# Elaboration Phase Status Assessment

## 1: Risks and Issues

The most significant risk faced was the risk of there being an inability to allow customized changes to the Error Form. Our response strategy was to communicate with the team if any issues arose while exploring the prerequisites for implementing the functionality. Our contingency plan if issues did arise was to strip back or completely remove the form editing functionality from the project. In the end, issues did arise, as we were unable to design a database to be able to be changed ad-hoc while still being able to be useful, and we had to fall back on our contingency. The functionality is now limited to enabling and disabling the provided fields – which will allow users to gather the most useful data while minimizing error entry time. The contingency plan was used, and this risk's status is Closed.

The next most significant risk was that of not being able to test development code on a local machine without having to start or restart the remote server that is the (developmental) production environment. Our response strategy was to have a team member assigned to write a guide to set up a testing server hosted on their own computers. Our contingency plan was to have that team member walk the team through the process during a team meeting. The response strategy was implemented and each team member was able to successfully run the server program on their local computer using the guide. The response strategy was successful, and this risk's status is Closed.

The most significant active risk is the loss of knowledge should a team member leave the project. Our response strategy is to share knowledge gained on completing tasks with other team members, and ensuring reporting is open and honest. Our contingency plan is to ensure knowledge is being shared continually throughout the project, and to support team members struggling to meet required expectations, without completing their workload for them. We are currently posting tasks completed in our push communication mechanisms, and explanations of work done are given when requested. However, this risk is still active, as one team member currently has most of the control (logins, passwords, experience with interacting) over the external server that is our production environment.

An issue that came up during the Elaboration Phase was external commitments, specifically exams for other subjects towards the end of the session, reducing the available time for work on the project as well as overworking those affected team members. We worked around this by having other team members pick up the slack once this issue was raised in the push communications. This issue is ongoing, in the sense that it predictably will arise again towards the delivery date of the project, but it should be easier to mitigate then.

An earlier issue was the inability to implement the "Edit Error Submission form" the way we had envisioned. We identified this as a risk and dealt with it as one (as detailed above). This is not an ongoing issue.

## 2. Executable Architecture

Our CCRD use cases were "Log In", "Add an Error", and "Produce Graphs/Charts/Visualisations of Errors". These use cases were chosen because they made up the core idea of the program and excersied the most important aspects of the architecture.

The Executable Architecture exercises the front-end data entry aspect, the back-end link to a database, and Metabase for the creation of graphs, charts, and visualisations. The aspect of the architecture that has been left out is the ability to send reports to contacts. This was left out as it was not a critical use case and it was not part of the core function of our program, and as such it was not chosen as a CCRD use case.

End-to-end production level support for the CCRD use cases, using the chosen architecture in the intended production environment, was successfully achieved. All parts of the normal flow and all alternate flows for the CCRD use cases were implemented. All architectural elements to support the CCRD use cases were implemented. The app was built and deployed on a basic Amazon web server, which is accessible from internet-enabled devices – This is the intended production environment, but better servers may be required during actual deployment. The only issues in deploying the application amounted to last-minute bugfixes.

## 3. Final project plan

The plan has been modified – the order of some use cases have been shuffled around so that work during each iteration is on one aspect of the program functionality. Following the reduction of the "Edit Error Submission Form" use case and it's reduction in scope and risk (as described in 1: Risks and Issues), that use case was moved from the Elaboration Phase to the Construction Phase. Some functionality has been dropped from the Edit Error Submission Form use case, as the use case in it's initial form was too risky to implement.

The specific project related elements specified in the final project plan add up to a completed project, as set out in the project Vision Document.

## 4. Testing

Our Master Test Plan identifies the major features and capabilites we want to make sure are working properly in the final product. Those features are:

• Error data entered is saved and stored in a database.

• Updating an error record will update the correct record in the database.

• Searches/filtering of the database will return all records that relate to that specific search/filter.

• When a graph, chart, or visualisation is requested, all data relating to that specific visualisation is extracted from the database to ensure a meaningful visualisation is created. The visualisation should be downloadable to the local machine on which the application is running.

• When extracting data to create an Excel spreadsheet report, all appropriate data relating to the report is included. The report is to be downloadable to the local machine on which the application is running.

• When a report is to be sent to a contact or contact list, an appropriate email application is opened with the report added as attachments to the message of the email. A subject line is to be added to the email giving the report name.

• When a new contact is added to the contacts, their data is saved to the correct table within the database and it is possible to send emails to the contact.

• Various contacts can be grouped together to form contact lists to enable reports to be sent to a contact list rather than individuals if required.

• Updating of contact information saves the new information to the correct record in the database and is recalled when a new email is sent to that updated contact.

• Deleting a contact from the contact list will make the record unavailable to authorised users.

• A newly added user can only perform tasks as outlined by the authorisation level given to the user (user, supervisor, administrator).

• When a user logs into the application, they will see the options available to them as per their authorisation level.

• When a user changes their password, they can login to application at their authorised level.

• An update of user preferences ensures user can only perform tasks outlined by the authorisation level given to the user (user, supervisor, administrator).

• A deleted user will not have access to the application. However, user details are to remain in the database for auditing purposes and to appear on reports as required.

• A form that is customised will still enable data to be saved to appropriate database tables, enabling data to be extracted in same method used prior to changes to form being made.

User Acceptance Tests will be carried out to test all of the above core functionality of the program. Unit and integration tests will be created where appropriate to ensure coding standards and functionality.  
  
As of the conclusion of the Elaboration phase, UATs for the CCRD use cases have been partially completed. Some UAT was done by Jette for logging in and by Beau for producing graphs, charts, and visualizations of data. The completed tests were successful, however not all tests were run due to time constraints. The incomplete tests will be run before the next oversight meeting. However, informal testing shows that all CCRD use cases are complete and functional.

## 5. Team Status

Everyone is doing their assigned work items on time and communicating with the team well. However, there are some concerns that Ryan in particular may not be taking on a similar work load to the other team members. We are continuing to monitor the distribution of work. Beau's work may not be as visible in the commit history, but his work has been mostly related to administering the completed code in the production environment, and we have no concerns that he is not pulling his weight.

We can forsee a similar issue regarding team member commitment towards the end of the Transitition Phase when team members have exams for other subjects, as had happened at the end of the Elaboration Phase. However, we plan to have less work to do during that time, and we will distribute more of the remaining work to less affected team members to make up for it.

## 6. Miscellaneous concerns

Initially the program was going to consist of multiple platform targets. There was going to be a backend server, web client, Android client and iOS client. However, due to the factor that none of the team members had any experience in any of the widely available cross-platform architectures it was decided to instead go for a purely web-based approach. By using a design framework that is responsive to mobile phones and desktop web browsers it is possible to make a user experience that feels seamless regardless of device.

All documentation was revised in order to show the changes in our project. For instance, originally the team was going to make the error form customizable. This would allow users to change the error form to have new fields or remove old ones. It was deemed too hard in the amount of time allocated, so the scope was reduced. Instead of the addition of fields, users can only toggle fields off they don't wish to use.

Team Pharmacon has an actual client. A main concern is if the client's needs have been met. If the clients needs haven't been met then the project is technically a failure. Once the LCAM milestone has been reached the client will be approached and the prototype given to them for a test run. Feedback will be gathered from the user and the scope will be adjusted next semester to match the clients concerns.