Blood, sweat and tears: a failed project and lessons learned

By Ruby Tomar

In a complex global organization initiatives are taken so that development of products/solutions gets easier, better and optimized. Such initiatives are not main stream projects because they do not contribute to the core of the company. These are typically taken up in the free bandwidth of engineers so that the development of the core is not disturbed. Failure rate of such "side" projects is high as such projects do not get the time and resources required for successful completion. This case study summarizes a real project which failed to deploy and highlights the lessons learned along the way.

Problem and Motivation

User Interface (UI) is the face of any electronic device. To serve a global consumer base, the strings in the UI are localized in different languages. Testing whether the strings display correctly in all the supported languages is a humongous task.

- Every screen needs to be generated in all supported languages. Our devices support 26 languages; average time spent to generate the screens per language is 140 hours. Therefore, the total time required just to generate strings in all languages is 3640 hours (26x140) per device.
- All strings in each screen need to be validated for errors and truncations. This is a very taxing and time consuming activity as in most cases the testers do not understand the language being tested. Testing involves manually comparing what is shown in the UI with the "expected strings" document. The comparison is manual and is prone to errors. The problem is aggravated if the testers do not understand the language which they are testing.

Our solution

The UI technology was being developed by another team. We wanted to create a tool to automate the localization testing of the UI which would help in saving time and resources. We took this up as an initiative or "side" project. The project involved generating snapshots of screens in the target in all the supported languages, transferring them to the host PC, extracting the text from the images using an OCR tool, comparing the extracted (actual) text with the "expected" text, and reporting the errors.

The team gathered some preliminary data from the UI team and started the implementation. Generating error condition screens from the device was tricky; however, in some cases it was possible to fake the system to generate the error screens. Our device has 5 flows of execution. Execution of 1 out of 5 flows was completed and successfully demonstrated to the prospective users of the tool. We had achieved a 600 times improvement in the execution time.

Automating the remaining 4 flows could not be completed on time due to the factors shown in the *Figure 1: Fishbone* diagram. The team remained stuck with several roadblocks, the solution for those was not very upfront and clear. The project was put on hold for some time and the resources had to be put into another project. Eventually, the R&D cycle for the current set of products was completed and the products got tested in the conventional way and released to the market.

For the next cycle, the UI technology evolved and was replaced by newer technology. The project was restarted to work with the newer UI. The team did not have a clear understanding of the new implementation so the same problems continued. Meanwhile, the UI team started developing another tool for automating localization testing, with a slightly different approach as shown in *Figure 2: Comparison of* the two approaches. In our approach we were testing *WYSIWUG*, whereas in the newer approach the data was captured before it is actually displayed on the UI.

Even with this merit, our project failed to deploy as we could not meet the deployment date.

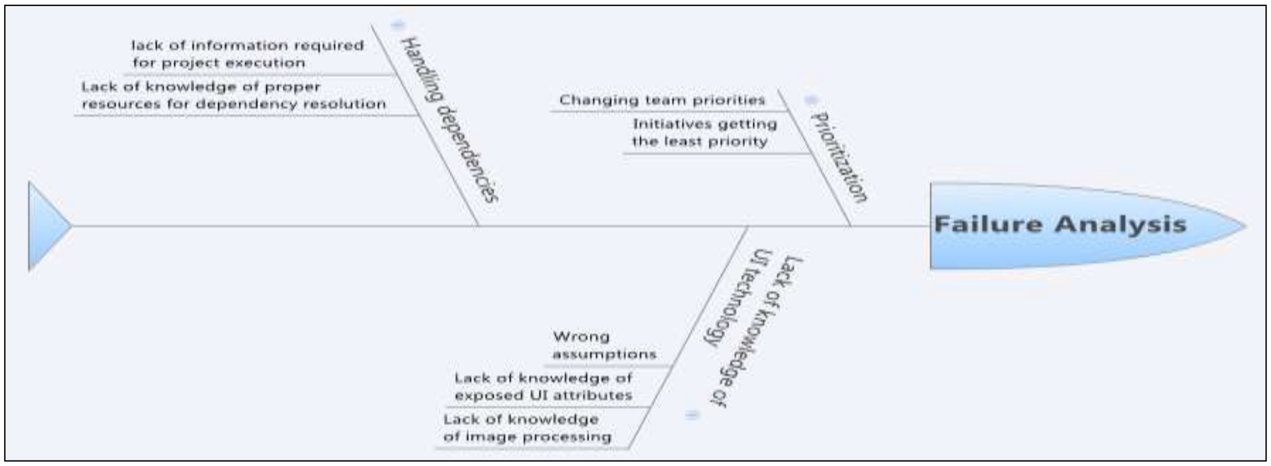
lack of information required for project execution
Lack of knowledge of proper resources for dependency resolution
Handling dependencies
Changing team priorities
Initiatives getting the least priority
Prioritization
Failure Analysis
Wrong assumptions
Lack of knowledge of exposed UI attributes
Lack of knowledge of image processing
Lack of knowledge of UI technology

Figure 1: Fishbone diagram

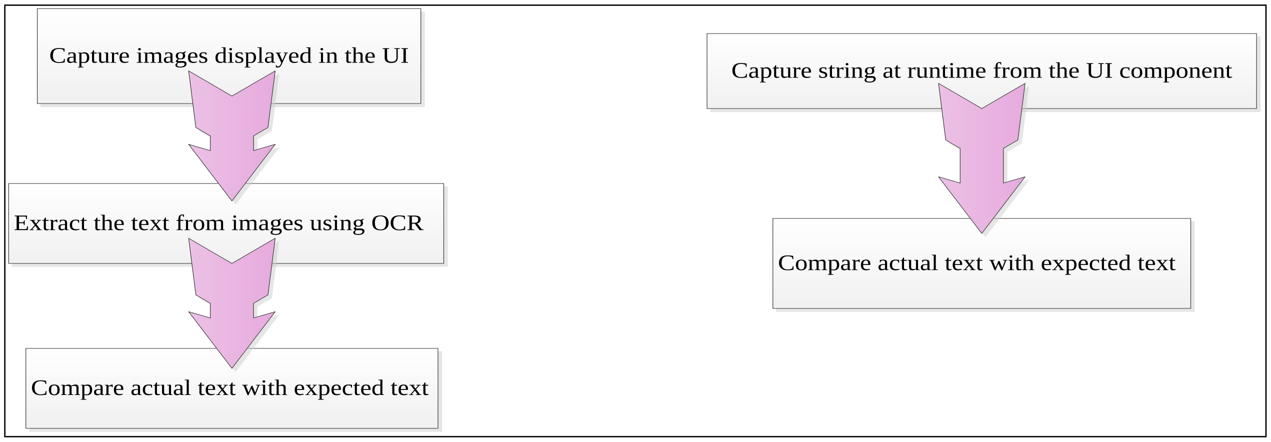
Capture images displayed in the UI
Extract the text from images using OCR
Compare actual text with expected text
Capture string at runtime from the UI component
Compare actual text with expected text

Figure 2: Comparison of the two approaches

Results and Contributions

What went wrong	Lessons Learned
Plan did not consider lack of domain knowledge of the underlying UI technology and image processing	Proper analysis of the project prior to execution: • Capitalizing the strengths / capabilities • Shoring up the weaknesses • Identifying and mitigating the risks will help in better planning. Mind mapping tools and SWOT analysis tools can be used.
Project execution was a challenge because of changing engineering priorities, and initiatives ("side" projects) getting least priority	Secure management backing for prioritizing Tools like Kanban help engineers in making work visible, controlling the work in progress, thereby improving prioritization
Lack of formal dependency management	Tools like Kanban make work visible and help focus the teams attention to the blocked/dependent issues

The team has benefitted from the enriching learning experiences of this project. I would like to thank Praveen Veerappa for all his contributions towards implementing the project.

About the Author

Ruby Tomar

India

Ruby Tomar is an action oriented, decisive and results focused Program and Project Manager with 16 years of experience in the IT systems. With three patents filed and eight disclosures to her credit, Ruby is process and technology savvy with a strong inclination towards innovation and process optimization. She has worked in automotive, consumer, networking, and telecommunications industries and is an avid reader of technical and management research. She has an MS degree in Software Systems from BITS, India and is currently working as a Program Manager at HP. She can be reached at rubytomar.sps@gmail.com.

About the Author