final solution idea in showing as a feedback during bug fixing process. Most users preferred them to have in combinations and especially two users felt progress bar with pending status pop up would be more useful. Nevertheless, each feedback is appreciated in context of its respective usage scenario. In terms of usability on scale of 0 to 10, where 0 being worst and 10 being the best, 5 users rated ‘animated icons’ solution idea as 7,9,5,8,7 which averages to 7.2, ‘progress bar’ solution idea as 6,9,7,9,9 which averages to 8 and finally ‘pending status pop up’ solution idea as 8,9,8,10,8 which averages to 8.6.

**Qualitative Results:**

Let us now investigate the reasoning behind the user’s choice of solution idea and respective ratings. Users felt that any animation will look good, the icons that are disabled in second solution idea i.e., ‘progress bar’ is confusing and especially the user with design interests stated that disabled icon could mean they are not clickable as a universal understanding in UI perspective. Users recommended to integrate ‘status pending pop up’ idea with others as it is more helpful.

**[ Research Question 3]**

Let’s switch to final main question;  
  
Q. How to carry traceability of bug fixing?

The following sub research question is evaluated in this context.

1. Whether the given UI i.e., previous commits in process of fixing a bug finding with numbers determining the adding or removing of other bugs be able to address the scenario from user perspective?

As part of this main research question, the solution idea of presenting the traceability scenario is being evaluated. The solution idea is been inspired from TeamScale and adapted to our context. This could be well understood looking at following figure or a complete prototype images added in Appendix . In user study, once the concept of traceability is explained to user, the following User Scenario and a follow up questionnaire is presented.

User Scenario:

The user is asked to assume he is working on fixing a bug by doing different commits for every change he did. The commits which are linked to the bug finding is listed out with numbers indicating the introduction of new bugs and fixing other bugs. This is assumed to be helpful in tracing the code changes with effect to code quality. In addition, there is a revert option to revert the codebase to that instant.

Follow up:

1. How you feel this UI will address the issue?

2. Is the provide user interface is convenient?

3. How do you rate in terms of perceived usability ranging from O be low to 10 be high for provided solution designs in comparison?

4. Do you imagine anything better UI than this? – Yes/No

5. If yes, what does it look like?

**Evaluation:**

**Quantitative Results:**

All the participants of user study has agreed upon the presented user interface address the scenario we are looking into and also it is convenient enough. Now, in terms of Usability they have rated as 7,7,8,8 and 7 which averages to 7.4.

**Qualitative Results:**

Let us now investigate the reasoning behind the user’s choice of solution idea and respective ratings. Users felt in general it is a good idea for tracing and also interface is good. The scenario being relatively new to users and also the proposed design is novel, users have suggested some improvements namely, a tool tip which explains what green and red values mean and description of bugs, a new column which shows total bugs introduced reported by all tools in combination which will be helpful in easy decision making about revert.

Post test questionnaire:

**Q.** Do you think onboard phase i.e., explain each screen when user starts up the UI will help you better understand the UI? - Is it helpful / distractive in this scenario?

Evaluation Result:

Users felt instead of welcome screens, enough text on user interface would be suffice. There is a general recommendation with having a ‘i’ kind of icon which on hover pops up with information saying about what the particular GUI element mean.

Miscellaneous

Users recommendations:

In addition to scenarios considered so far with some suggestions as scope of improvement. There are few others that could be looked into as well namely, users suggests drop down option for filters, collapse option would scale better in UI when more tools are integrated, some developers might want to see the bugs with different status as example fixed, pending, not fixed etc. , having a search for tool selection instead of long filter list will be more user friendly, consider renaming ‘Fix Now’ to ‘Submit’ which is shown in the prototypes, consider to add ‘stop analysis’, add a ‘select’ and ‘deselect all’ option for tools in filter list, it is not useful in having bug description again while on code editor screen, the colours used in design should be common. Some technical feedback and suggestions from users are that they felt struggle usually in configuring multiple tools and so every tool should come up with easy installation process integrated with IDE. One user recommends to go with 2 or 3 tools in general and the selection of tools should be done prior and integrate with IDE.

Limitations:

The limitations observed during this first UX design cycle is having limited mockup screens sometimes surprises when interface does not respond while clicking on some other GUI elements such as buttons or other links on interface which are out of context from designer perspective. The other limitation that comes with chosen prototype tool builder i.e., Balsamiq is that when user clicked on keyboard input or with random mouse clicks, mock up screens are jumped.