Durham University
School of Engineering and Computing Sciences

Whitespace

Software Engineering

Durham Software Wind Tunnel

Requirements Document

SEG 2

Prepared By:

Adam White
Ed Henighan
Fergus Marr
Hamish Robertson
Nicholas Sellen

Table of Contents

1. Purpose

1.1. Executive Summary

The Engineering Department of Durham University (subsequently referred to as *the client*) has a number of wind tunnels. Data from the wind tunnels is logged and subsequently analysed by bespoke software. This software, called Durham Software for Wind Tunnels (DSW), has been written by professors and students of the department over the last ten years. Current users have varying levels of experience with this software.

DSW is a suite of Windows executable files that are run from a command-line, most of which require several arguments to be specified in a text 'control' file. These control files can be large and complex; getting the syntax right is often challenging and time consuming, especially for novice users.

Our brief is to engineer a graphical interface that facilitates the simple and intuitive creation of these control files and the execution of the individual programs. Our solution, subsequently called *Air*, will identify the formats of the parameters required by each program, providing assistance and input restriction to reduce the number of errors made by the user and make the whole process of control file creation easier.

This document aims to detail the specific functions required by the client and how we analysed the problem domain to identify these requirements.

2. Domain Analysis

2.1. Analysis Process

The primary analysis methods that were used to gain an insight into the domain for the project were:

- **Practical Demonstration of current system:** This involved the client showing simple examples of the system's typical uses and outlining certain flaws within the system in use.
- **Project Brief (Document):** This is the original specification given to us by the client, which covers their basic expectations for the system, further desirable features, and serves to outline both the current state of the Durham Software for Wind Tunnels (DSW) Suite and the problems with its use.
- **Tour of current facility where the system is used:** This showed us the actual environment in which the system is used.
- **Interview Q&A session with current users:** These users with varying skill and experience levels of using the software explained their issues with the system and enabled us to prioritise several key issues.

2.2. Analysis

From analysis undertaken by our team members we gained a deeper understanding of the problem domain. The current system consists of a suite of around 80 programs which are written in-house in the programming language C++. These programs are split up into two main categories, data-logging and data analysis.

These programs are currently run from the command-line by many users of varying skill levels. Each of the wind tunnels has a PC connected to it on which the calibration and data logging programs are run. The wind tunnels are located in several different rooms in the Engineering Department. After the data has been logged the analysis programs can be run on the data from any ITS networked machine.

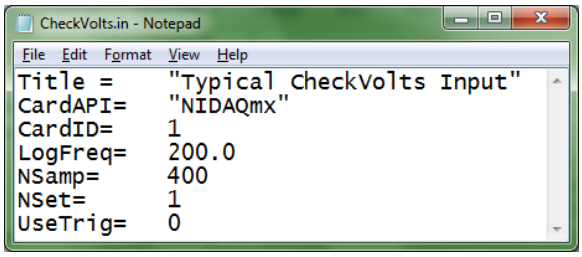
The output of the logging programs and the input and output of the analysis programs are Tecplot files (.tec), which use the ASCII character-encoding. Tecplot is a commercial application for the visualisation of data. Its use is not required and is outside the scope of this project.

In addition to these data files, certain programs also require settings to be specified in 'control' files, which have the file extension '.in' and are also ASCII. The format of these control files is very strict and slight mistakes, like a simple carriage return in the wrong place, can cause dramatic failures; this can be very frustrating as users can struggle to find an errant character in their file.

These input and output files are passed to the programs by command-line arguments.

The output (.tec) files of the analysis programs usually contain the data columns from the input files with an additional column for the newly calculated results. Some programs perform further analysis on the results of other programs and so require those columns to be present in the input file.

Example Control (.in) File (For CheckVolts program):



Some of the programs in use require basic user input after execution of a program – 'basic' meaning to the confirmation of readings and calibration of equipment. This is done through the command-line.

The majority of program-related files are located on a network drive (storage that is not on the PC); this has caused a few issues with the current users as the network is not regularly accessible.

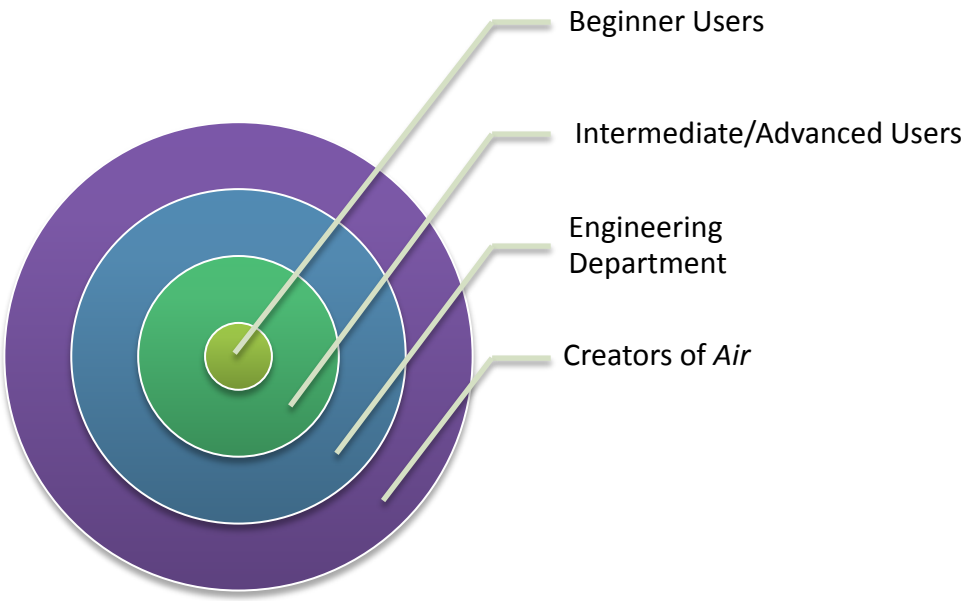
During the interview with the users and the client it was explained that the users currently refer to the user manual a great deal when creating new control files. It was also mentioned that the users often run tasks consecutively and tend to create their own sequences of tasks. An example sequence would be:

> **Task:** 5 hole probe at single point (unsteady)
> **Program Sequence:** LogVoltsOnce - ApplyCal - TFCorrect - Apply5HCal – Excel

The more experienced users also build their own bespoke programs from scratch. These programs are required to use the file formats as described above.

2.2.1. Stakeholders

Throughout our analysis process we have recognized that there are a large number of people that use the current system. We have identified the following stakeholders in the creation of a new user interface for the program suite; the closer they are to the centre, the more influence *Air* will have on them:



Typical Users

Beginner Users: Users that are not experienced enough to complete tasks fully without referring to the user manual for guidance. Our new system *Air* will aim to enable these users to be confident in creating and running tasks and reduce the need for the user documentation. These users would be the most affected by the new system as it will provide a graphical user interface to manage their session with the DSW suite. This will help reduce the steep learning curve that they are currently faced with. *Air* will provide several forms of help such as; helpful comments and validation of the user's inputs to ensure that the user is able to perform tasks with fewer errors.

Intermediate/Advanced Users: Users that are more experienced than beginners and are competent enough to utilise all the programs, requiring little to no help from the user manual. These users are also capable of building new programs (analysis and data-logging). These users will not be affected so greatly as they have a good knowledge and understanding of the programs and it takes them far less time to produce valid control files. *Air* may however, still reduce that time still further and facilities such as batch process creation will be of benefit.

Engineering Department: This refers to the engineering department as a whole, and is included only for completeness; the overall effect of the introduction of a new system has a 'knock on' effect on all connected to the wind tunnel project. The engineering department will be effected as *Air* will streamline the current use of the system by enabling users to focus more on the engineering aspects of their work, and less on the issues of correcting computer system errors.

Creators of *Air*: We are responsible for creating the new system *Air* which conforms to the client's requirements.

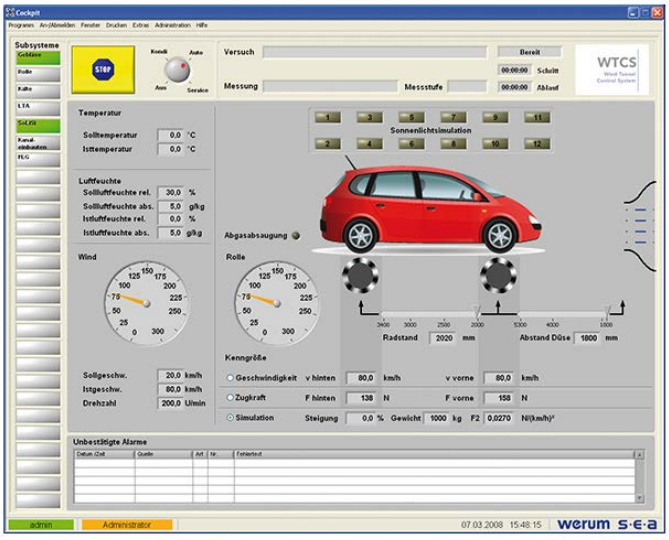
2.2.2. Current similarly marketed software

As the engineering department's current system as a whole is primarily bespoke and the programs (analysis and data-logging alike) are created to solve a problem specific to the client's specific Wind Tunnel setup, there does not exist a commercial program specifically targeted to our problem brief of

creating a GUI for the DSW suite. However, our research has found programs for wind tunnel control which may mirror some features of the existing suite.

For Example:

WTCS - Wind Tunnel Control System by S-E-A (Science and Engineering Applications)



[Image: S-E-A – Product Overview - WTCS - Wind Tunnel Control System. Accessed 14th November 2010]

This program mainly manages the control of a wind tunnel setup; it caters for recording and analysing aerodynamic, climatic and acoustic properties within that setup. The application has many features that form its main role, these are described below:

- configuration management
- error and event logger
- test sequencer & test management
- central data management
- data processing & storage
- data visualization
- data evaluation modules
- external programming interfaces

"User interfaces and dialogues are integrated seamlessly into the modular concept. Dedicated communication interfaces provide data exchange between the central servers. Dialogues are independent from the sub-systems and represent these on a functional level."

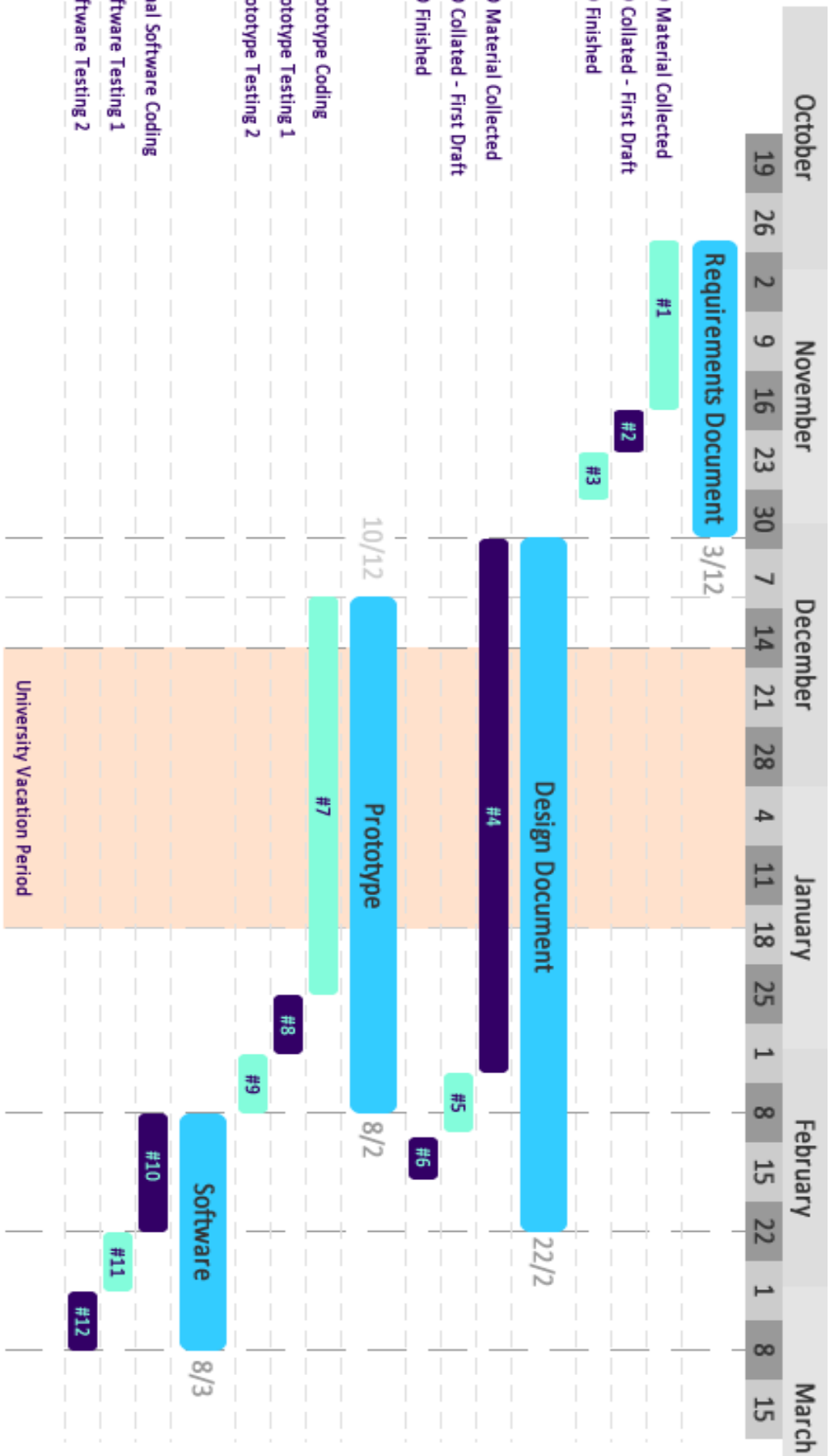
[Information: Wind Tunnel Control System- Features – Base Features. Accessed 14th November 2010]

3. Project Plan

Shown below is a breakdown of the various sections of the development of our *Air* software. This is displayed in both a table format, and as a Gantt chart. Note that although a deadline is shown for the requirements document at 03/12/10, this document will continually be amended during the lifespan of the project. The testing sections are split into 2 portions to allow time for both testing and feedback to occur within the 7 day time windows.

Key –
RD = Requirements Document
DD = Design Document

Item #	Item	Start Date	Soft Deadline	Hard Deadline	Duration (days)	Members Responsible
1	RD Material Collected	28/10/10	16/11/10		19	All
2	RD Collated - First Draft	16/11/10	21/11/10		5	AW
3	RD Finished	21/11/10	25/11/10	03/12/10	4	AW
4	DD Material Collected	03/12/10	03/02/11		62	All
5	DD Collated - First Draft	03/02/11	10/02/11		7	AW
6	DD Finished	10/02/11	15/02/11	22/02/11	5	AW
7	Prototype Coding	10/12/10	25/01/11	08/02/11	46	All
8	Prototype Testing 1	25/01/11	01/02/11		7	FM + HR
9	Prototype Testing 2	01/02/11	08/02/11		7	FM + HR
10	Final Software Coding	08/02/11	22/02/11	08/03/11	14	All (EH + NS)
11	Software Testing 1	22/02/11	01/03/11		7	FM + HR
12	Software Testing 2	01/03/11	08/03/11		7	FM + HR


October
November
December
January
February
March
19 26 2 9 16 23 30 7 14 21 28 4 11 18 25 1 8 15 22 1 8 15
Requirements Document
3/12
#1 RD Material Collected
#2 RD Collated - First Draft
#3 RD Finished
Design Document
22/2
#4 DD Material Collected
#5 DD Collated - First Draft
#6 DD Finished
Prototype
10/12
8/2
#7 Prototype Coding
#8 Prototype Testing 1
#9 Prototype Testing 2
Software
8/3
#10 Final Software Coding
#11 Software Testing 1
#12 Software Testing 2
University Vacation Period

4. Hardware and Software Platforms to be used

4.1. Hardware

Users will use a desktop computer next to each wind tunnel to run their experiments and gather appropriate data. Analysis of the data may be run on any computer which has access to the experiment's data.

If a computer can run Windows XP then it can run the Java Virtual Machine (JVM) and so the hardware requirements of *Air* are defined by those of Windows XP:

- Pentium 233-megahertz (MHz) processor or faster (300 MHz is recommended)
- At least 64 megabytes (MB) of RAM (128 MB is recommended)
- At least 1.5 gigabytes (GB) of available space on the hard disk

During development we will use the university servers, where we have a Subversion repository, to store *Air* and the DSW suite. This repository is only accessible to team members and provides a safe backup location for *Air* and related work. Another benefit is the magnetic tape data recovery offered by the university on their servers.

4.2. Software

The operating system that is widely used on Durham University computers is Windows XP. The client has specified that Windows XP is the operating system that *Air* will run on.

Air will be primarily written in Java using the Eclipse IDE. Our decision to use Java was mainly due to its portability; the DSW suite are Windows executable files and so the program will only be run on that OS, but using Java ensures *Air* will be compatible with future versions of the Windows OS because of the platform-independent nature of the JVM.

5. Solution Requirements

5.1. Functional Requirements

Req. ID	FR1 – Adding Programs
Description	*Air* will provide the ability to add new data logging and analysis programs to its list (or update existing programs) without modification of *Air* source.
Priority	High
Input	A user will add a new program to *Air* by the provision of a definition file for that program.
Operations	*Air* will check to ensure that the file has been created and is in the correct format.
Expected Results	The new program will be added to the list of available programs in *Air*. If the file is not in the correct format the user will be notified and the program will not be added.

Req. ID	FR1a – Definition File
Description	The definition file for a program will include all parameters accepted/required by that program, and the formats of their values.
Priority	High
Prerequisites	FR1
Input	A user adding a new program to *Air* will provide a file defining the command-line parameters and the control file parameters of the program and the required format of their values.
Operations	*Air* will check to ensure that the file has been created and is in the correct format.
Expected Results	The new program will be added to the list of available programs in *Air*. If the file is not in the correct format the user will be notified and the program will not be added.

Req. ID	FR1b - Definition File
Description	The definition file will include a short help tip taken from the user manual for each parameter. This tip will be displayed when the user is required to enter a value for that parameter.
Priority	High
Prerequisites	FR1
Input	A user adding a new program to *Air* will provide a file containing the above information.
Operations	*Air* will check to ensure that the file has been created and is in the correct format.
Expected Results	The new program will be added to the list of available programs in *Air*. If the file is not in the correct format the user will be notified and the program will not be added.

Req. ID	FR1c - Definition File
Description	The definition file for a program will include a description of the function of that program, in the same style as the user manual.
Priority	High
Prerequisites	FR1
Input	A user adding a new program to *Air* will provide a file containing the above information.
Operations	*Air* will check to ensure that the file has been created and is in the correct format.
Expected Results	The new program will be added to the list of available programs in *Air*. If the file is not in the correct format the user will be notified and the program will not be added.

Req. ID	FR2 – Initial Inclusion of Programs
Description	*Air* will, on release, allow the user to select and execute 20 of the programs in the DSW suite, including the most commonly used logging programs (LogVoltsOnce, Travlog5H, Log6xBAL) and analysis programs (ApplyCal and Apply5hCal).
Priority	High
Prerequisites	FR1
Input	User selects a program and can execute it after providing the correct command-line and control file arguments.
Operations	The program will supply a method of selecting a program/multiple programs.
Expected Results	Programs initiated by the user will execute without errors.

Req. ID	FR3 – Control File Creation
Description	*Air* will allow the user to create control files for all included programs which require them.
Priority	High
Prerequisites	FR1
Input	User enters values for the parameters required by the control file for the selected program.
Operations	*Air* will provide input fields for the parameters required by the program and validate user input for those parameters as defined in the definition file of that program. It will allow the user to enter other field/value pairs if they choose.
Expected Results	User will be prompted if they enter a field/value pair that is not in the definition file. Where the user has provided valid values for all the parameters required by the GUI and no others, the resulting control file will be valid for that program.

Req. ID	FR4 – Control File Modification
Description	*Air* will allow the user to modify existing control files for all included programs.
Priority	High
Prerequisites	FR1, FR3
Input	User selects an existing control file from their computer or a networked storage location. User enters or modifies values for the parameters required by the control file of the selected program, and can alter the fields/add new fields.
Operations	The program will interpret the control file and display the existing fields and values from it using the same interface and validation described in FR3.
Expected Results	User will be prompted if they enter a field/value pair that is not in the definition file. Where the user has provided valid values for all the parameters required by the GUI and no others, the resulting control file will be valid for that program.

Req. ID	FR5 – File Selection
Description	*Air* will provide a file browser interface for selecting input files. Users will be able to manually enter file paths for non-existent files.
Priority	High
Prerequisites	None
Input	The user will specify a file by selecting it from a file browser interface or typing it in.
Operations	When a file path is required, *Air* will provide a text field as well as an interface which displays existing files for the user to select from.
Expected Results	*Air* will warn users if they enter a path to a file that does not exist.

Req. ID	FR6 - Help
Description	*Air* will provide real-time help information e.g. descriptions of parameters and functionality of DSW programs.
Priority	Medium
Prerequisites	None
Input	The user will select an item (program, field, input device etc).
Operations	*Air* will use the user manual and/or the program's associated definition file to find relevant help information.
Expected Results	*Air* will display help information for any selected item.

Req. ID	FR7 – Batch Execution
Description	*Air* will provide the ability to set up batch execution of multiple programs operating on multiple files.
Priority	Medium
Input	The user will specify programs, control files and parameters and the data files to be operated on by the programs.
Operations	*Air* will compile the program/file sequence into a batch file and then allow the user to execute it through the GUI.
Expected Results	The user will be able to set up and then execute a batch of operations through *Air*.

Req. ID	FR8 – Run-time Interaction
Description	*Air* will display the output data for those programs requiring run-time interaction.
Priority	Low
Prerequisites	FR1
Input	User responds to prompts from *Air*.
Operations	*Air* will display the output data and provide methods for the user to respond to that data. *Air* will monitor the command-line window in case the user interacts directly with the command-line instead of through *Air*.
Expected Results	The user will be able to see and respond to any input requests from executing programs through *Air*. If the user interacts directly with the command-line then *Air* will stop prompting the user for input.

Req. ID	FR9 – Program Sequence Creation
Description	*Air* will allow users to save their own sequences of commonly used programs.
Priority	Low
Input	User creates a batch file using *Air*. User then selects whether to save that program sequence for later use, and whether to save it just for themselves or for every user.
Operations	*Air* will store the program sequence either globally (for all users) or locally (for that user only) as the user requires.
Expected Results	The user will be able to save preset sequences of programs.

Req. ID	FR10 – Loading Saved Sequences
Description	*Air* will allow users to reload their own sequences of commonly used programs.
Priority	Low
Prerequisites	FR9
Input	The user selects an existing batch file or saved program sequence.
Operations	*Air* will load the file ready to be run again.
Expected Results	The user will be able to reload preset sequences of programs without rewriting the entire sequence from scratch.

Req. ID	FR11 – Editing Saved Sequences
Description	*Air* will allow users to alter their own sequences of commonly used programs.
Priority	Low
Prerequisites	FR9, FR10
Input	The user enters changes to be made to the batch file/program sequence.
Operations	*Air* will validate any changes as described by FRs 1-5.
Expected Results	The user will be able to alter preset sequences of programs to, for example, reuse the basic sequence on a new set of files. The user will only be able to directly change batch files/program sequences they created themselves. Any alteration of someone else's will cause a new file to be created instead of the original being overwritten.

Req. ID	FR12 – Commonly Used Sequences
Description	*Air* will present the user with a list of recently/commonly used program sequences to facilitate program sequence generation.
Priority	Low
Prerequisites	FR14
Operations	*Air* will maintain usage data on existing program sequences to determine which are most commonly/frequently used.
Expected Results	*Air* will display a list of the most recently/commonly used sequences and allow the user to select and load from that list.

Req. ID	FR13 – Help Options
Description	*Air* will allow the user to select from three levels of visible help: • No Help • Simple Help • Full Help
Priority	Low
Prerequisites	FR6
Input	User selects one of the three options.
Expected Results	**No Help:** *Air* does not automatically display any help but will still allow the user to specifically request it. **Simple Help:** *Air* does not automatically display verbose help, but still displays small tips and allowed values while allowing the user to specifically request any verbose help they desire. **Full Help:** *Air* automatically displays all default help.

Req. ID	FR14 - Tooltips
Description	Almost all default help will appear in the form of tooltips or upon selection of a specific item. Almost all other help will only appear when specifically requested.
Priority	Low
Prerequisites	FR6
Input	User hovers the cursor over or selects an item.
Expected Results	*Air* displays the default help for that action.

5.2. Non-Functional Requirements

Req. ID	NFR1 – Consistent Colour Scheme
Description	*Air* will have a consistent colour scheme throughout.

Req. ID	NFR2 – Consistent Layout
Description	*Air* will have a consistent content layout throughout.

Req. ID	NFR3 – Font Style
Description	*Air* will use an easy-to-read font in a clear size and colour, which will be determined during the design phase.

Req. ID	NFR4 – Program Icon
Description	*Air* will have a recognisable program icon, which will be created during the design phase.

Req. ID	NFR5 – Section Sizing
Description	Sections of the GUI will be appropriately sized for their content – content should all fit inside its section without too much wasted space.

Req. ID	NFR6 – Response Time
Description	The system will respond to user inputs within an acceptable time period (~0.1s).

Req. ID	NFR7 – Layout Design
Description	*Air* will have a logical and intuitive layout designed from the feedback given by existing users.

Req. ID	NFR8 – Window Resolution
Description	The main window will fit on the most common desktop resolutions.

Req. ID	NFR9 – Time to Execution
Description	Using *Air*, a beginner user will (on average) be able to set a single program or a combination of programs running correctly in less than ⅓ of the time it would take using the existing system.

6. Definitions of Terms

Air	The name given to our software solution.
Argument	The value given to a specific instance of a parameter.
ASCII File	ASCII or 'American Standard Code for Information Interchange' is a common way of encoding text data as it is recognised by the majority of machines. This is the format of the files the Tecplot program currently uses.
C++	A programming language developed by Bjarne Stroustrup.
Command-Line	The command-line is a text based interface where users can type in commands which the computer will execute. The specific command-line interface (CLA) referred to in this project is Microsoft Windows Command Prompt.
Data Analysis Program	A data analysis program is a program which takes a collection of pre-existing data and performs a function or set of functions on it to convert the raw data into another form of more useful information.
Data Logging	Data logging is the collection of data from a specified source over a period of time.
Data Logging Program	A data logging program is a program which will digitally record the output from a piece of data logging equipment.
DSW	Durham Software for Wind Tunnels.
Eclipse	An open source IDE developed by the Eclipse Foundation.
Front-End	A front-end is an abstract term for the components of the software which are displayed to the user to interact with. Here this will refer principally to the GUI/CLA.
GUI	GUI stands for Graphical User Interface - this is the visual component of a piece of software which allows the user to navigate through documents/software and carry out commands using click buttons, lists, hotkeys etc.
IDE	Integrated Development Environment – this is used to aid in software development.
Java	A programming language developed by Sun Microsystems.
Java Virtual Machine (JVM)	Interprets compiled Java binary code for a computer's processor, so that it can perform a Java program's instructions.
Magnetic Tape	A physical storage medium for digital data.
Network Drive	A shared storage location that is made available to several users over a network.

Parameter	A specific field required by a program. For the program to execute all parameters must be provided with arguments (see above).
Runtime	The time during which a program is in execution.
Server	A computer that handles requests for resources.
Subversion	A software versioning and revision control system.
Tecplot	The commercial application currently used by the Engineering department to visualise data outputted from the Durham Software for Wind Tunnels suite.
Tooltips	A graphical component of the interface displayed at the cursor point.
Wind Tunnel	A tube-like structure where wind is produced, usually by a large fan, to flow over a test object. The object is connected to instruments that measure and record aerodynamic forces that act upon it. (Yager, 1994)
Windows XP	An operating system produced by Microsoft, which is prevalent on the computers used by the university.

7. References

Hessdörfer, Jörg; Kühnle Ulf. (2009). *Wind Tunnel Control System*
http://www.sea-gmbh.com/en/products/product-overview/software/wtcs/. Last accessed 8th Nov 2010.

Yager, Bryan. (1994). *Glossary*
http://www.nas.nasa.gov/About/Education/Racecar/glossary.html. Last accessed 4th Nov 2010.

Sims-Williams, D. (2010). User Manual. *Durham Software for Wind Tunnels Version 2.0+*
Last accessed 2nd Dec 2010.