



Iteration 2 Report

Steve Thorpe

Contents

1. Introduction.....	0
2. JavaFx 2.2 to JavaFX8 Review	1
3. Completed User Stories Review	1
4. Project Wide Standards Review	2
5. Contracted Handlers Review	3
6. Test Driven Development Review	4
7. Pair Programming Review	4
8. Coding Standards Review	5
9. Testing Standards Review.....	5
10. Integration Process Review	6
11. Jira Practices Review	7
12. Version Management Review	7
13. Final Product Review.....	8
14. Conclusion	9

1. Introduction

This iteration report covers the time between the hand in of the first Iteration on Friday Spring Week 10, to the final hand in Thursday Summer Week 8. It is intended as both a critical review of the product and the software practices used to produce it. It will contain:

- Discussion of change from JavaFX 2.2 to JavaFX8
- The user stories covered and how successfully they were integrated within eCook
- Analysis of how eCook meets the PWS (Project Wide Standards)
- A review of the contracted modules and integration of them within eCook
- Review of Test Driven Development
- Review of Pair Programming
- Review of Coding Standards
- Review of Test Standards
- A review of JIRA practices
- A review of version management using GitHub.
- Software managers review of the product

2. JavaFx 2.2 to JavaFX8 Review

Over the Easter break I made the decision to change the Java frame work again from JavaFX 2.2 to JavaFX8. This was done firstly to allow us to use new classes and methods that have been included within JavaFX8, which solved problems that would have been difficult to solve using JavaFX 2.2. JavaFX8 operates almost identically to JavaFX 2.2 and switching to it did not cause any significant problems other than having to download the latest Java JRE8. Ultimately this change should extend the working lifetime of eCook further as it now operates using the very latest version of Java.

3. Completed User Stories Review

Iteration 2 covered sprints 3, 4, 5 which were detailed in the Sprint Plan document and the final sprint 6 which was intended to be only used for bug fixes and general polishing of the product. In the first Iteration report I stated that the deadline for sprint 5 was to be the end of Summer Week 3. This time scale was not kept to due to additional user stories that were added to sprints along with almost two weeks lost to the exams as nearly all group members devoted their time to revision. Adding these users stories and extending the sprint times has added significant value to the product.

Out of all of the user stories within the sprint plan only one was not attempted which was "Share links to recipes over social networking". This was removed from the sprint plan due to other user stories which we developed while writing the program which addressed problems with eCook that had not been anticipated within the original design. The additional user stories were:

- Be able to pre fetch media which is available via HTTP in a recipe playlist
- Create an XML validator
- Scale slideshows to the computers resolution
- Produce the GUI
- Produce a logger and error handler

Pre fetching the media directly to the computer before displaying the slideshow is intended to speed up the performance of eCook when a recipe contains many instances of media content. Fetching the media can happen in parallel with using eCook and the software team felt this was an important feature to improve the users experience with eCook.

The XML validator was intended first to ensure that any XML playlist that is loaded into eCook would be compatible with it. Future versions of this feature are intended to be distributed independently of eCook to allow chefs to write their own recipes for distribution within the eCook online store.

Scaling slideshows is necessary as the positions of all content in XML playlists are defined in pixels. If the XML resolution does not match the screen resolution it is being used on, content may not appear correctly on the slide. This user story corrects these discrepancies.

During the design process we decided to let the GUI be driven by the features without any formal design, relying on only mock-ups of the GUI produced by Sam as a guide. As the project progressed it became clear that trying to have the GUI evolve out of the coding process without a formal direction was impractical and unlikely to produce the results we desired. Jon and I produced a new GUI design which was based on achieving a specific workflow within eCook leaving the details of the GUI content up to the programmer assigned to it. This method was also ineffective; the GUI was redesigned at least three times with constant tweaks meaning that at times that GUI was changing hourly.

A waterfall method of GUI design would have been more efficient, after agreeing the workflow we should have had new designs for the GUI drawn up and agreed to them before writing any more GUI code and a direct user story to design and produce the GUI should always be a priority on any future project.

The error handler is to provide information to the user in a controlled way that some kind of error has occurred within eCook followed by the action it will be taking, usually exiting the program. The logger is to provide a comprehensive record of all the actions undertaken by eCook while running which can be used for debugging and user for providing user feedback of errors once the program has been released.

4. Project Wide Standards Review

Meeting the PWS has always been a priority within eCook. As with any standard the PWS process was rife with disagreements and I definitely do not consider it to be an optimum solution as I believe it tries to emulate the function of Microsoft PowerPoint too closely. The PWS that was agreed however does provide all of the functionality required for a functioning

slideshow containing the five different types of media so in that regard I believe the PWS process was a success.

Once the PWS had been agreed I wanted to keep any changes to an absolute minimum. The only change we requested as a group was to define that the slides display order must be given consecutively in the XML playlist as this was not explicitly defined prior to our suggestion.

Changes to the PWS throughout the project have required the content handlers to be redesigned on at least two occasions but the result is a product that can accept a playlist containing all of the elements and attributes defined for a PWS playlist.

The only exception to this is the YEnd value for the Text Handler. Our implementation of the Text Handler uses the Text Flow class from JavaFX 8 which will resize itself to be larger than its defined size (which is defined by the XStart Xend YStart and YEnd values) if the input text string is larger than the box. This effectively renders the YEnd value irrelevant as the Text Flow will ignore it if the string is too large.

I believe that this is a better implementation however as I believe that the user would want all of the text that has been entered into the text string to be displayed and would not want it chopped off because the size of the box containing the text is too small.

5. Contracted Handlers Review

Both of the handlers we received (the Audio and Graphics Handlers) were fully working handlers in their own right but did not meet our design aim of having entirely encapsulated handlers which required a single constructor to display content. This later became less important however as we were later forced to break the encapsulation of our handlers in order to meet all PWS requirements

All of the handlers were refactored however to avoid the significant amount of repeated code which was present in the design of all of the handlers by using abstract classes and inheritance to provide all of the methods common to the handlers. Both of the contracted handlers were integrated successfully by the software team after modifications to make them compatible with eCook's workflow.

6. Test Driven Development Review

Our QA manual states that where possible and appropriate we would use TDD to develop eCook. TDD was very successful for some aspects of project, particularly classes which involve large amounts of data manipulation in sequential programming context.

Problems however were encountered when trying to use TDD to test multithreaded code or testing GUI interfaces. The test frame we used (JUnit 4.0) is not intended to test multithreaded code or GUI elements, which resulted in difficulty when attempting to write tests for timed elements or GUI elements more complicated than using a button.

If attempting on another project I would encourage the use of a mocking framework which would allow us to more easily test the principles of the code using mock objects as opposed to trying to use actual class implementation but more importantly we should consider the objective of test driven development; that unit tests should be carried out in isolation of other code and that running tests of multiple threaded code breaks this premise.

7. Pair Programming Review

All programming within the company was initially supposed to be done in pairs, however the logistics of programming over the Easter break and during exam period meant this was not always possible.

For the first 2 sprints the pairs were changed around each time, the intention being that everyone can learn from other people's strengths, it did however become apparent that some members of the group were significantly stronger programmers than others and the decision was made to fix the pairs from iteration 3 to ensure that two people who are not as confident with Java were not paired together.

The danger of this arrangement would be that the weaker members would effectively become silent partners with all of the coding being performed by the stronger programmer of the pair but this was not quite the case, all group members have contributed towards the code in a meaningful way, but some programmers have contributed significantly more than others.

In regards to the effectiveness of programming in terms of the quality of code produced, there were examples of both good and bad code being produced by pairs in similar ratios to code produced by programmers operating on their own. Pair programming has a better opportunity to raise

the overall programming standard within the company and should be used where practical.

8. Coding Standards Review

The coding standards document was intended to provide rules for writing code in eCook which did not interfere with a programmers natural work flow, in that aim I believe that were successful, many of the conventions within the document have been used by all members of the group when programming at the university and after the second iteration were well followed within all code produced.

There were only two changes to the coding standards during the project. I decided to remove the version control within the headers. Many classes underwent so many revisions so quickly that the revision history would have been very large, which would have just put people off from reading it. I also believed that as we had much more comprehensive version control system in the form of GitHub which can retrieve the state of any class from any commit, that a written form of version control is unnecessary as a comprehensive history can be viewed from there.

The second change was to include @Param and @Return comments into the method comments. This is to give a clearer description of the arguments a method requires and the code it gives and also allows this information to be viewed in eclipse using the ctrl space auto code function.

The software team initially began the project by having formal meetings to review for coding standards but these quickly became infrequent as we felt that our time was better spent improving the code itself rather than checking regularly for coding standards. Instead coding standard reviews were conducted as deadlines approached, only in rare cases was there a significant amount of work to do to bring a piece of code up to coding standards therefore I believe this method was a success.

9. Testing Standards Review

The Testing Standards Document was only amended to remove the version control requirement for the same reasons as the coding standards document. The biggest issue with testing was test classes not being changed as code was refactored or altered. This led to a significant number of classes

failing their unit tests, when they in fact worked fine the tests were either calling out of date XML test files or checking for refactored variable names. All of the tests were refactored but such a large period of time between the class and the test class being changed should not have been allowed to pass.

The Deterministic Test Plan proved to be an effective idea, the tests within it were detailed and comprehensive however the DTP sometimes did not cover all of the features currently within eCook, there were a few instances where code was integrated without a DTP being submitted to the appropriate JIRA ticket which contributed to program breaking bugs within eCook due to missed null pointer exceptions or faulty logic. In almost all instances these were only minor errors which were easily fixed by either the programmer who made the commit or another member of the team and at no point was the code ever reverted to a previous commit due to a testing failure but these were still lapses in company policy which should not have happened.

10. Integration Process Review

The first classes produced in the project were integrated into eCook by the software team which was conducted successfully and was the agreed strategy by the entire group. The intention was then for the other members of the group to integrate user stories they were assigned by themselves with the software team to be contacted for advice or if there was a problem with the integration.

Integration did not proceed in this manner during the Easter break however; the Development Manager produced the Integration Standards Document which gave a guide to integration but as each user story was different from the other it was impossible to produce a comprehensive document for every integration. The management team suggested having weekly lab sessions with members of the software team in attendance to provide direct support to the other group members, which proved successful and vastly increased the progress on the project.

Towards the end of the project the integration process was conducted well by the majority of the group with multiple members of the group being able to make successful changes to the code at the same time.

11. Jira Practices Review

JIRA was intended to streamline the tracking of work on the project and to remove the need to provide separate documentation for each of the classes produced. The company's use of JIRA was defined in the QA manual and all programmers were expected to conform to these standards.

Our use of JIRA for documenting the code was successful; all the user stories that were produced had a JIRA ticket assigned to them which contains a description of how it was implemented and the DTP (Deterministic Test Procedure) result. JIRA was also very useful for determining the work that had been completed at the end of each sprint and which user stories needed further work.

What was less successful was tracking the work done in real time; the main selling point of JIRA is that the board is a reflection of the work completed or being worked on. This only works however if JIRA is updated regularly and for the majority of the project this did not happen. Towards the end of the project (during iteration 6 especially) updates were made to JIRA much more regularly and the data on JIRA reflected the work done more accurately. I believe that using JIRA was a good experience for the group, there are many software companies that use it and similar technologies to track their workflow which meets the aim of the SWEng project to "gain experience in the procedures... through applying established industry practices".

12. Version Management Review

GitHub desktop was the primary method of accessing eCook, which is widely used for group software projects. While it is a useful tool our groups experience with it (including my own) has caused a wide variety of problems and errors throughout the project due to incorrectly committing files.

Attempts were repeatedly made by myself, the development manager Jon and the management team to urge everyone not to commit files to GitHub which would cause problems, such as the .classpath files which were intended to be relatively so that eCook could be compiled on any computer but were repeatedly set by everyone in the group at some point to their own machines class path which would propagate to the other group member's computers and break eCook.

This problem is mitigated through the use of branching within GitHub, which requires each programmer to take a branch of the code, produce some work and then merge their branch back into the master branch. However as using GitHub in a project of this scope was a new experience for almost all the members we decided not to use this feature as the merging of branches together can become very complicated and we feared that the problems caused by this would be more significant than simply committing the wrong files.

The group's confidence and knowledge of using GitHub has increased substantially throughout the weeks of the project and were it to continue further the software team would begin to encourage the use of branches and merging to group members which would hopefully mitigate the majority of the problems that were encountered while using GitHub. Another improvement to be made is to standardise commit messages in GitHub as the quality and quantity of the information provided in each commit varied greatly from commit to commit.

13. Final Product Review

eCook has all the features that were to be provided in the second iteration, with the exception of the Sharing over social networking user story, and all work to a good standard. The product can use a variety of playlists, which are tested by eCook before being used to ensure that they are valid, which can contain all of the require media types and is fully controllable by the user.

The program is highly robust; our testing of the program has found that it is difficult to crash or subvert with many of the problems eCook currently has being only superficial in nature and any errors that do occur can be logged for further review and can be handled cleanly by the error handler class which informs the user that there has been a problem.

Attention has been given not only to the features of eCook but also to its system performance; taking care to tidy up threads or processes that do not end automatically and providing features to allow the user to copy all content to their hard drive to prevent eCook being limited by the user's internet connection.

The final GUI I believe is quite attractive and has a strong brand identity that firmly asserts that this is a program about food and cooking.

One feature to be improved is the GUI workflow. When designing the workflow, the Development Manager and I came to the conclusion that the GUI workflow for this iteration should be very simple and compartmentalised to make it easier to integrate the code designed to meet the user stories within the GUI.

While this method was successful in producing a GUI that works successfully and with few bugs, it does mean that the interface is somewhat clunky. Reaching another section of the program is done entirely through the main menu, which for example means to if you are looking at the list of recipes in the Recipe Menu and wish to add another one, you must go back to the Main Menu, into the Import Recipes Menu, import the recipe, go back to the Main Menu again and then back in the Recipe Menu to see if the recipe has been added correctly. While each step in this example does have a clear role and functions correctly, it would be much more intuitive to be able to go from the Recipes Menu straight to the Import Recipes Menu without having to go by the Main Menu.

This is a compromise that was made in the design process; now all we know all of the features can operate correctly together we can begin streamlining the work flow in further iterations, which is a process many software products go through their entire product lifetimes.

Future iterations would seek to added further features to the product such as: the Sharing via Social Networks user story; other nutritional information such as a calorie counter or the estimated amount of fat, protein and carbohydrates; and move from a windows only platform to the Android and iOS operating systems.

14. Conclusion

In summary I believe that the design practices we have used throughout eCook have been successful. I have detailed changes that could be made such as better practice when using JIRA and GitHub but I believed that a lot has been learnt from the project experience and that it has definitely met the aim of using techniques and methods from industry within the software project.

