#BIKES4ERP Change Request

Virtual Visionaries
We See What You Don't

INF 370 - Deliverable 9

10/20/2020

Byron Donald
u17237913

Hansen Li
u17073708

Group Leader
Tapiwarufaro Ndoro
u17210918

Cameron King
u17165904

Gareth Phillips
u17004162

A MEMBER OF: groupelephant.com™
BEYOND CORPORATE PURPOSE

ERP™

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document serves as the starting point of our Deliverable 9. The Deliverable as a whole, seeks to serve as an extensive and comprehensive User Documentation for all those who will participate as users of our system.

This document sets the pace by laying out the change request procedure for our Client to follow if they are ever to be in a position where they require certain changes to be done. It is important to note that certain changes can have an adverse effect on some core system functions, and these are worth noting as one carries out this process.

The change request flowchart will graphically depict the steps to be followed. The change request form is the document that will serve as the paperwork detailing the reason and details of the change request desired by the client. If the change request procedure seems unclear at any point, the client is to contact any one of the team members with their issues for clarity.

2 Change Request Procedure

2.1 Introduction

At any point in the product life-cycle, if our client is to ever be in a position where they require certain changes to be done, this section will breakdown the step-by-step guide as to what they should do to get a hold of the team. Thereafter, the changes will be made and will be implemented. This is what this section will be covering.

2.2 Procedure/Process

Step	Details
i	The Client fills out the change request form with all the details of the desired change indicated clearly.
ii	After internal approval, the client submits the change request form to the team, either via email or hand delivers the document. The form will indicate the level of urgency.
iii	The project team evaluates the suggested changes, and analyses the impact in performance, functionality, and usability in light of the depth of the change. Their availability and ability will also be considered in this analysis.
iv	The analysis will present the team with objective information to decide whether the changes are feasible or not (time, cost, functionality, scope). The team will notify the client of what they propose as a counteroffer of changes, showing what the outcome might be and the effects, as well as availability. Additional comments will be made as well, including the impact/risk assessment.
v	The Client or representative signatory will decide whether the offer is satisfactory, adequate, and worth proceeding with. They should consider all the risks comments made in this offer. They will sign-off on this agreed-upon set of work and the deadlines therein.
vi	The team makes the necessary documentation changes to the system design and make the development changes to a separate version of the system.
vii	The team gets to work within the schedule and budget constraints to deliver the expected changes to the client. Extensive testing will be done. All related changes to the User & training Manuals will be reflected accordingly.
viii	Upon completion, the team signs-off on the product and hands it over to the Client for sign-off. These changes will be implemented in a "Phased conversion" plan. Starting with different user types in different schools, then working it up to see how stable it will all be. The change request procedure concludes here.

Table 1: Change Request Procedure

2.3 Change Request Flowchart

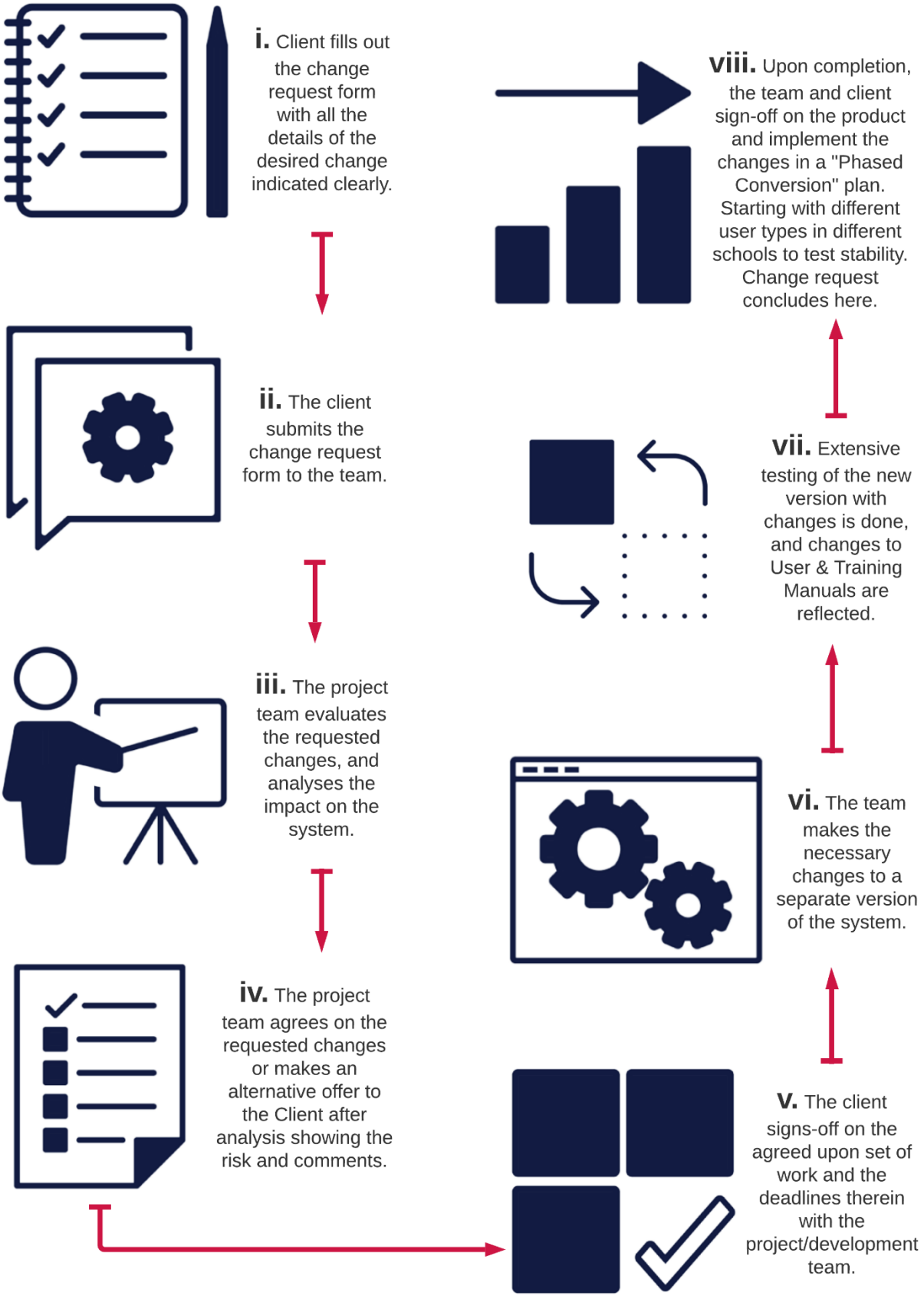

Figure 1: Change Request Procedure Flowchart

2.4 Change Causes, Steps after change implementation

Situations may arise whereby our client's business changes, which affects the business requirements upon which the initial system is built. In this case, once the new business requirements are laid out, a change request can be submitted by following the change request procedure laid out above. Businesses evolve, new developments and innovations come up as the system users interact with it. This also may give rise to possible changes. However, it is worth remembering that changes will come with an impact on the budget, scope, schedule, and additional resources. Further impact may be on system stability or core functionality. All these need to be weighed properly and thoroughly while looking at the system holistically.

After the change is implemented, there is then a report which is produced which documents the changes made, risks involved, and additional comments worth mentioning about what may arise as a result of the change.

2.5 Conclusion

This section of this document has laid out the underlying reasons that could give rise to the need for a change request procedure. Further to that, a step by step procedure to be followed when the need arises has been shows. The flow diagram shows it graphically in a summarized manner, giving color to the process, helping the Client visualize the process a lot more clearly.

3 Type of Change

3.1 Introduction

There are various changes that a client could request for. To make it easier, we have categorized these into sub-groups of "Change Types". These categories of types of change are available on the Change Request Form. Knowledge of what they cover helps to know what to select or indicate when carrying out a Change Request Procedure.

3.2 Change Types

Change	Details
Hardware	Since we integrate hardware like BLE Antennas and QR Scanners into the system, if there is an improvement needed to accommodate new and improved hardware alternatives, it would be a change of this type.
Functionality Enhancements	If something changed in the business requirements that need a change in the way certain system functionalities happen, it would be a Functionality enhancement change being done.
Design	This is all aesthetic changes to the system. i.e., changing color schemes, layouts, flow of screens in the system, etc. It is purely how the system is styled, designed, and structured with no tight link to functionality.
Bug Fixes	In cases where certain bugs or errors remain persistent even after troubleshooting with the help manual and user manual, the system might need to get checked by the development team.
Other	All additional System changes that do not fall within the above-specified categories, fall under this. A detailed description of the type of change that needs to be done will need to be indicated within the form.

Table 2: Type of Changes

3.3 Conclusion

This section has clarified the major types of change that can give rise to a change request procedure to be made. With this description and explanation, the Client is now in a better position to know what category of classification their changes fall under so that they indicate accordingly on the Change Request Form.

4 Change Request Form & Report

4.1 Introduction

This section is going to be showing the change Request Form that will be completed by the client and the Change Request Report that will be filled by the Team lead on the system developers' side. This is some of the required paperwork when carrying out a Change Request Procedure.

4.2 Change Request Form

Below is the form that the user will fill in to have a Change Request processed by the Development Team. The Client must aim to fill in as much detail as possible.

Virtual Visionaries
We See What You Don't

CHANGE REQUEST FORM

This form is to be completed by the Client upon the request for a change to the System.
Please Complete this Form in as much detail as possible.
Contact any one of the Virtual Visionaries if you need assistance.

Client Details							
Title: *(Tick ✓)*	Mr. ☐	Mrs. ☐	Ms. ☐	Dr. ☐	Other: ☐	Position:	
Name:						Surname:	
Company:						Request No.:	
Email Address:						Tel No.:	
						Cell No.:	

Request Details							
System Name:						Version No.:	
Change Type: *(Tick ✓ All that Apply)*	Hardware ☐	Functionality Enhancement ☐	Design ☐	Bug Fixes ☐	* Other ☐		
Description *(*For Other)*							
Description for Change:							

Motivation for Change:				
Additional Details:				
Sub-Systems Affected: *(Tick ✓ All that Apply)*	User ☐	Mechanic Schedule ☐	Tracking Hardware ☐	
	Student ☐	Bicycle ☐	Bicycle Part ☐	
	Reporting ☐	Notification ☐	Administration ☐	
	Tracking ☐	Job Tasks ☐	School ☐	Job ☐
Description of Effect on Sub-System(s):				
Sign-Off:	Client Signature: _____		Date: _____	

Figure 2: Change Request Form

4.3 Change Request Report

Below is the form that the Team Lead of the development team will fill in to have a Change Report done that will be sent back to the Client in line with their Change Request. This report must be as exhaustive as possible in terms of showing all risks involved and comments.

Virtual Visionaries
We See What You Don't

CHANGE REQUEST REPORT

This form is to be completed by the Team Leader of the Virtual Visionaries after analyzing the Client's Change Request Form.
Please Complete this Form in as much detail as possible.

Team Leader's Details							
Title: *(Tick ✓)*	Mr. ☐	Mrs. ☐	Ms. ☐	Dr. ☐	Other: ☐	Date:	
Name:						Surname:	
Company:						Request No.:	
Email Address:						Tel No.:	
						Cell No.:	

Report Details					
System Name:				Version No.:	
Priority of Change: *(Tick ✓ One)*	Critical ☐	High ☐	Medium ☐	Low ☐	
Impact (Area) of Change: *(Tick ✓ All that apply)*	Budget ☐	Scope ☐	Schedule ☐	Performance ☐	* Other ☐
Description *(*For Other)*					
Description of Impact:					

Possible Risk on Sub - Systems & Functionality:			
Additional Comments:			
Proposed Solution: *(Tick ✓ One)*	Approved ☐	Counter-Proposal ☐	Rejected ☐
Details of Solution:			
Sign-Off:	Team Leader: _____	Date: _____	
	Team Member: _____	Date: _____	
	Team Member: _____	Date: _____	
	Team Member: _____	Date: _____	
	Team Member: _____	Date: _____	
	Client Signatory: _____	Date: _____	

Figure 3: Change Request Report

4.4 Conclusion

Upon grasping the contents and layout of the Forms above, both parties involved should be able to communicate effectively: The client can state clearly and fully what they want done, and the team responsible for making the changes can report on all the possible implications, resources and possible solutions going forward. The Client can sign off on these proposed solutions, and the changes can be made and implemented.

5 Configuration Management Principles

5.1 Introduction

Configuration Management principles are a process that is used in the implementation and maintenance of consistency of several attributes of a new system. The system attributes that can be under consideration here are Functional requirements, stability, and performance. It becomes easier to track changes and pickup where problems arose. This is directly applicable here because a System Change would have just gotten request by the Client. These Configuration Management Principles ensure that the system quality stays on an upward trend. When Configuration Management principles are applied using effective practices, return on investment is maximized and product life cycle costs are reduced. The small investment in resources necessary for effective Configuration Management is returned many-fold in cost avoidance.

5.2 Principle 1: Protect Critical data and other resources

Data that was already contained in the system database must be kept safe and secured. It must not be lost during the implementation of a change. A system change merely serves as an update or an upgrade, depending on the depth of the changes implemented. You would not want to have to redo data capturing from scratch after losing data. To make sure of this, there must be a data backup procedure that is done before the system changes get rolled out, so that a stable version stays in place.

5.3 Principle 2: Automatic processes and procedures when cost-effective

Wherever possible, some changes should be rolled out automatically wherever possible. The less manual intervention, the better. This avoids the risk of human error when interacting with the system in the state of change or production. In some cases, it's cheaper to go manual. In that case, this principle cannot be fully observed. But wherever applicable, it must be considered.

5.4 Principle 3: Provide value to customers

Every change that is to be implemented, must always keep the customer in mind. The customers' perception and experience are always very important to the system. It must always remain usable and give value to the user in a way that they can perceive their attained benefits to be greater than their investment or cost to use. In our case, this system needs to continually improve business operations at every turn. Every change that gets implemented, adds on to this and not subtract.

5.5 Principle 4: Software artifacts should have high quality

The product should remain operating at a high level of efficiency and have no unnecessary errors and bugs. Everything must continue to streamline and connect smoothly and seamlessly. This includes the scenarios where multiple versions are being maintained, multi-platform systems, as well as standalone one platform only implementations.

5.6 Principle 5: Monitor and control software development procedures and processes

Every process in the development and implementation process needs to be watched closely. The monitoring thereof gives us a rich window of opportunity to check out loopholes and areas that can be taken up for improvement. Every one of the software development processes must not run independently and in isolation. They must be watched as they run.

5.7 Conclusion

This is the end of the brief run-down of the intersection between Configuration Management Principles and Change Request Procedure. Everything about the systems operations and improvements, need to be scrutinized under the lens of the above-mentioned principles.

6 Conclusion

Upon reaching this point, the full picture of the Change Request Procedure has been painted fully. Should the client ever wish to add, remove, or make any amendments of any kind to the system, they carry out this process. It ought to be followed strictly. It will also serve as documentation of the changes that the system goes through. This is our proposed most effective way to implement a change in our system.