

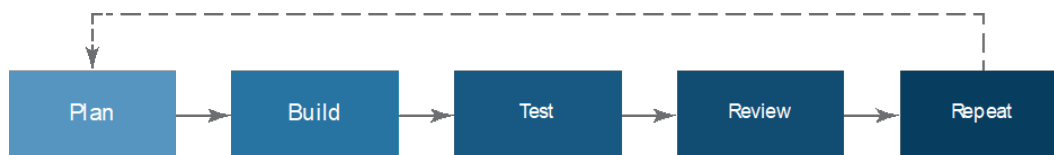


Contents

1	The Build.....	3
2	Steps	4
2.1	Project Planning.....	4
2.2	Iteration Roadmap.....	4
2.3	Iteration (1 to N)	5
2.3.1	Plan	5
2.3.2	Build	5
2.3.3	Test	5
2.3.4	Review	6
2.3.5	Feedback Assessment	7
2.3.6	Repeat	7
3	Feedback Assessment	8
3.1	Good Reasons to call something “In Scope.”	9
3.2	Bad Reasons to call something “In Scope.”	10
4	Change Requests.....	11
5	Read More.....	11

1 The Build

EBMS makes use of an iterative development methodology.



We build your solution in layers like a tradesman building a house.

A builder might build a house in this order:

1. Foundations
2. Wiring and plumbing
3. Walls
4. Windows
5. Roof

With each iteration, eBMS will add different layers to the solution; for example, we might add items in the following order:

1. Placeholder forms
2. Basic inputs and navigation
3. Workflow
4. Permissions
5. Reporting

If you live in an Agile world where an iteration might be focused more on a functional area, this approach will be slightly unfamiliar.

We will present our 'work in progress' to you regularly (In the review phase) so that you have an opportunity to guide the direction of development. Experience has taught us that no matter how comprehensive a Functional Specification is, it will not cover every one of the hundreds of design decisions that need to be made while building a system, and there will be questions that can only be answered by you.

Depending on the size and complexity of a project, eBMS can have from one to five iterations during this stage. We will reserve the last iteration for bugfixes and final polish and will strongly recommend against any changes to design at this point.

2 Steps

2.1 Project Planning

At this time, the client Project Manager and eBMS Project Manager will negotiate an updated project plan. With the detail in the Functional Specification agreed upon, we now have much greater clarity on exactly 'what' it is that we are planning to build.

Collaboratively you as the client and eBMS as the vendor will need to define:

1. What the priorities are for implementation – More complex areas might be useful to start addressing first, so there is more time for feedback and review;
2. How many iterations we would like to break the project into (Keeping any budget restraints in mind);
3. What testing will need to occur, when, and by who;
4. What timeline constraints exist, when are key milestones expected to occur, when are vital resources unavailable?
5. What new risks and issues have come to light.

It is crucial to book in your IT, Infrastructure and Integration support resources at this point as these tasks often require a long lead time

2.2 Iteration Roadmap

Now the eBMS Technical Lead will develop an Iteration Roadmap. This document will describe what parts of the implementation will occur in which iteration. This document will often change over time as dependencies and information gaps become clearer.

2.3 Iteration (1 to N)

2.3.1 Plan

eBMS will review the results of the previous iteration (if any) and make any adjustments to the Iteration Roadmap that are required. If adjustments are in fact required, the Technical Lead will raise any possible timeline concerns at this point.

2.3.2 Build

The Technical Lead and any further Technical resources will build the parts of the solution agreed for the current iteration. The person working on each individual task will test the deliverable for that task as well. You may receive questions during this step, as we discover ambiguities or missing information in the specification, or we may need your input on what the best user experience will be for a particular workflow.

The Technical Lead will draft a Handover Report for the current iteration. This is essentially a guide to 'what has been done' and 'how should you try it out'. It is not a full user manual but should be 'just enough' to let you review the new parts of the system.

2.3.3 Test

The eBMS Business Analyst will test the deliverables from the iteration against the Functional Specification, and then hand over to your team for more intensive, transactional and functional testing.

2.3.4 Review

2.3.4.1 Handover Meeting

The eBMS Project Manager will book a time with your Project Team to hand over the system for you to review the additions and changes that have been completed. Typically, this will be an in-person or screen sharing meeting, where the technical lead will talk you through the system in the same order as the Handover Report.

2.3.4.2 Your 'Work in Progress' Review

We strongly recommend that you select a core group of people for a review at the end of each iteration. This way, you are not going to the broader business, but restricting the collation of feedback to just the core project team, which allows for easier and more effective control of testing and feedback.

It is extremely important to include your subject matter experts so as to ensure our understanding of your requirements is correct and that nothing has been missed. Representatives of impacted areas are also vital to include. If they're not involved early on, you may find these stakeholders resistant to the changes implied with significant impacts on their business areas later.

At this stage, we are more interested in 'have we understood your requirements' than we are about 'this shows me an error message'. Keep a close eye on your Iteration Roadmap and Handover document – as we sometimes receive reports of 'bugs' that are just tasks that are scheduled for implementation two weeks later.

To save on unnecessary rework, the version of the system you are given to review will often be unpolished. It will probably have bugs, and some of them might stop your review while we fix it. We will do our best to avoid it, but there is little value in performing a full end to end testing process at each iteration while the system is still under construction.

2.3.4.3 Collating Feedback

We will provide you with a basic Excel template to fill out.

It is essential that you use this template in the format supplied and that you provide us with a single collated copy of the spreadsheet, as this is a crucial part of the Change Management process.

The nominated **Client Business Analyst** is responsible for collating the feedback from your testers into a single document. The input must be compiled in collaboration with the relevant stakeholder of each area, as there will be many differing opinions amongst your reviewers, and it is important that we are presented with only one clear definition for each change so as to avoid rework, delays and extra costs.

2.3.5 Feedback Assessment

(Mandatory for Iteration 1) Now would be a good time for an Account Manager Checkpoint (Refer back to Field Guide S2 – SIMPLE Requirements (Stage 1) –Account Manager Checkpoint for the procedure). Your eBMS Project Manager will organise this for you.



Once we have received your feedback, we will assess it.

Items that are defined as bugs will be scheduled for resolution in later iterations. Other items **may** be included subject to the budget constraints of the project, and the time taken to implement them. We will generally try to give you high-value low-cost changes to the extent possible while staying within the budget.

The word 'change' will come up a lot at this stage. Please bear with us in our usage, as when eBMS uses the word 'change', we mean that either a requirement must change, or a design must change. We do not necessarily mean that 'this is out of scope and will cost you additional money', it is merely a marker for tasks that were not planned for and may need to be further considered.

Please refer to Section **3. Feedback Assessment** for details

If you are keen for eBMS to implement feedback items that cannot be completed in the current scope and budget, we can draft you a Change Request document with costings for budget approval. Once approved, these will be incorporated into an updated project plan.

Unfortunately, the project scope is defined at the start to be able to provide an accurate estimate of costs and timelines, meaning that under many circumstances additional alterations or changes will be impossible without a corresponding increase in Budget and Timeline.

Please come and talk to us if something assessed as out of scope is of particular importance, and we will work with you to determine an appropriate solution.

2.3.6 Repeat

Refer to your Iteration Roadmap to understand the iteration structure of your project. The same process will be followed for each iteration, except for the last (referred to as the final iteration) which is dedicated to 'cleaning up'.

3 Feedback Assessment

When assessing an item against scope, there are several factors to weigh up. This section defines the process that we will use to evaluate any feedback provided to EBMS during the implementation of your project.

At a high level, there are four categories:

Scope Category	Nature of Request	Meaning
In Scope	Bug	These items are explicitly defined in the Requirements Specification or Contract.
	Not Yet Implemented	There is further detail on assessing scope later in this document.
Out of Scope	Out of Scope	These are items that would result in increasing the scope significantly. This could include a whole new process or significant changes to the structure of the system. This would also apply to items which might increase maintenance or licensing costs.
Discretionary – Negotiable (subject to fit into the existing budget, or if required, seeking further budget)	Change of Requirement	Projects are typically priced without a detailed Requirements Specification having been produced. After we have signed off a Requirements Specification, we will have a much clearer understanding of how much discretionary budget there is available in the project after allowing for the 'In Scope' (Must Have) items.
	New Requirement	
	Design Change / Refinement	
		Generally: we will deliver the 'In Scope' items first, and only limited minor discretionary items, so that we can remove as much budget uncertainty as possible before looking at the rest of the 'nice to have' items. After this there is some scope for including discretionary feedback should there be budget remaining
No Action Required	Information Only	Examples could be suggestions, questions or clarifications. Generally, there is no contention here as primarily included for communication and not asking for action.

Remember that unless EBMS has made an **unreasonable error**: the budget for the project must cover every action by staff performed in delivery of a project, including analysis, planning, design, implementation, testing, documentation, fixing bugs, and reasonable rework.

3.1 Good Reasons to call something “In Scope.”

When assessing for ‘in Scope’ there is often some debate about definitions. Requirements Specification documents and Contracts, no matter how detailed, will usually not include every little thing that the client wants. This framework can be used to reasonably assess (when there is doubt) whether an item is ‘In scope’ or ‘not in scope.’

Weight	Description	Rationale
Very High	Item is explicitly defined as a deliverable in the Requirements Specification, or Contract.	These items will be delivered unless there are extraordinary circumstances, eg an incompatibility with an associated system which was not able to be identified prior to testing.
High	Item was demonstrated to the client in the context of the <u>specific</u> solution that was being sold to them. (Some demonstrations do show multiple options, in which case this would not apply)	These items will be delivered where possible.
High	EBMS previously agreed in delivering the item in writing with a clear indication that it would be at zero cost.	These items will be delivered unless mutually agreed otherwise.

3.2 Bad Reasons to call something “In Scope.”

The below reasons have been given before as reasons that something is ‘In scope’ – as these are not related to a commitment we have made; these reasons will probably not be strongly considered.

Rather than debating whether an item should be considered ‘in scope’, please consider using the ‘Discretionary’ category instead.

Weight	Description	Rationale
Low	The current design doesn’t make sense.	<p>The design of a system is based on EBMS trying to build a system to meet your requirements.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin: 5px 0;"> <p>If the requirements are incorrect, then this will flow into the design of the system – for which EBMS cannot take financial responsibility. There is ample opportunity to review your requirements before we start building and using up the build budget.</p> </div> <p><i>On the other hand: <u>Sometimes we do something dumb, and we will fix it.</u></i></p>
Low	This item is implied by another requirement.	Generally, this is not a good reason to assume that a task or requirement is ‘in scope’. If it were obvious, it would be written in the requirement specifications.
Low	Without this item, the system will be hard to use.	EBMS will try to deliver a system that is usable to the client, but we do operate under budget constraints and cannot indefinitely implement improvements.
Very Low	This item is in our Business Requirements Document	It is the responsibility of the client to ensure that their Business Requirements are met by the Requirements Specification. EBMS generally has no obligation here; however we will do our best to point out gaps to you. If there are issues, make use of the discretionary category to fill them.
Very Low	This item has been previously mentioned; or sent in an email, or was in a document that we were provided.	The mention of a business need does not constitute an agreement to do a task, or that it is in scope.
Very Low	“I assumed that this would be included”; “This is common sense.”	EBMS spends a lot of time and effort, putting together documentation with the client so that we know what they expect and what they do not. If you are not sure you must ask, never assume.

4 Change Requests

Throughout the life of a project, it is not unusual to identify gaps in the original requirements or to see opportunities to improve workflow or functionality – resulting in a potential change to the system and hence a shift in the scope of the project.

Any change to scope will have implications, usually about the effort required to implement that change, the subsequent effects on the time required, and the cost of the engineers who will make those changes. *NB that there will also likely be internal costs and resourcing impacts that you should consider in your interior departments' planning & budgeting!

These potential changes must be analysed and actioned in a planned and considered manner, and the process will start by formally documenting the details of the change.

Your eBMS Project Manager will provide you with a template where you will document the change (or changes) required. Upon receipt of the completed template, eBMS will undergo a process of analysis, considering implications on other data, functionality, effort required and associated cost. It is likely you will be asked to clarify parts of the request to ensure a thorough understanding of the requirements. Note that time spent analysing the request will be charged against the project, so be concise and accurate.

eBMS will provide their analysis, which will include any identified risks or issues for undertaking the change, and will include details of any anticipated change to your project timeline and additional costs (if applicable). This will be documented in a Change Request form.

Upon acceptance of the Change Request your eBMS Project manager will schedule the work required and update affected documentation (eg Project Schedule, Budget)

5 Read More

The next guide in this sequence is S4 - SIMPLE Requirements (Stage 1).

Document	Title	Content
S4	SIMPLE Launch (Stage 3)	Explains UAT, Training and Release to Production.



© Copyright 2019 eBMS Pty Ltd

For further information, please contact:
info@ebms.com.au | 1300 721 159 | 03 9958 0000