



Momentum Software Engineering

School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6

CMPT 275 Group T

Momentum Software Engineering's

Flight the Freights Manager

Integration Report

Revision 1

July 31, 2002

Approved by: (By alphabetical order)

Vincent Chu _____
Steven Essen _____
Sung Hwang _____
David Jeong _____



Revision History Page

Revision 1

-Written jointly by David Jeong, Sung Hwang, Steven Essen
and Vincent Chu. Compiled together into one document.



Table of Contents

- Title Page.....p.
- Release History Page.....p.
- Table of Contents.....p.

1.0 Integration Discussion.....p.

2.0 Black Box Test Results

- 2.1 Performance Test Result.....p.
- 2.2 Stress Test Result.....p.
- 2.3 Functional Test Result.....p.

3.0 Project Summary

- 3.1 Bug Report.....p.
- 3.2 Project Difficulties.....p.
- 3.3 The Things Momentum Did Right.....p.
- 3.4 Recommendations to Other Groups.....p.



Section 1.0 – Integration Discussion

The integration of Flight the Freights Manager involved linking 8 modules (or classes) together: Main, ui, Customer, Cargo, Flight, fCust, fCargo, and fFlight. Each module was tested individually using stubs and drivers. Even though Momentum created stubs and drivers in hopes of doing top-down integration or even bottom-up integration, Momentum, in the end, decided to do a big bang integration.

The reason Momentum chose big bang integration was that Momentum wanted to see if all the implemented modules can work together on the first try. Another reason was that time was limited as the amount of time allotted to integration was only 1.5 weeks instead of two weeks. Since Momentum encountered several bugs when doing system testing, lots of time was needed to fix the bugs in order to demo the system to the customer right after the integration period.

Big bang integration meant that Momentum put all the implemented modules in the same location (i.e. directory) and tried to link and run them at the same time. Momentum hoped for the best that there would be no major problems when they did this. Indeed, no major problems were encountered except for assigning a cargo item which did not work properly. However, when errors popped up, pinpointing the module or modules that were causing the errors was difficult.

Section 2.0 – Black Box Test Results

For the description, preconditions, test procedure, and expected results of each test, please refer to sections 7.1, 7.2, and 7.3 of the Architectural Design Document. In this section, the actual results are discussed as well as the pass/fail criterion of the performance test.

2.1 Performance Test Result

Momentum will performance test the system as per the instructions indicated in section 7.1 of Architectural Document. The test is run against release 4.

The following is the screen dump generated:

```
====Main Menu====
1) Customer Maintenance
2) Cargo Maintenance
3) Flight Maintenance
4) Reports and Inquiries
0) Exit
Enter the number you want(0-4):
1
```



```
=====Customer Maintenance=====
1) Add a new customer
0) Go back to Main Menu
Choose an option [0-1] and press ENTER:
1
Enter the Customer Name (Length: 30 max.)
Chris McNeal
Enter the phone number of the customer (Length: 13 max.)
(555)123-1234
Are you sure you want to add the customer Chris McNeal(Y/N)?
Y
Do you wish to add another customer(Y/N)?
N
=====Customer Maintenance=====
1) Add a new customer
0) Go back to Main Menu
Choose an option [0-1] and press ENTER:
1
Enter the Customer Name (Length: 30 max.)
```

The test has failed. The list of all customers is not displayed. According to the test case, a list of customer, including the one we just entered should be displayed. However, such list is not seen.

2.2 Stress Test Result

Momentum will stress test as per the instructions indicated in section 7.2 of Architectural Document. The test is run against release 4.

The following is the screen dump generated:

```
=====Main Menu=====
1) Customer Maintenance
2) Cargo Maintenance
3) Flight Maintenance
4) Reports and Inquiries
0) Exit
Enter the number you want(0-4):
2
=====Cargo Maintenance=====
1) Add a new cargo item
2) Assign a cargo item to a flight
3) De-assign a cargo item from a flight
4) Delete a cargo item
5) Reassign all cargo on a flight to a different flight.
0) Go back to Main Menu
Choose an option [0-5] and press ENTER:
1
=====Add a new cargo=====
```



```
1) Standard Container
2) Pallet
0) Cancel
Choose an option [0-2] and press ENTER:
2
Enter the Customer Name(Length: 30 max.):
Issac Newton
Enter the cargo weight (0.1 - 999999.9):
Thomas
Invalid data. Please re-enter a numerical cargo weight.
Enter the cargo weight (0.1 - 999999.9):
90dave
Invalid data. Please re-enter a numerical cargo weight.
Enter the cargo weight (0.1 - 999999.9):
k103
Invalid data. Please re-enter a numerical cargo weight.
Enter the cargo weight (0.1 - 999999.9):
```

The test is passed. The result generated is exactly as that indicated in the test case.

2.3 Functional Test Result

Momentum will functional test as per the instructions indicated in section 7.3 of Architectural Document. The test is run against release 4.

The following is the screen dump generated:

```
=====Main Menu=====
1) Customer Maintenance
2) Cargo Maintenance
3) Flight Maintenance
4) Reports and Inquiries
0) Exit
Enter the number you want(0-4):
1
=====Customer Maintenance=====
1) Add a new customer
0) Go back to Main Menu
Choose an option [0-1] and press ENTER:
1
Enter the Customer Name (Length: 30 max.)
Johnthisisalongname Idontknowi.
Enter the phone number of the customer (Length: 13 max.)
```

The test is passed. The result generated is exactly as that indicated in the test case.

Section 3.0 – Project Summary



3.1 Bug Report

3.2 Project Difficulties

(In this and the following sections, “we” refers to Momentum Software Engineering.)

As we developed the airfreight load management system, we faced many problems and difficulties.

One of the problems we encountered was the lack of suitable meeting times. As each member took different classes and jobs, it was hard for us to meet in one place and share the ideas of each one of members with enough time. This sort of caused a lack of communication among team members.

The next problem was the lack of communication among team members which caused inconsistencies in documentations. An example of this occurred when two of our members were supposed to write how to use each function, but they did not discuss the format for the screen output. When we got back their documentations, we noticed the significant format difference between one and the other, so other group member had to spend extra time to fix the screen output format.

Finally, underestimating time was also another problem. When we doing Release 3 of the product, we were in hurry and did not have enough time to correctly do and proofread the interfaces. We spent too much time on doing the Object Communication Diagrams (OCDs) for each scenario. Some errors were found in documents so we tried to fix them; however, we failed to do so in time and had to release the product with still some errors. As a result, when doing the implementation of the modules, we found even more errors (mostly in the interfaces but also in the OCDs).

3.3 The Things Momentum Did Right

Although we made some mistakes, we could finish our project without any major disasters. This was all possible because we tried many methods to solve as many difficulties as possible.

We planned our time efficiently most of the time. We started each assignment as soon as Mr. Russ Tront posted the assignments. We also divided our workloads evenly, so that no one would have too much work or too little work.

As we mentioned in the section “project difficulties”, we had a problem of lack of communication because everyone was taking courses on different time. We coped with this problem by using the Internet chatting program. Since everyone had an Internet connection, it was easy for us to send our drafts/documents to each other in order proofread and fix the errors.



Our group also backed up document files and code files as often as possible. We did all of our back up through SFU E-mail. And for the emergency situation, we made another back up on the website of one of our group member. This enable us to work with our files without having worries of losing or contaminating the files.

3.4 Recommendations to Other Groups

Reading is one of the most important things to do for a big project such as in this course. You must read not only your coursepak but also the assignment PDFs. The assignment PDF is your customer's request to you and your company. It is your obligation to read it and make sure to find out what the customer wants from you.

Checking and reviewing are also important. Sometimes, finding the bugs of your own work can be hard. That's why you let others review your work and you review the other members' work. By doing so, you can find errors on the assignment easily and more efficiently. This will help you a lot to make your project one step closer to the perfection.

Finally, do not procrastinate and do not wait till the last minute when completing the project or else you will be in rush to get it done. This could result in poor quality projects with lots of errors and inconsistencies.