RIHAM AHAMED ABDUL RAHEEM

HND COMPUTING IDM

Feasibility Study

Contents

[**FEASIBILITY CRITERIA** 1](#_Toc76316381)

[Problem 1: Receive information of customer orders 2](#_Toc76316382)

[Problem 2: Print reports of customer orders 3](#_Toc76316383)

[Problem 3: Create sales orders and send them to suppliers in order to satisfy the customer sales orders for the coming month and create lists of items that are required to complete a particular customer sale 4](#_Toc76316384)

[Problem 4: Create a daily report of customer orders that have been completed 5](#_Toc76316385)

[Problem 5: Delete customer orders from the system once they have been completed 6](#_Toc76316386)

[Problem 6: The new workflow must have levels of access 7](#_Toc76316387)

[Problem 7: The new work flow system should be able to print information of customer orders at a rate of 15 per hour 8](#_Toc76316388)

[**THE CHALLENGE OF MOVING FROM TRADITIONAL TO AGILE** 9](#_Toc76316389)

[References 11](#_Toc76316390)

# **FEASIBILITY CRITERIA**

Rather than having an overarching scope, requirements, approach and evaluation, as I suggested using the agile scrum approach, I will break down each issue and then assess each issue's scope, requirements, approach and evaluation. IFR belts seem to solve some basic problems that:

* Receive information of customer orders
* Print reports of customer orders
* Create sales orders and send them to suppliers in order to satisfy the customer sales orders for the coming month
* Create lists of items that are required to complete a particular customer sale
* Create a daily report of customer orders that have been completed
* Delete customer orders from the system once they have been completed
* The new work flow system should have the following levels of access:
* Report and update - for the Assistant Stock Controller
* Report, update and delete - for the Stock Controller
* The new work flow system should be able to print information of customer orders at a rate of 15 per hour

As such, this feasibility report will address each topic and then assess its scope, requirements, approach and evaluation. At the end it will be discussed how difficult it is to go from traditional to agile.

## Problem 1: Receive information of customer orders

This task has to be carried out by the Assistant Stock Controller, and he has only report and update permissions. The approximate time taken to develop a solution for this problem should only variate from 2 to 5 working days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This part of the system is focused on notifying assistant stock controller of customer orders, as such this part of the solution must result in a fully functional notification system. | Database put in place to store customer orders in order to retrieve them.  Appropriate permission hierarchy set in place to ensure assistant stock controller is capable of receiving information of customer orders. | Start with designing the database, then test the design with test data, then proceed to implement the database.  String or Integer based permission hierarchy can be set in place, this may be stored in memory or in the database. | This will reduce the necessity of stock controllers having to go through multiple documents, which is a strenuous task.  All information will be delivered as they happen (real-time). |

## Problem 2: Print reports of customer orders

This task is also performed by the Assistant Stock Controller. The estimated time required to develop a solution to this issue should be as little as 1 to 2 business days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This part of the system is printing reports, so the solution must be capable of syncing with printers. This is a print feature, either as soft copy or as hard copy. | Database put in place in order to retrieve customer orders. | First, the data must be formatted properly (prettified) so that it’s humanly readable,  Then use language native approach for printing soft-copies and synchronization with printers to print hard copies. | This will allow the assistant to send reports of customer orders to other parties. This is a low-priority problem. |

## Problem 3: Create sales orders and send them to suppliers in order to satisfy the customer sales orders for the coming month and create lists of items that are required to complete a particular customer sale

These 2 tasks are performed by the inventory controller and assistant respectively, the inventory controller has excessive permissions that can modify the data (remove permissions), so when creating the solution, make sure to provide confirmation windows for actions that modify the data. The time required should range from 4 to 9 business days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This part of the system is a creation feature. Extremely caution must be applied so as to not overwrite/delete existing data. | Database put in place in order to store customer orders.  A graphical interface to create an abstraction level over data storage, so that the stock controllers can store data easily. | Start by designing the graphical interface, then do simple tests and verify the design’s functionality.  After which it is possible to write the business logic. Precautious steps have to be taken. | This feature will completely overhaul the necessity of writing sales orders, and vastly decrease human error.  This is a high priority feature, as customer sales orders satisfaction monthly and list completion depends on this feature. |

## Problem 4: Create a daily report of customer orders that have been completed

This task is performed by the wizard; Precautions must be taken not to overwrite existing reports. The time required should range from 2 to 3 business days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This is yet another creation feature, but must be done daily so use the notification system from problem #1 to notify the assistant to make a daily report. | Database put in place in order to store customer orders.  A graphical interface to create an abstraction level over data storage, so that the stock controllers can store data easily. | Start by designing the graphical interface, then do simple tests and verify the design’s functionality.  After which it is possible to write the business logic. Precautious steps have to be taken. | This feature can be automated, if the clients will cooperate it is best to automate this feature.  Daily reports don’t contain a lot of information so, it’s a low priority feature. |

## Problem 5: Delete customer orders from the system once they have been completed

This task is performed by the inventory controller, who has deleted rights. Take the necessary precautions to receive confirmation from the user before deleting orders. The time required should be 1 to 2 business days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This is a deletion feature, completed customer orders must easily be separable from those that are not. | Database put in place in order to store customer orders.  Backup system and a logging feature as failsafe procedures. | Start by ensuring that there is a distinct way to separate completed orders from incomplete orders, then implement the delete feature.  It is recommended to log each delete to a file including what was deleted and when, by whom. | This feature can only be implemented after the previous features have been implemented in order to ensure what’s necessary has already been put into place.  As such this is a low priority feature. |

## Problem 6: The new workflow must have levels of access

This is a passive function that is required for all other functions, it is the highest priority task after creating the database. All warnings and traps should be carefully considered. The time required should be 3 to 5 business days.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This is a permission hierarchy, the foundation for all other features. | Database at least in design stages. | Understand the 2 levels of access, assistant controller and controller, each have their own permissions.  Use bitflags to generate a permission sequence that will resolve to a particular set of permissions for a person. | This is a high priority feature, and must not be erroneous at all. Do appropriate testing to ensure that the permission hierarchy is not breathable. |

## Problem 7: The new work flow system should be able to print information of customer orders at a rate of 15 per hour

This is also a passive feature that may or may not come after the permission hierarchy. Resource allocation and memory management will be two groundbreaking points to focus on during this track. Performance and memory profilers will be your greatest ally.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Requirements** | **Approach** | **Evaluate** |
| This is a rate limiting feature, it must be made sure that only 15 prints can be made per hour and that this applies to all users of the system. | Database put in place in order to test the rate limiting feature.  A logging system. | The logging system will be used to ensure the rate limiting. Each time a print is made, the time must be logged and a global counter must be incremented.  Once the counter has reached 15 then time must be checked in order to ensure whether 1 hour has passed or not. | This is a low priority feature and can be implemented the last if it has to be.  As it does not affect the workflow directly. |

# **THE CHALLENGE OF MOVING FROM TRADITIONAL TO AGILE**

When your projects are behind schedule and the results aren't up to the standards you hope to achieve, it's time to take a closer look at your project management approach to see if it's holding you back. Old traditional project management methodologies can be cumbersome and inefficient, and if you're still using them, it's very likely that a newer and better approach like agile could be a game-changer for your team.

A sudden switch from traditional project management to agile can be a tricky event, especially if your team members are unfamiliar with it and need to learn a whole host of new technical terms. Seamlessly reconciling your old project management methods with an agile framework requires flexibility, but it's worth it

Although agile started out as an approach to software coding, the sprint-based structure uniquely translates into project management.

As such, I believe that during this transition there must be intensive training and proper orientation, that the company must invest in educating its members to understand the new concepts and deal with them one by one.

That is why it is important to choose the right agile approach when moving from a traditional approach. Scrum is the best option if you are moving from a traditional approach to an agile one because the greatest strength of scrum is that it emphasizes teamwork and communication rather than learning a lot of heavy words or starting a new process.

Everyone knows how to communicate in the right way and to communicate efficiently, that's what Scrum does. Despite the traditional approach that the IFRB has taken before, this transition procedure will be the most ideal as the responsibilities will be brought together.

Why is this ideal? However, because it is our human nature to never intentionally jump into a fire pit, although it is also our human nature to jump into a fire pit when many other people jump into it too. What I mean by this is that when responsibility is shared, each individual is under less pressure, expectations don't come individually, they come as a team. Each individual will come to know their role and contribute in inappropriate ways. If they make a mistake, responsibility will be shared at the next Sprint meeting.

The transition from traditional to agile seems very difficult at first, but after starting the transition you never want to stop. Once you become agile, you can never go back. It's so great. (WEB DEVELOPMENT BY ONCE INTERACTIVE, 2020)

# References

WEB DEVELOPMENT BY ONCE INTERACTIVE, 2020. *executiveoption.* [Online]   
Available at: https://executiveops.us/challenges-of-transitioning-from-traditional-project-management-to-agile/  
[Accessed 18 June 2021].