

# A(UTOMATED) F(ILM) D(ISTRIBUTION), C(REATION), AND C(OLLECTION) K(IOSK) PROJECT DOCUMENTATION EVALUATION

A 2813ICT Software Engineering Fundamentals  
Assignment

## Abstract

This document contains the demonstration of the application of several software best practices on the AFDCK project in addition to answering the questions asked by assignment 2

Sebastian Perry  
sebastian.perry@griffithuni.edu.au

## Contents

Critical review of the initial documentation (1): .....	2
Issue 1: .....	2
Issue 2: .....	2
Issue 3: .....	2
Question 10 revision (2): .....	3
How issue 1 (or question 10) could be resolved: .....	3
Answer Revision Justification: .....	3
Project testing plan (3): .....	4
Overall Test Strategy: .....	4
Projected testing team costs tables: .....	4
System Level Test: .....	4
Integration Testing: .....	4
Unit Testing: .....	4
Test Cases based on listed Requirements (4): .....	5
Test Case 1: .....	5
Test Case 2: .....	6
Test Case 3: .....	7
Test Case 4: .....	8
Project Goals (5): .....	9
Project Goal 1: .....	9
Goal: .....	9
Question: .....	9
Metric: .....	9
Justification: .....	9
Project Goal 2: .....	9
Goal: .....	9
Question: .....	9
Metric: .....	9
Justification: .....	9
Process Objectives of PM.02 (6): .....	10
Process activities of PM.02 (7): .....	10
Work Products, Process Activities, and Process Objectives table (8): .....	11
Appendixes: .....	13
Appendix 1 Bibliography: .....	13

## Critical review of the initial documentation (1):

Below is a set of three issues that were discovered through self-revision and third party feedback.

### Issue 1:

For the Requirement Behaviour Tree and the Integrated Behaviour tree (question 9 and 10 respectively) the displayed graph is not a true oriented tree as there is a point where two parent nodes link up to the same child nodes; fixing this graph up to the standards and specifications of an oriented tree would aid the project by removing any ambiguity that could cause a different interpretation of intended behaviour and thus a product which deviates from the clients initial vision.

### Issue 2:

For the Component Behaviour Tree (question 11) some of the ending nodes located at the ends of the tree's branches lack the revision symbol stating that the process would start over again from the beginning, because of this the tree could not be turned into a state machine; If this problem was rectified the component behaviour tree would be usable as a state machine, this would allow developers of the project to use it towards the end of the production phase for checking that the product behaves as originally planned.

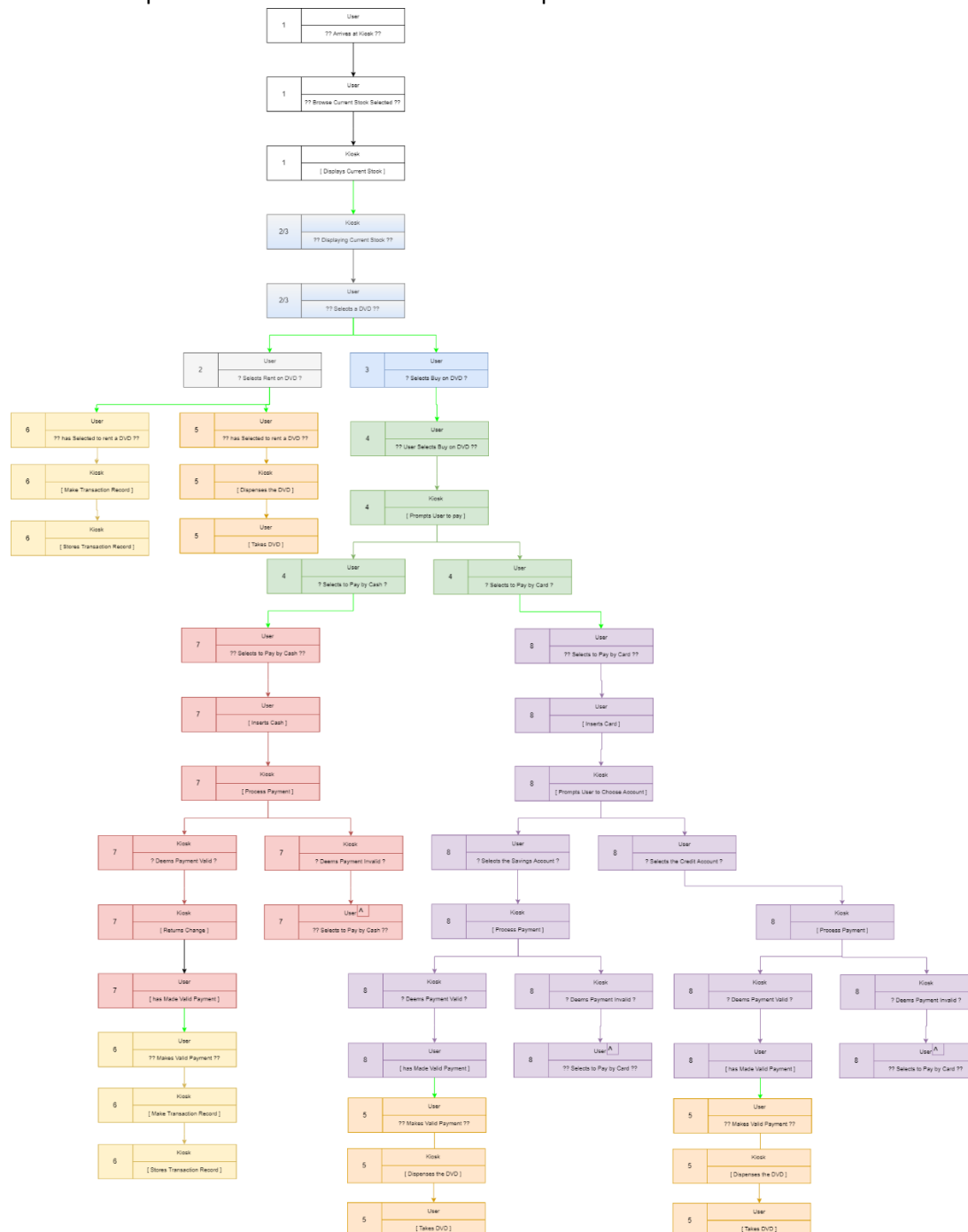
### Issue 3:

There is a lack of correspondence between the Component Interaction Network and the class Diagram (questions 12, and 14 respectively), this is due to the class diagrams containing classes for features that weren't discussed as part of the aforementioned component behaviour trees; the rectification of this issue would be invaluable to the project as correct and complete planning of all features desired in the final product is the only way to ensure sure features are implemented into the products final version.

Question 10 revision (2):

How issue 1 (or question 10) could be resolved:

The Requirement Behaviour and Integrated Behaviour trees could be fixed by removing the connection between two parents and a child, this was replaced with both parents reconnecting to their own duplicate of the child node and it subsequent sub tree.



Answer Revision Justification:

All instances where 2 parent nodes hooked into the same child node have been removed and subsequently been replaced with duplications of the child node and connected sub tree, this is a clear improvement over the previous iteration of the diagram as it follows the standards expected of an oriented tree diagram.

## Project testing plan (3):

### Overall Test Strategy:

The overall plan on testing consists of two main focuses, one insuring that all core system fuctions work efficently and correctly in order to ensure smooth running of the business, and two testing the user experience and allowing for the generation of feedback to the developers with the end goal of making a product which can be intuitivly used by the majority of users without any prior information. In order to achieve these goals the system level testing will consist of a series of Customer (/end User) Tests, the intergration testing will consist of Top Down test model with a particualar focus on regression testing, and lastly for the unit testing portion the Black Box testing process will be used. To complete these tests the graphical interface will be loaded onto a tablet device for so that is can be easily be tested by people inside and outside the development enviroment. The testing team (breakdown in teble form can be seen below) will consisting of one test manager and 3 software testers will be used on and off through development over the course of development to a total predicted time of around 1 year and predicted cost of ammount of \$301,229.

### Projected testing team costs tables:

The information showed in this table was gathered from sources which can be found in the bibliography (appendix 1).

Job Title	Predicted Yearly Salery	Number of Each Required	Total Cost
Test Manager	\$113,780	1	\$113,780
Software tester	\$62,483	3	\$187,449
Overall total			\$301,229

### System Level Test:

On the system Level our teams and select individuals from the company as in additon to volunteers from the general public (the intended end users) will be asked to conduct Customer (/ end User) Acceptance Tests. The reasoning for this choice is that any time someone uses the kiosks in the real world they won't have an employee to give them instructions for how to use the machine, it is for this reason that it would be advantagous to make a kiosk with designs, features, guis which feels intuitive and natural to the inteneded user.

### Integration Testing:

For the intergration Testing the Top Down method with a focus on regression testing will be used to ensure the protection of work down to the high level gui modules and it's various graphic features. As the development of the user experience is so important in the context of the kiosk great efforts will be made to ensure any and all features and improvements do not get lost in the intergration of new product features.

### Unit Testing:

During the unit testing phase the White Box testing process will be used to analyse the products proformance from the perspective of the average end user (e.g little to no prior knowledge on the kisoks use). This choice will allow the gathering of information and feedback from the perspective of the end user and made modification accordingly with the end goal of creating a system that is both easy and intuitive in use.

## Test Cases based on listed Requirements (4):

### Test Case 1:

PROJECT:		AFDCCK		
MODULE:		Browse Current Stock		
REQUIREMENT		R1		
TEST CASE ID:				
TEST OBJECTIVE:		To test if the browse current stock button is functioning correctly		
TEST DATE AND TIME				
STEP NO.	STEP	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Without having signed in select to browse current stock	Push the browse current stock button	The screen should change to display the current available stock, but you should not be able to proceed continue with the transaction until logged in	
2	having signed in select to browse current stock	Push the browse current stock button	The screen should change to display the current available stock	

## Test Case 2:

PROJECT:		AFDCCCK		
MODULE:		Rent		
REQUIREMENT		R2,R5		
TEST CASE ID:				
TEST OBJECTIVE:		To test if the act of renting a DVD functions correctly, and if used correctly it will: make a transaction record and dispense a DVD		
TEST DATE AND TIME				
STEP NO.	STEP	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Without having signed in and selected to browse current stock, select to rent a DVD	Push the rent button on the desired dvd	The screen should prompt the user to login before this action can be possible	
2	having signed in and selected to browse current stock, select to rent a DVD	Push the rent button on the desired dvd	The screen should change to display the rent a dvd form with the appropriate DVD's details filled in where appropriate	
3	Fill in the displayed rent form with the duration if the rent in days	"abc"	The screen should prompt the user that "no number has been put into the duration field"	
4	Fill in the displayed rent form with the duration if the rent in days	"-14"	The screen should prompt the user that "no valid number has been put into the duration field"	
5	Fill in the displayed rent form with the duration if the rent in days	"15"	The screen should prompt the user that "The duration of rent is too long, the max is 14"	
6	Fill in the displayed rent form with the duration if the rent in days	"0"	The screen should prompt the user that "The duration of rent is too short, the minnium is 1"	
7	Fill in the displayed rent form with the duration if the rent in days	"12"	The screen should display a "success" message then move on to dispensing the desired DVD	
8	The machine should now create a transaction record		A transaction record is created	
9	Wait for the machine to dispense a DVD		The machine should dispense the desire DVD	

### Test Case 3:

PROJECT:		AFDCK		
MODULE:		Buy with cash		
REQUIREMENT		R3, R4, R6, R7, R5		
TEST CASE ID:				
TEST OBJECTIVE:		To test if the operation of buying a DVD with cash is functioning correctly, and if used correctly it will: make a transaction record and dispense a DVD		
TEST DATE AND TIME				
STEP NO.	STEP	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Without having signed in and selected to browse current stock, select to buy a DVD	Push the buy button on the desired dvd	The screen should prompt the user to login before this action can be possible	
2	having signed in and selected to browse current stock, select to buy a DVD	Push the buy button on the desired dvd	The screen should change to display the buy a dvd form with the appropriate DVD's details filled in where appropriate	
3	After filling in the buy form select confirm	Select confirm	The display should prompt the user with the question of payment method	
4	Select the pay by cash payment method	Select pay by cash	The display should inform the user on the amount of money required	
5	Check reaction to not giving enough funds	Give the machine insufficient amounts of money	The machine's display of the amount of money required should decrease by the amount given	
6	Check the machine's reaction to too many funds	Give the machine too much currency	The machine should consider the payment complete and dispense valid funds before proceeding	
7	Check the machine's reaction to the right amount	Give the machine the correct amount of required funds	The machine should consider the payment complete and proceed	
8	The machine should now create a transaction record		A transaction record is created	
9	Wait for the machine to dispense a DVD		The machine should dispense the desired DVD	



#### Test Case 4:

PROJECT:		AFDCCK		
MODULE:		Buy with card		
REQUIREMENT		R3, R4, R6, R7, R5		
TEST CASE ID:				
TEST OBJECTIVE:		To test if the operation of buying a DVD with card is functioning correctly, and if used correctly it will: make a transaction record and dispense a DVD		
TEST DATE AND TIME				
STEP NO.	STEP	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Without having signed in and selected to browse current stock, select to buy a DVD	Push the buy button on the desired dvd	The screen should prompt the user to login before this action can be possible	
2	having signed in and selected to browse current stock, select to buy a DVD	Push the buy button on the desired dvd	The screen should change to display the buy a dvd form with the appropriate DVD's details filled in where appropriate	
3	After filling in the buy form select confirm	Select confirm	The display should prompt the user with the question of payment method	
4	Select the pay by card payment method	Select pay by cash	The display should inform the user on the amount which will be withdrawn from their account upon payment	
5	Check reaction to an invalid card	Insert an invalid card into the machine	The machine should check the card and upon discovering it to be invalid should inform the user as such and give them the opportunity to try again	
6	Check reaction to a valid card	Insert a valid card into the machine	The machine should check the card and upon discovering it to be valid the machine should make the withdrawal request on the account and when complete should proceed with the transaction	
8	The machine should now create a transaction record		A transaction record is created	
9	Wait for the machine to dispense a DVD		The machine should dispense the desired DVD	

## Project Goals (5):

### Project Goal 1:

Goal: ensure the project stays within budget

Question: during the last development was the budget exceeded and if so by how much?

Metric: *% by which the budget was exceeded* =  $\frac{\text{actual expenditures}}{\text{projected budget}} \times 100$

Justification:

The chosen metric is both reliable and valid, this will be discussed now. It is reliable because it can be easily repeated every development cycle and while it would only have a sample size of 1 after the first development cycle, it can be computed with the information from subsequent cycles to calculate the average overspending percentage per cycle thus making it a more accurate reading. It is valid as overspending in a development cycle has a direct correlation with going over the project budget. This metric will be measured by making an expenditure prediction at the beginning of the development cycle, recording incurred expenditures during the development cycle, and compare the actual expenditure recorded against the expenditure first predicted while also using these two numbers to calculate the ' % by which the budget was exceeded ' (the formula to which can be found above).

### Project Goal 2:

Goal: ensure the project stays within the planned time frame

Question: during the last development cycle did any tasks exceed their planned time frame

Metric: *% of overtime in relation to predicted time* =  $\frac{\text{actual time frame}}{\text{projected time frame}} \times 100$

Justification:

The chosen metric is both reliable and valid, this will be discussed now. It is reliable because it can be easily repeated every development cycle and while it would only have a sample size of 1 after the first development cycle, it can be computed with the information from subsequent cycles to calculate the average overtime (as a percentage) that was required to generally complete goals that was set by the team at the beginning of the cycle, calculating this average would have the effect of supplying a more accurate reading. It is valid as going into overtime in a development cycle has a direct correlation with not completing the project in the planned amount of time. This metric will be measured by making a decisions will be made at the beginning of the development cycle which will pertain to what should be completed by the end of the development cycle, during the development cycle the time required to complete the various tasks should be recorded, and upon completion of the allotted tasks the total time should be recorded; this total required time should be compared against the predicted while also using these two numbers to calculate the ' % of overtime in relation to the predicted time ' (the formula to which can be found above).

## Process Objectives of PM.02 (6):

Below is a list of the detailed process objectives to the PM.02 based on the ISO/IEC DTR 29110-5-1-1

(Can be found on pages 5-6 of the ISO/IEC 29110)

### 6.3.2 Project Assessment and Control Process

- a) progress of the project is monitored and reported;
- d) project objectives are achieved and recorded.

### 6.3.7 Measurement Process

- d) the required data are collected, stored, analysed, and the results interpreted; and

### 6.4.8 Software Acceptance Support Process

- a) the product is completed and delivered to the acquirer;

## Process activities of PM.02 (7):

Below is a list of the detailed process activities to the PM.02 based on the ISO/IEC DTR 29110-5-1-1:

(Can be found on pages 12-14 of the ISO/IEC 29110)

### 6.7.1.2 PM.2 Project plan execution (PM.02, PM.03, PM.04, PM.05, PM.07)

The Project Plan Execution activity implements the documented plan on the project. The activity provides:

- Monitoring the project against the Project plan.
- Status of the Project Plan Execution
- Change Request accepted by the Customer
- Reviews and agreements with the Customer

### 6.7.1.3 PM.3 Project assessment and control (PM.02)

The Project Assessment and Control activity evaluates the performance of the plan. The activity provides:

- Evaluation of actual plan performance and progress against targets.
- Track change requests.
- Documented problem, corrective action defined, and tacked to closure.

### 6.7.1.4 PM.4 Project closure (PM.02)

The Project Closure activity provides the project's documentation and products in accordance with contract

requirements. The activity provides:

- Support of Customer product acceptance.
- Completion of the project and sign of the Acceptance Record
- Summary and updated project repository for project closure

## Work Products, Process Activities, and Process Objectives table (8):

Below is a table displaying the work products, process activities, and process objectives:

(Can be found on pages 10-14 of the ISO/IEC 29110)

Work Products	Process Activities	Process objectives
Statement of Work	Project Planning	<b>PM.1.1</b> Review the Statement of Work <b>PM.1.2</b> Identify the specific tasks to be performed in order to produce the deliverables and their software components identified in the Statement of Work. Include tasks in the SI process along with verification, validation and reviews with Customer and Work Team tasks to assure the quality of work products. <b>PM.1.4</b> Identify and document the resources: human, material, equipment and tools.
Project Repository	Project Planning	<b>PM.1.11</b> Establish the project repository.
	Project Closure	<b>PM.4.2</b> Update Project Repository.
Meeting Record	Project Plan execution	<b>PM.2.2</b> Conduct meetings with the Customer, record agreements and track them to closure. Change Request initiated by Customer, needs to be negotiated to reach acceptance of both parties.
	Project Closure	<b>PM.4.2</b> Update Project Repository.
Change Request	Project Plan execution	<b>PM.2.2</b> Conduct meetings with the Customer, record agreements and track them to closure. Change Request initiated by Customer, needs to be negotiated to reach acceptance of both parties.
	Project Assessment & control	<b>PM.3.2</b> Evaluate and Track the changes request from customer.
	Project Closure	<b>PM.4.2</b> Update Project Repository.
Progress Status Record	Project Plan execution	<b>PM.2.1</b> Monitor and record status of the Project Plan execution. <b>PM.2.2</b> Conduct meetings with the Customer, record agreements and track them to closure. Change Request initiated by Customer, needs to be negotiated to reach acceptance of both parties.
	Project Assessment & control	<b>PM.3.1</b> Evaluate project progress with respect to the Project Plan, comparing: <ul style="list-style-type: none"> <li>- actual tasks against planned tasks</li> <li>- actual resource allocation against planned resources</li> <li>- actual cost against budget estimates</li> <li>- actual time against planned schedule</li> <li>- actual risk against previously identified</li> </ul> <b>PM. 3.3</b> Establish actions to correct deviations or problems and track them to closure.
	Project Closure	<b>PM.4.2</b> Update Project Repository.

Work Products	Process Activities	Process objectives
Project Plan	Project Planning	<p><b>PM.1.2</b> Identify the specific tasks to be performed in order to produce the deliverables and their software components identified in the Statement of Work. Include tasks in the SI process along with verification, validation and reviews with Customer and Work Team tasks to assure the quality of work products.</p> <p><b>PM.1.3</b> Establish the Estimated Duration to perform each task.</p> <p><b>PM.1.4</b> Identify and document the resources: human, material, equipment and tools</p> <p><b>PM.1.5</b> Establish the Composition of Work Team assigning roles and responsibilities according to the Resources.</p> <p><b>PM.1.6</b> Assign estimated start and completion dates to each one of the tasks in order to create the Schedule of the Project Tasks.</p> <p><b>PM.1.7</b> Calculate and document the project Estimated Effort and Cost.</p> <p><b>PM.1.8</b> Identify and document the risks which may affect the project.</p> <p><b>PM.1.9</b> Generate the Project Plan integrating the elements previously identified and documented.</p> <p><b>PM.1.10</b> Review and accept appropriate parts of the Project Plan. Customer reviews and accepts the Project Plan.</p> <p><b>PM.1.11</b> Establish the project repository.</p>
	Project Plan execution	<p><b>PM.2.1</b> Monitor and record status of the Project Plan execution.</p> <p><b>PM.2.2</b> Conduct meetings with the Customer, record agreements and track them to closure. Change Request initiated by Customer, needs to be negotiated to reach acceptance of both parties.</p>
	Project Assessment & control	<p><b>PM.3.1</b> Evaluate project progress with respect to the Project Plan, comparing:</p> <ul style="list-style-type: none"> <li>- actual tasks against planned tasks</li> <li>- actual resource allocation against planned resources</li> <li>- actual cost against budget estimates</li> <li>- actual time against planned schedule</li> <li>- actual risk against previously identified</li> </ul>
	Project Closure	<p><b>PM.4.1.</b> Formalize the completion of the project, providing acceptance support and getting the Acceptance Record signed.</p> <p><b>PM.4.2</b> Update Project Repository.</p>
Software Configuration	Project Closure	<p><b>PM.4.1.</b> Formalize the completion of the project, providing acceptance support and getting the Acceptance Record signed.</p>
Acceptance Record	Project Closure	<p><b>PM.4.1.</b> Formalize the completion of the project, providing acceptance support and getting the Acceptance Record signed.</p> <p><b>PM.4.2</b> Update Project Repository.</p>

## Appendixes:

### Appendix 1 Bibliography:

Payscale.com. (2019). Software Tester Salary in Australia | PayScale  
[online]  
[https://www.payscale.com/research/AU/Job=Software\\_Tester/Salary](https://www.payscale.com/research/AU/Job=Software_Tester/Salary)  
[Accessed 18 Sep. 2019].

Payscale.com. (2019). Test Manager Salary in Australia | PayScale  
[online]  
[https://www.payscale.com/research/AU/Job=Test\\_Manager/Salary](https://www.payscale.com/research/AU/Job=Test_Manager/Salary)  
[Accessed 18 Sep. 2019].

ISO/IEC. (2012). ISO/IEC 29110 Software Development Lifecycle Model for Very Small Entities (VSEs).  
[Accessed 18 Sep. 2019].