## Executive Summary

This document outlines some of the considerations associated with developing a source code, bug tracking, and software design collaboration tool. Although individually these tools have existed independently of each other, it has been decided that I would be best to incorporate these tools into a ‘super collaboration’ tool, giving all the power of source control, mixed with bug tracking and reporting; while allowing communications between the team to be tracked during the software development process.  
One such software comes close in providing this service, but with the ready availability of multiple tools are open-source or freeware, so why not apply a mix readily existing tools, providing the development community at large, with the tools necessary to complete complex design tasks with minimal concern.

## Product Services

The problem with a product such as this is that, a single all in one solution isn’t readily available on the market, software such as ‘Bugzilla’, is able to accurately keep track of bugs, whereas software such as ‘Gource’ is excellent visual tool for seeing member contributions to software repositories. But none of these packages do everything, ‘Unfuddle’ probably being the closes

### Gource

One aspect of the software that’s being commissioned is being able to graphically visualise the changes being performed to the repository, by tracking the logs (through trunks, commits and checkouts) (acaudwell, 2013). Gource, another open source project provides an amazing display to visualise the data as it makes changes- of course this software is inherently limited since it provides no real other features related to this specification.

### Bugzilla

According to the Bugzilla website, this open source software tracks bugs and code changes, allows you to communicate with teammates and submit/review patches (Bugzilla.org, 2012). It markets itself as being able to “reduce downtime, increase productivity, raise customer satisfaction, and improve communication”.

### Fossil

Fossil (Fossil, 2014) is probably the closest to the sort of software that we’re after. It provides a web-based interface with graphical tools, bug tracking and support for creating your own wiki. The main issues with using this software is that email control, since the requirements specify importing project emails into the source control, although the use of the Wiki could work around this.

### Unfuddle

Unfuddle appears to fill most of the requirements, allowing bug tracking and management of source code, this software can also allow collaboration through a messaging service and wiki. The major downside of this software is that unlike all the previous software, its not freeware, and requires a minimum $15 a month (Unfuddle).

## Technology Considerations

## Product/Service Marketplace

The only reputable competition to this software came from Unfuddle, and with their signup fees, would be a major upset, especially for the casual developer who prefers to use free and easy tools.  
All the other tools failed to capture all the necessity, and it should be noted that even Unfuddle still doesn’t quite do what’s required, its just happens to be the closest thing.  
This sort of software, due to its cost, would be targeted at developers and businesses that can both afford and are large enough to require using a comprehensive source control strategy. Smaller developers have to rely on existing open source and freeware tools, in order to collaborate and create their products, its here that exists a niche market for the serious but small developer.

## Marketing Strategy

Since this software is more likely to be used in-house, there doesn’t really require marketing a program like this. However, if this program is to be released externally, then it should be released as free, allowing for all users to popularise its use and become a staple of the software development community. In order to generate a profit (or at lease break even), advertising space can be bought to complement this software and pay for its development. As well as this, for large organisations a premium service can be offered that gives companies extra support and training for its staff, for a nominal fee, and as such generating a profit.

## Organisation and Staffing

## Schedule

## Financial Projections

At this stage there isn’t a real projection of financial income due to the nature of this software. If the above suggestions are taken on-board, that is, use the software to advertise and offer training programmes, it can be assumed that money can be made. However this hinges on the fact that this software must become popular to start with, and that is why it will be released as freeware.

## Findings and Recommendations

It is our recommendation that software be immediately be commissioned to fill a need, that collaboration isn’t cost effectively managed by any existing software. The conclusion being that a better and more tailored piece of software can be developed, that meets the requirements specified for this project.

# Works Cited

Unfuddle. (n.d.). *Unfuddle Signup.* Retrieved 03 26, 2014, from https://unfuddle.com/signup

acaudwell. (2013, 04 26). *Gource-Software Version Control Visualisation.* Retrieved 03 26, 2014, from https://code.google.com/p/gource/

Bugzilla.org. (2012, 11 19). Retrieved 03 26, 2014, from http://www.bugzilla.org/about/

Fossil. (2014, 03 15). *Fossil.* Retrieved 03 26, 2014, from https://www.fossil-scm.org/index.html/doc/tip/www/index.wiki