Faveo User Manual

Version 2.0

Email: Helpdesk@gogh.co.za | Tel: (021) 930 0010 | Fax: (021) 930 0035

Table of contents

Faveo User Manual Version 2.0 1
1 Faveo - A Workflow Communications Tool 4
1.1 Introduction 4
1.2 Access 4
1.2.1 Connecting 4
1.2.2 Login Page 4
1.2.3 Navigation 4
1.3 User groups 4
1.3.1 Vendor User 5
1.3.2 Client User 5
1.3.3 Customer User 5
1.3.4 The Client and Customer structure 6
1.4 Application role based groups 6
1.5 Application Data Security 6
1.6 Maintaining versions of the Product 7
1.6.1 Application 7
1.6.2 Environment 7
2 Main menu options 8
2.1 Requests 8
2.1.1 Search 8
2.1.2 List 9
2.1.2.1 Me 9
2.1.2.2 My Group 9
2.1.2.3 Me and My Group 9
2.1.2.4 All 9
2.1.3 New 9
2.2 Build 9
2.2.1 Build Submission 10
2.2.2 Compile Build 10
2.2.3 Vendor Build Acceptance 10
2.2.4 Ship Build 10
2.2.5 Client Build Acceptance 10
2.2.6 Client Build Deployment 10
2.2.7 Customer Build Deployment 10
2.2.8 Build Acceptance 10
2.2.9 Build Rejected 11
2.3 Workflow 11
2.4 Administration 11
2.5 Reports 11
2.6 Help & Support 11
2.7 Logoff 11
3 Working with Requests 12
3.1 Referencing Requests 12
3.1.1 Faveo Reference Number 12
3.1.2 Client Reference Number 12
3.1.3 Reference to a prior Request 12
3.2 Capture a New Request 12
3.3 Edit a Request 13
3.4 Request Details 13
3.4.1 Client 13
3.4.2 Customer 13
3.4.3 Scheme 13
3.4.4 Application 13
3.4.5 Environment 13
3.4.6 Functional Area 13

3.4.7 (Request) Type 13
3.4.7.1 Support/Maintenance 14
3.4.7.2 Defect/Fault........ 14
3.4.7.3 Enhancement........ 14
3.4.7.4 Project 14
3.4.8 Defect Severity........ 14
3.4.9 (Request) Priority 15
3.4.10 Client Acceptor........ 16
3.4.11 Reference to Prior Request........ 16
3.4.12 Client Reference 16
3.4.13 Expected Date........ 16
3.4.14 Update Training Manual 16
3.4.15 Build Version........ 16
3.4.16 Short Description........ 16
3.4.17 Requirement (of Full Description)........ 16
3.4.18 Optional Attachment 16
3.5 Advance Search........ 17
3.5.1 Selection by Elements........ 17
3.5.2 Search Output........ 18
3.6 Action Request 18
3.6.1 Action Request Header 18
3.6.2 Action Request Attach Manager 18
3.6.3 Action Request Log........ 18
3.6.4 Logging 19
3.6.5 On Hold........ 19
3.6.6 Rules when Cancelled or Completed........ 19
3.7 Request Status 19
3.7.1 Customer Submission 19
3.7.2 Client Submission 19
3.7.3 Vendor Submission 20
3.7.4 Customer Manager Approval........ 20
3.7.5 Client Verification 20
3.7.6 BA Investigation........ 20
3.7.7 Client Approval........ 20
3.7.8 Vendor Decision 20
3.7.9 Solution Specification........ 21
3.7.10 Solution Specification Approval 21
3.7.11 Quote Acceptance 21
3.7.12 Quote Investigation 21
3.7.13 Awaiting Development 21
3.7.14 Development 21
3.7.15 Vendor Pre-build Testing........ 21
3.7.16 Client Pre-build Approval........ 22
3.7.17 Client Pre-build Testing 22
3.7.18 Client Development 22
3.7.19 Client Development Testing........ 22
3.7.20 Development Testing........ 22
3.7.21 Build 22
3.7.22 Client Acceptance........ 22
3.7.23 Client Testing........ 23
3.7.24 User Acceptance Testing 23
Addendum A: Request Workflow 24
Addendum B: Build Workflow........ 25

1 Faveo - A Workflow Communications Tool

1.1 Introduction

Faveo is an internet based software application which was built and is maintained by ***PBT insurance technologies***.

It is used to manage multiple streams of development work requests from the time one is logged until it is signed off as complete. It records each of the steps a request goes through during its lifecycle together with any comments entered for later reference. It is geared to improve communication between the producer of a product – who is referred to on *Faveo* as the Vendor, and who makes changes to and provides upgrades for a product - and the users of said product. It allows for up to 2 groups of such users, viz. Client (primary group) and Customer (secondary group).

1.2 Access

1.2.1 Connecting

To access the Faveo application, enter this internet address into your internet browser:
http://faveo.gogh.co.za
It is recommended that you bookmark this location.

1.2.2 Login Page

You must always log into the system before using it. The login page requires a user name and password which will be used to authenticate you on the system. You will be redirected to the List Requests screen upon a successful login. Should you bookmark a particular screen within the application and try to go there directly, the system will pass you to the login page before completing your navigation request. It is not recommended that you bookmark pages within the application. Only bookmark the home page.

1.2.3 Navigation

The top of each screen displays the application banner that includes information on which user you are currently logged in as on the right hand side. Below this is a row with a list of links that will enable you to go to different areas within the application.

1.3 User groups

All users on *Faveo* must be identified as belonging to one of the 3 possible groupings listed hereunder.

Users are registered within their specific group by a Vendor Administrator, and their functions will be controlled by means of the responsibilities ascribed to them during registration, or at any later time when their roles may be changed.

A Vendor Manager will approve the registration of Vendor Users, together with their authorised functions, while a Client Manager must approve the same for all Client, Customer or Scheme Users.

1.3.1 Vendor User

The company who maintains the product is called the Vendor. The Vendor is the primary user of *Faveo* and uses it to manage requests through to finalisation.

A Vendor User submits requests on behalf of the Vendor, Client or Customer, where these requests are typically internal modifications aimed at improving the performance and stability of the application. These requests are recorded so that the Client and Customer have full knowledge of all the changes that are delivered in an Application software release. No change impacting the business will be submitted via a Vendor User.

The *Faveo* Administrator belongs to the Vendor User group.

1.3.2 Client User

The business entity, which deals directly with the Vendor regarding issues relating to the product, is called the (Vendor's) Client.

It may be that the Client is not the end user of the product, but acts as an intermediary between the Vendor and one of its Customers.

A Client User is a Client employee who submits or manages requests on behalf of the Client or its Customer. It is possible that more than one of the Client's Customers might be using the same application using a common base of code. The Client must ensure that changes to an application must benefit all their Customers that are using that application.

It is the responsibility of the Client to ensure that all requirements for requests are understandable by the Vendor in terms of business logic, and that all clarifying documentation has been captured on *Faveo*.

1.3.3 Customer User

Any other business entity which is the end user of a specific version of the product and is supported by the Client is referred to as the Customer.

The Customer uses the product to manage the business affairs of one or more Schemes.

A Customer User is a Customer employee who submits requests on behalf of themselves. These requests must be approved by the Client, before it will be actioned by the Vendor. These requests might also apply to other Schemes. The Client must ensure that this is checked, and that Requests are logged accordingly.

A special type of Customer User can also be a Scheme employee who acts on behalf of the Customer and submits requests for one specific Scheme only.

1.3.4 The Client and Customer structure

The Vendor deals directly only with Clients, who in turn may have their own Customers.

iMed is capable of supporting multiple concurrent Schemes on the same application. A Customer administers one or more Schemes.

As an example, the table below explains one structure for the Vendor, *PBT it*. Here, Cumulus is the direct Client of *PBT it*', and they in turn have 2 Customers in Eternity Health and Resolution Health. Eternity administers 3 Schemes as listed, while Resolution Health only does their own.

Client	Customer	Scheme
Cumulus IS RSA	Eternity Health	CAMAF
		Rand Water
		Pure Health
	Resolution Health	Resolution Health

Faveo is capable of dynamically coping with changes in this structure.

1.4 Application role based groups

The approval process must progress as fast as possible and should not come to a standstill because of the unavailability of a single individual. For this reason the system behaves in the following manner. When the status of a request changes all users who are able to action a request at that level will see the request appear in their group based requests requiring action. The user must first take ownership of the request before performing any other action of the request. Once ownership of the request has been taken, the request will no longer appear in the group listing. In other words the group mechanism exists so that a pool of users performing a particular function in the system can check for and action requests. This addresses the issue of the process breaking down when a single individual is not available to action a request.

1.5 Application Data Security

Application security as far as data is concerned is maintained at client level. This means that a client and all the customers for that client will be able to view all their common requests. The workflow approval lifecycle determines who may edit, approve or cancel a request.

It is not possible for a client or a client's customer to view any information for a different client!

The vendor naturally has access to all the data.

1.6 Maintaining versions of the Product

In the above example, the product is a Healthcare Administration System, called ***iMed***, which has been built, and is also supported and maintained by *PBT it*.

1.6.1 Application

The Vendor may support multiple versions of the software product used by various Clients and Customers, situated on different servers, each called an Application. Different Applications will have different names to avoid confusion.

1.6.2 Environment

The physical computer (server) on which a version of an Application has been installed is referred to as an Environment.

2 Main menu options

The main options for use on *Faveo* are situated in the top line menu just under the logo. Those that are greyed out means that the current user is not authorised to use them.

By holding the mouse over an option its background colour will change and a dropdown selection appears of functions that can be executed for that particular option. Clicking with the mouse selects the function. Again any function greyed out are not authorised for the current user.

For **Requests** and **Build**, the dropdown is presented as 3 sections, each split from the one above by a solid line.

a) In the top section the functions for any Request or Build are shown.
b) If the user has already selected a particular Request or Build, then that item's identifying number, together with the functions available for that selected item are shown in the middle section.
c) The 3rd section shows the previous 10 items last worked on, and can be reselected from here.

Similar as above, any function greyed out are not authorised for the current user.

Note that where lists of Requests or Builds are displayed, the column headings in the output can be clicked on to change the **sort order** of the list. Clicking once orders the results by the selected column in **ascending** sequence. Clicking twice on a column will do a **descending** sort on that column. This is a **one dimensional sort** meaning that you cannot sort on multiple columns.

2.1 Requests

A Request is a singular piece of work which the Vendor is asked to perform in the support of running the Scheme's affairs. It gets logged on *Faveo* and is to be done on a specific version (application) of the product.

A Request should be broken down into only one unit of work to be able to manage its resolution from investigation through preparation, testing and conclusion in a controlled manner.

The following functions can be selected to be done on Requests.

2.1.1 Search

Faveo allows a User to **Search** for a specific Request using various search criteria. When Search is selected, an Advanced Search screen opens up. Composite search criteria are allowed.

Refer to paragraph **3.5 Advance Search**.

2.1.2 List

A User can also select to **List** Requests, and is presented with a choice of 4 different lists. The List screen has buttons called “Me”, “My Group”, “Me and My Group” and “All”.

2.1.2.1 Me

The “Me” section lists all requests requiring the user’s attention in the workflow process. This is the default setting.

2.1.2.2 My Group

The “My Group” section lists all requests requiring attention from role based groups of which you are a member. These Requests have not been assigned to any user, and any user within this particular user group can access them for further action.

Once you take ownership of the request it is transferred into the “Me” section and disappears from the “My Group” section for you and all other users that could see it there.

2.1.2.3 Me and My Group

This is a combined list of the two above.

2.1.2.4 All

This list presents the user with all the captured Requests that User has been authorised to view.

2.1.3 New

A User may also log a **New** Request when required.

For more details on Requests refer to paragraph **3 Working with Requests**.

2.2 Build

A Build is the release process where one or more Requests are packaged together and sent as a change to the current application or an update to the database.

There are 5 different types of builds, viz.

(a) Delta = all changed procedures plus a full front end,
(b) Emergency Bug Fix (EBF) = an urgent fix for procedures that are incorrect,
(c) Interim Release (IR) = new and/or changed procedures for enhancement to the current application,
(d) Data Fix (DF) = scripts to add or amend data to correct transactions in error,
(e) Data Load (DL) = scripts to add new data into the database.

A Build has 8 steps, which are:

2.2.1 Build Submission

Only a Vendor manager will authorise the creation of a **New** Build and group relevant requests to it, as well as preparing and attaching the Release Notes. He then actions the Vendor DBA.

Requests must be at Development Testing status to be selected.

2.2.2 Compile Build

Vendor DBA packages the release and then takes it to the next step by deploying the release into a QA environment.

2.2.3 Vendor Build Acceptance

QA is done before it is sent back to the DBA for shipment.

The status of Requests in the build is automatically changed to Build.

2.2.4 Ship Build

The Vendor DBA will ship the release, i.e. place it on a FTP site that is accessible to the Client DBA. Then the build is marked for Client Build Acceptance.

All Requests statuses are automatically advanced to Client Acceptance.

2.2.5 Client Build Acceptance

The Client manager must select from the **List** of Builds which he will Accept or Reject. Once the Client Manager has decided to accept a build, he will request the Client DBA to deploy it into a Client test environment where each request is to be tested.

All Requests statuses are automatically advanced to Client Testing.

2.2.6 Client Build Deployment

Only when all the requests have been tested successfully, can the build be accepted and deployed to the Customer environment. The Client Manager then takes the Build back to Client Build Acceptance and instructs the Client DBA to deploy it to the Customer’s test environment.

All Requests statuses are automatically advanced to Client Acceptance.

2.2.7 Customer Build Deployment

A similar process of testing takes place at the Customer’s test environment. Only when all the requests have been tested successfully, i.e. their status changed to Complete, can the build be accepted and deployed to the live production environment. The Client Manager then instructs the Client DBA to deploy it to the Customer’s production environment.

2.2.8 Build Acceptance

The Client Manager then marks the build as accepted, thereby closing it.

All Requests remain at Complete.

2.2.9 Build Rejected

When a build is rejected, all the Requests in it are automatically returned to the Vendor in the status of Development Testing to make them accessible for further work.

2.3 Workflow

This option is for use by the Client manager and Vendor manager only. Its sole purpose is to indicate which Requests are to get preference in the work schedule.

Each new list overwrites the previous one.

An upload from Excel is available where only the priority sequence number and the Faveo Reference Number are required to be copied.

2.4 Administration

This option is used by the Vendor only.

It is used by the Vendor Administrator to set up Users or modify their details.

It is also used to investigate email logs or event logs to ensure that *Faveo* is functioning correctly as an application.

2.5 Reports

The only Report currently available is a summarised breakdown presenting the number of Requests per Status for each registered User.

2.6 Help & Support

Currently the User Manual and Vendor contact details are available here.

2.7 Logoff

This Logoff must be used by all Users to properly close their sessions.

3 Working with Requests

3.1 Referencing Requests

For ease of identification of Requests, Faveo provides for the storing of 3 different reference numbers.

3.1.1 Faveo Reference Number

This is an Application generated reference number which is automatically created when the Submit button is activated during the initial capturing of a Request. This number is unique within the system and the numbering is maintained at Client level only. It is made up of a combination of 3 letters, which is an acronym to identify the Client, followed a sequential number of 6 numerics.

3.1.2 Client Reference Number

This is an external number which can be captured on Faveo, and by which the Client, Customer or Scheme can identify a Request. (The standard format, if a RFS number from *iMed* is used as reference, is RFSnnnnn, where n denotes a numeric value between 0 and 9.). This number is optional.

3.1.3 Reference to a prior Request

This field is also optional and allows one to link a Request to any relevant previous one in order to facilitate understanding of requirements.

3.2 Capture a New Request

The new request Capture Screen has many fields. It is recommended to complete as many of the fields as possible, because a comprehensive request will result in fewer mistakes and will provide a quicker turnaround, because all the required information is immediately available. The Capture Screen has many dropdown lists and it is recommended that you complete them from top to bottom. Changing a dropdown can result in other following dropdowns being reset because the options they display may be dependent on options selected in previous dropdowns.

The fields with an asterisk (*) in the label are required fields. You will not be able to save the request until all these fields have been entered.

If you are not sure what information to capture in a specific field just hold the mouse over the input area for that field. A tooltip will be displayed giving you information about that field.

Once you have completed filling in the fields, click the submit button. If all validation passes the request will be saved in the system and the saved request will be displayed on the screen.

At this point you must take note of the system generated *Faveo* Reference number.

3.3 Edit a Request

Clicking the edit button allows you to modify the details of the request. Please note that once the request commences down the workflow and approval process, you will not be able to change the request. Ensure that all details are correct before submitting for approval.

3.4 Request Details

3.4.1 Client

Selected from dropdown options.

3.4.2 Customer

Selected from dropdown options.

3.4.3 Scheme

Selected from dropdown options.

3.4.4 Application

Selected from dropdown options.

3.4.5 Environment

Selected from dropdown options.

3.4.6 Functional Area

In support of and for ease of maintenance of a Scheme's affairs various functions can be grouped into one or more Functional Areas, where each contains specific components supporting one aspect of the Scheme's business. Examples could be such as Products, Membership, Claims, Finance, Interfaces or Correspondence, to name a few.

Selected from dropdown options.

3.4.7 (Request) Type

Selected from dropdown options.

Requests are seen as belonging to one of the following four types:

3.4.7.1 Support/Maintenance

This is a Request to provide assistance to a Client or Customer in order to maintain the parameters required by an Application for proper execution.

3.4.7.2 Defect/Fault

This is a Request highlighting a shortcoming or imperfection found in an Application which causes an incorrect processing result.

3.4.7.3 Enhancement

This is a Request for a change to current or additional new functionality which is required to by an Application and is intended to enhance the capability of that Application.

3.4.7.4 Project

This is a Request for a task that is part of, or is to be managed as, a formal project with other Requests. Various Requests can thus be grouped as belonging to one Project.

3.4.8 Defect Severity

Selected from dropdown options.

The level of severity indicates the impact of a defect on the system and should be recorded in accordance with the following table:

Severity	Definition
1 - Critical	A critical, high impact defect which would create major disruption to the business operation, for example: ▪ A failure in one application affecting other downstream systems ▪ A severe application problem causing considerable down time, financial penalty or loss of integrity with customers ▪ Failure to meet very basic customer service levels.
2 - Major	A major defect, which would result in loss of business functionality and would require a work around in production. Defect impact would be contained within originating system, but would create accumulative errors, for example: ▪ Incorrect data can be entered that truncates data or fails to correctly deliver major business functionality and could potentially cause considerable down time.
3 - Medium	A defect which would have a medium level impact to business functions, but could be immediately managed or worked around, for example: ▪ No loss of functionality, but presents or processes data in a manner that could lead to errors.
4 - Minor	A cosmetic only defect which would have no business or user

Severity	Definition
	impact, for example: ▪ Misalignment of report headings/screen labels.

3.4.9 (Request) Priority

Selected from dropdown options.

The priority of a Request indicates the level of importance with which the Business views this Request and should be recorded in accordance with the following table:

Priority	Definition
1 - Urgent	This is the highest level priority. Defect : It signifies a problem that has virtually stopped the application from being usable. An important function or process is waiting on this issue to be remedied. I.e. Nothing else can be executed until this problem has been corrected, tested and re-migrated. Enhancement: As a change (or enhancement) to the system that is of the utmost importance to the business.
2 - High	This is a high level priority. Defect: It is secondary only to Critical defects. Although the critical path is impacted may be severely impacted, there are still other processing streams that can be tested while a solution is being developed for this defect. Enhancement: This request is less important than the Urgent priority.
3 - Medium	This is a medium level priority. Defect: One of the multiple streams of processing (other than critical path) is impacted, but other streams of processing can still continue. Enhancement: This request is important to the business but not critical. This functionality is not required immediately.
4 – Low	This is a low level priority. Defect: Users can still proceed to use the application with a work around and the solution can be applied in isolation. Enhancement: Work on this request is not a priority at the moment.

3.4.10 Client Acceptor

Selected from dropdown options.

When a Request solution is released to the Client and is deployed, the Build mechanism automatically assigns the Request for Client Acceptance to the Client Acceptor, if it is filled in. Else it will leave it unassigned.

3.4.11 Reference to Prior Request

Refer paragraph 3.1.3.

Selected from dropdown options.

3.4.12 Client Reference

Refer paragraph 3.1.2.

3.4.13 Expected Date

Only if there is a date where this request must be resolved and the system must be enhanced by, then the appropriate date must be selected here.

This can otherwise be left blank.

3.4.14 Update Training Manual

A senior Client User may see that the resolution of a particular Request highlights a gap in their training manual, whereupon he could flag this Request via the Training indicator to be incorporated into said manual.
This flag is a Yes/No selection.

3.4.15 Build Version

To help the Vendor find to trace the problem, the precise build version in which the problem was discovered should be entered here, else left blank.

3.4.16 Short Description

Compulsory description of the problem in less than 40 characters.

3.4.17 Requirement (of Full Description)

A detailed description can be typed in here. Remember to provide all identifying numbers, such as membership, claim, etc.

3.4.18 Optional Attachment

In addition, any number of documents or files may be attached to a Request at any time to clarify its requirements or to keep a history of decisions or events.

3.5 Advance Search

3.5.1 Selection by Elements

The Advanced Search screen presents the user with a dynamic list of search elements, where either
(a) all possible selections for an element are presented in a dropdown, or
(b) a box for the element where search characters are allowed to be captured into.
In the case of (b) the capture of partial search characters is supported.

The list contains the following elements:

○ *Faveo* Reference Number	capture
○ Client Reference	capture
○ Work Action	dropdown selection
○ Assigned User	dropdown selection
○ Short Description	capture
○ Application	dropdown selection
○ Environment	dropdown selection
○ Functional Area	dropdown selection
○ Assigned To (group listing options of Me, My Group, Me and My Group, All)	dropdown selection
○ Logged Between (date from) AND (date to): a date range for when the request was originally captured	dropdown selection
○ Training Manual	dropdown selection
○ Client	dropdown selection
○ Customer	dropdown selection
○ Scheme	dropdown selection
○ On Hold	dropdown selection
○ Business Requirement	capture
○ Type	dropdown selection
○ Severity	dropdown selection
○ Priority	dropdown selection
○ Originator: the user that initially submitted the request	dropdown selection
○ Work Sequence	dropdown selection, where multiple selections can be made by holding down Ctrl and clicking on the ones required

The Advanced Search uses **AND LOGIC** which means that if more than one element of criteria are specified, both must be true before the request is listed in the result that follows. If no results are returned consider changing your selection criteria. Typical searches will answer the following questions:

- What are all the requests that I submitted?
- What are all the requests that are at a particular Work Action status? Example: waiting for Client Approval.
- What are all the requests submitted between two different dates?
- What are all the high priority requests?
- Combinations of the above.

3.5.2 Search Output

The result of the search is displayed in the application, but can also be redirected to either a single consolidated page, or an Excel spreadsheet.

The single page option allows the user to copy and paste the list into other applications.

The spreadsheet option makes it easy for the user to use the list in Excel, but keep in mind that changes made outside of the system are not reflected inside the system until the user updates the requests accordingly.

3.6 Action Request

The action request screen is the screen that is used to manage a request through its various statuses after it has initially been entered into the system. Typically you find yourself in the Action Request screen once selecting a request from a List or Search Request screen.

This screen has three main parts.

3.6.1 Action Request Header

The header displays a subset of the requests information which can be listed in more detail by clicking on the **Details** button. You can also edit the request by clicking on the **Edit Details** button.

3.6.2 Action Request Attach Manager

The **Attach Manager** button takes you to the Request Attachment Manager screen where attachments may be saved into the system for the current request.

Take care when attaching anything, because once attached, it cannot be deleted.

Versions of attachments are handled automatically within *Faveo* only if the correct process is followed. The process is simple:

a) Open a request and select Attach Manager
b) Mark the document (click on the **Select** sign on the left side)
c) Then click on **Check Out** below the list of documents
d) Save the document
e) IMPORTANT: CHANGE THE NAME TO BE THE SAME AS ATTACHED ON *FAVEO*
f) Edit the document
g) Browse for the newly edited document via Attachment Source from *Faveo*
h) Select and Open it
i) Click on **Check In**.

3.6.3 Action Request Log

The Action Log section lists an audit trail of events for the current request. Here you have two options to put the request on hold or to action the request. When actioning a request you select the next valid status (forwards or backwards) and optionally select a user to assign it to should you not want the request to appear in the group listing.

3.6.4 Logging

The system keeps track of all changes to the workflow of a request as it goes through the lifecycle. This includes instances when a request goes On or Off Hold. This logging is important when trying to determine why a request has taken a certain amount of time to complete.

3.6.5 On Hold

A request can be placed On Hold which suspends the request at the current point in the approval process. When the On Hold status is lifted, the request continues at its previous point in the lifecycle.

3.6.6 Rules when Cancelled or Completed

The status of a cancelled or closed request may not be changed. A new request must be submitted if the request was closed or cancelled and then later needs to be reinstated.

3.7 Request Status

Refer to the diagram in Addendum A to see the full flow of Request statuses. In its life cycle a Request can be passed through various statuses (or steps) en route to completion and different users with different authority roles will interact with the request when at particular stages.

It may go back and forth in certain instances, but for control purposes it will frequently be passed between Client Approval (for a Client manager to be informed) and Vendor Decision (for a Vendor manager to be informed).

The current status on which a Request is sitting indicates the next action which is required. A Request can have forward or back steps depending on the situation.

Each time a request status is changed, it is compulsory to add some valid comments in the appropriate window to improve communication.

3.7.1 Customer Submission

A user with *Customer Submission* authority will be able to log a new request. The request will stay his responsibility until such time as he moves it to the next step. Until then, he may edit the request as many times as he wants to. The next forward step is to *Customer Manager Approval*. The request may also be Cancelled at this stage.

3.7.2 Client Submission

A user with *Client Submission* authority will be able to log a new request. The request will stay his responsibility until such time as he moves it to the next step.

Until then, he may edit the request as many times as he wants to. The next forward step is to *Client Verification*. The request may also be Cancelled at this stage.

3.7.3 Vendor Submission

A user with *Vendor Submission* authority will be able to log a new request. The request will stay his responsibility until such time as he moves it to the next step. Until then, he may edit the request as many times as he wants to. The next forward step is to *Vendor Decision*. The request may also be Cancelled at this stage.

3.7.4 Customer Manager Approval

A user with *Customer Manager Approval* authority will be able to approve requests at customer level to be carried forward. He must ensure the request is valid and carries enough information to be acted upon. He may edit the requirements to achieve this or refer the request back to the submitting user for more detail. To approve a request he will change the status to the next step. The next forward step is to C*lient Approval.*

3.7.5 Client Verification

A user with *Client Verification* authority will be able to verify requests at client level. If he is unable to verify the request himself, he may return it to from whence it came, or forward it for *BA Investigation.* When satisfied the request is valid and in need of further attention, he forwards it to the next step, viz. *Client Approval*.

3.7.6 BA Investigation

A user with *BA Investigation* authority will be able to investigate requests at client level. Once he has achieved what was required, he makes the request ready for verification again by adding information or a message and changing the status back to *Client Verification.*

3.7.7 Client Approval

The user with *Client Approval* authority plays a pivotal role and is the main communicator between Client and Customer on the one side and the Vendor on the other. He will be able to prioritise requests from the Client perspective by allocating a desired Work Sequence number via the *Workflow* option. He may pass back a request from whence it came for further details, put a request on hold or forward it to the Vendor for development. He passes forward a request by changing the status to *Vendor Decision*.

3.7.8 Vendor Decision

Much like *Client Approval,* the user with *Vendor Decision* authority plays a pivotal role, but on the Vendor's side. He will decide what action to take with regards to a request, depending on what stage the request has reached. Apart from sending it back to *Client Approval* with an explanation or for further detail, he will forward it elsewhere, depending on the stage the request has reached. Initially it will go forward to a Designer for *Solution Specification*. When it comes back, and if it is a new enhancement, he will forward it for *Quote Acceptance,* else he will allocate it to a Developer as *Awaiting Development*. If a *Quote Acceptance* has been approved, he will also allocate it as *Awaiting Development*. If the solution is ready for release, he will forward the request to *Development Testing*. Or if a solution has already been released, he will forward it to *Client Acceptance*.

3.7.9 Solution Specification

A Vendor Systems Analyst/Designer will have this authority. He will investigate the nature of a request and analyse what is required. If he requires further information, he will return it to Vendor Decision with an appropriate message. After the analysis and design stage, he will document his findings in detail in an Impact Analysis (IA) document, which will include an estimated time for development and guidance as far as testing is concerned. Then he will either forward the request for *Solution Specification Approval* by the Quality Manager, or for *Vendor Decision*.

3.7.10 Solution Specification Approval

Either the Quality Assurance Manager or, if the IA was done by a junior resource, then a Senior Systems Analyst can have the authority to this role. The request specification will be compared to the IA to ensure all aspects have been thought of and recorded. He can then either send it back with instructions to further expand on certain issues, or forward it to *Vendor Decision*.

3.7.11 Quote Acceptance

The client user with *Quote Acceptance* authority will have 2 roles to play. First he sends the request to *Quote Investigation*, for a knowledgeable user to scrutinise the IA and give approval, and then when it is returned, he needs to get a Client Manager to agree to pay the hours required for development and implementation before he returns the request. Whatever the outcome, he returns the request then to Vendor Decision with an appropriate message (of acceptance or rejection).

3.7.12 Quote Investigation

A knowledgeable Business Analyst will have the authority for this role. He will check the IA and compare it to the requirement specification to ensure that a complete solution is being proposed. He will return the request to *Quote Acceptance* with an appropriate message.

3.7.13 Awaiting Development

When a request is assigned to a Developer, it will sit on this status until the Developer has time in his schedule to attend to its execution. Then he will forward it to himself by setting the status to *Development.* Alternatively, he can pass it back to *Vendor Decision* to be assigned to another Developer.

3.7.14 Development

After a Developer has started with this stage, he may find that he needs further information or help, and returns it to *Vendor Decision*, or he may complete what the IA stated. When the latter is achieved, he will decide if the changes can be tested in-house or whether it needs to be sent to the client for (pre-build) testing. If the former, he will change the request status to Development Testing, and if the latter to Vendor Pre-Build Testing.

3.7.15 Vendor Pre-build Testing

When a Developer sets a request to this status, he indicates that all changes have been done and the code changes or scripts have been put into the Client test environment for them to help with its testing. The Vendor user will now pass the status to *Client Pre-build Approval* to request the Client user's help in testing.

3.7.16 Client Pre-build Approval

The Client Manager will take the request from here and allocate it to the appropriate tester via *Client Pre-build Testing*. When the request is then returned by the tester, he will send it back to *Vendor Pre-build Testing* with an appropriate message.

3.7.17 Client Pre-build Testing

An appropriate Client tester will test the application to see if the change is successful. He will comment on his findings as complete as possible, with screen shots or the like to prove his point. Then he will return the request to *Client Pre-build Approval.*

3.7.18 Client Development

In very special and isolated instances, the Client may have a Developer to look after a specialised part of the product (eg. Website). Then the Client Approval user will schedule the development in-house and assign a request for Client Development. This piece of development is also to be tested by the Client and thus the status is forwarded to Client Development Testing.

3.7.19 Client Development Testing

A request which have been done by a Client Developer will be tested by a Client tester and if the result is negative, it is sent back to Client Development with an appropriate message. If successful, the request is forwarded to *Client Approval*. If such a request is required to be included in a release, then the *Client Approval* will forward it to *Vendor Decision* with an appropriate message to, in turn, forward it to *Development Testing* for inclusion in a build.

3.7.20 Development Testing

When a Developer sets a request to this status, he indicates that all changes have been done, the code changes or scripts have been checked back in, and the request is ready for the QA and release process. If the request is of an urgent nature with no front end development, he may make the call to have the request be released as an EBF.

3.7.21 Build

When a request is included in a release, the release process itself will automatically take it to this step from *Development Testing*. No user will be able to make any changes to the requests during the build process. Should the build be successful and the software released to the client, the status will be changed to *Client Acceptance.* If the build is rejected, all requests therein will automatically revert back to *Development Testing.*

3.7.22 Client Acceptance

A Client Manager will have the authority to forward a request from this status. He could either send it to a Client user for it to be tested via *Client Testing*, or to a Customer user for *User Acceptance Testing*. If the testing has been successful, the request is marked as *Complete*. On the other hand, if either returns the request with a defect, it will be sent back for *Vendor Decision*.

3.7.23 Client Testing

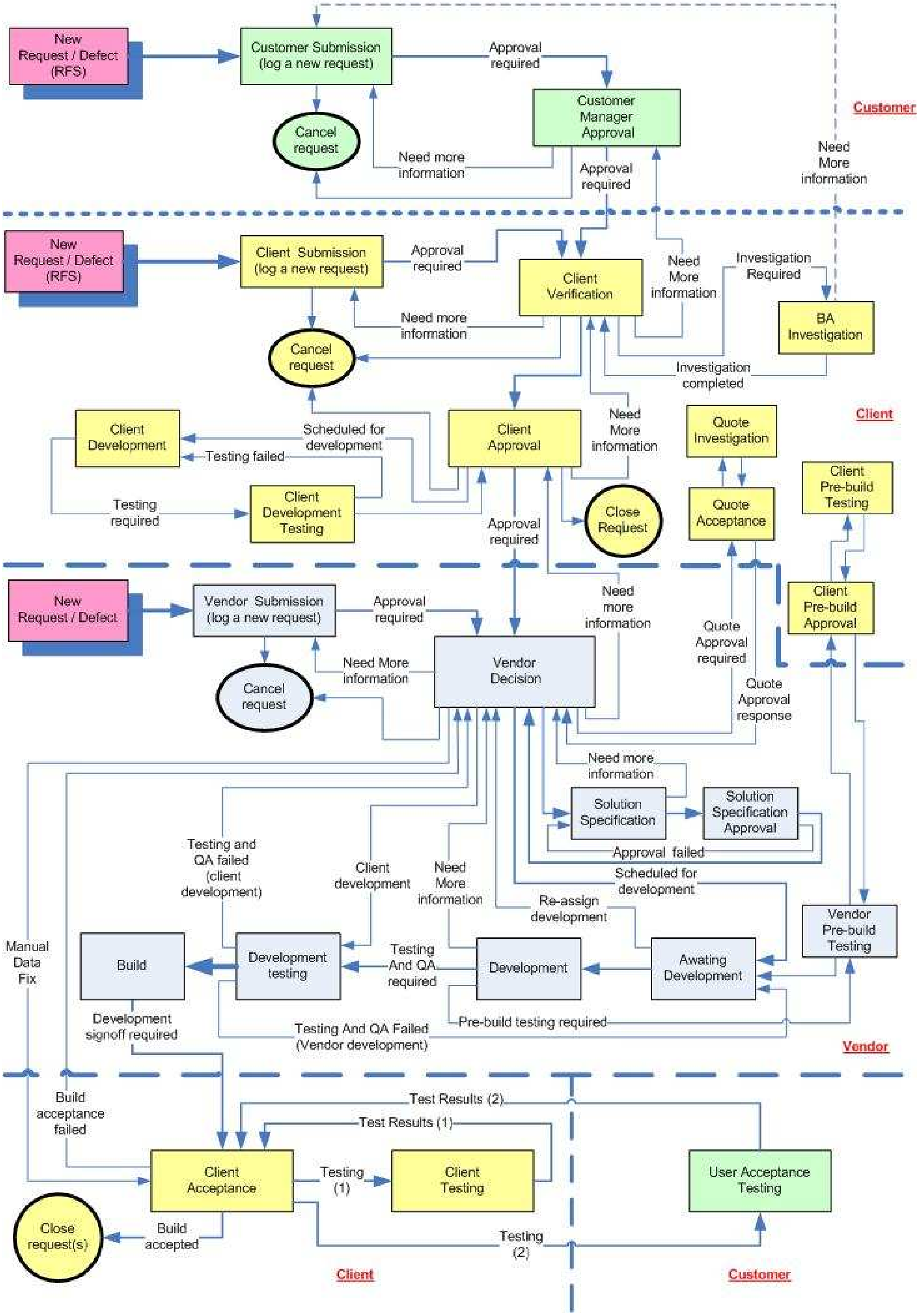
A Client user is required to test the solution of the request in the Client's test environment. Once he reaches an outcome, he returns the request to *Client Acceptance* with an appropriate message.

3.7.24 User Acceptance Testing

A Customer user is required to test the solution of the request in an appropriate test environment. Once he reaches an outcome, he returns the request to *Client Acceptance* with an appropriate message.

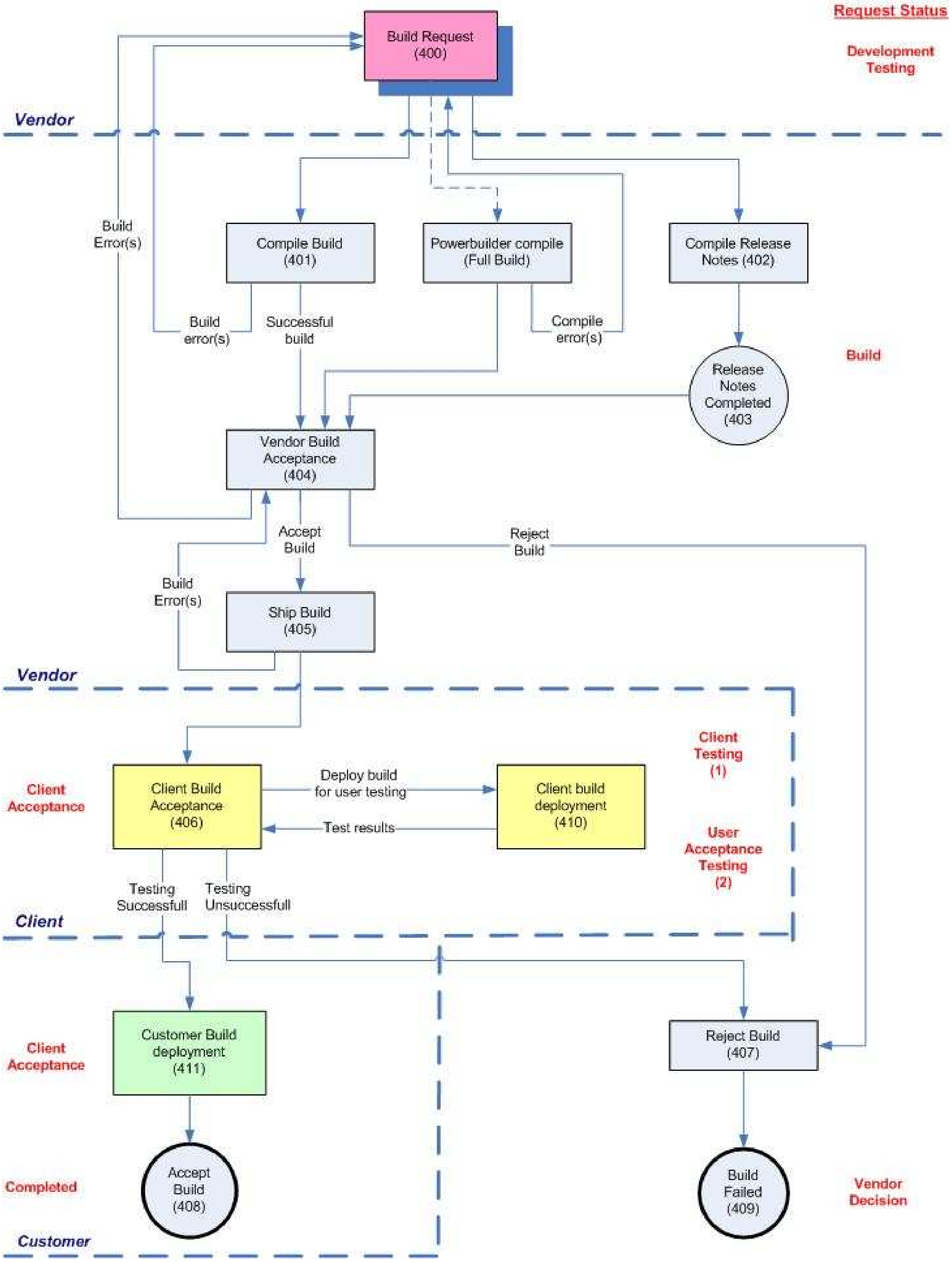
Addendum A: Request Workflow

Faveo : Request workflow – August 2008



Addendum B: Build Workflow

Faveo : iMed build workflow – August 2008


Request Status
Development Testing
Build Request (400)
Vendor
Build Error(s)
Compile Build (401)
Powerbuilder compile (Full Build)
Compile Release Notes (402)
Build error(s)
Successful build
Compile error(s)
Release Notes Completed (403
Build
Vendor Build Acceptance (404)
Accept Build
Reject Build
Build Error(s)
Ship Build (405)
Vendor
Client Testing (1)
Client Acceptance
Client Build Acceptance (406)
Deploy build for user testing
Test results
Client build deployment (410)
User Acceptance Testing (2)
Testing Successfull
Testing Unsuccessfull
Client
Client Acceptance
Customer Build deployment (411)
Reject Build (407)
Completed
Accept Build (408)
Build Failed (409)
Vendor Decision
Customer