



Getting Started with SmartBPM

An appraisal of the Pega Developer Network article and a proposed alternative

by Declan Chellar

Version 1.3



0. Contents

0. Contents.....	2
1. Introduction	4
1.1. Purpose	4
1.2. Scope.....	4
1.3. Intellectual property	4
1.4. Assumptions.....	4
1.5. Future versions of this document	4
1.6. About the author	4
2. PERCEIVED AREAS FOR IMPROVEMENT	5
2.1. Divergence from the Rational Unified Process	5
2.2. Confusion of activity with tool use	5
2.3. Key Deliverables not always correspond to Activities	5
2.4. Confusion of activity, outcome and deliverable	5
2.5. Imprecise use of grammar	5
2.6. Too few roles.....	6
2.7. Assignment of responsibilities	6
2.8. Lack of phase pre-requisites	6
3. <i>GETTING STARTED WITH SMARTBPM</i>	7
3.1. SmartBPM Methodology: Project Initiation	7
3.2. SmartBPM Methodology: Inception Phase.....	8
3.3. SmartBPM Methodology: Elaboration Phase	10
3.4. SmartBPM Methodology: Construction Phase	12
3.5. SmartBPM Methodology: Transition Phase.....	13
3.6. SmartBPM Methodology: Go Live.....	14
4. REVIEW OF SMARTBPM CRITICAL FLAWS.....	15
3.7. Using the Inception phase to investigate current processes.....	15
3.8. Not highlighting the limitations of the Application Profiler	15
3.9. Not addressing the mitigation of key technical risks	15
3.10. Not addressing the need to build testing environments during Construction	15
3.11. Iterating between Elaboration and Construction	15
5. AN ALTERNATIVE TO SMARTBPM	17
5.1. Key principles	17
5.2. Roles.....	17
5.3. Project Initiation	18



5.4.	Phase Objectives (High Level Tasks)	18
5.5.	Pre-requisites	18
5.6.	Medium Level Tasks	18
5.7.	Inception Phase.....	19
5.8.	Phase Objectives (High Level Tasks)	20
5.9.	Pre-requisites	20
5.10.	Medium Level Task	21
5.11.	Elaboration Phase	23
5.12.	Phase Objectives (High Level Tasks)	23
5.13.	Pre-requisites	23
5.14.	Medium Level Tasks	24
5.15.	Construction Phase	26
5.16.	Phase Objectives (High Level Tasks)	26
5.17.	Example Pre-requisites	26
5.18.	Medium Level Tasks	26
5.19.	Transition Phase.....	28
5.20.	Phase Objectives (High Level Tasks)	28
5.21.	Example Pre-requisites	28
5.22.	Medium Level Tasks	28
5.23.	Go Live.....	29
5.24.	Phase Objectives (High Level Tasks)	29
5.25.	Example Pre-requisites	29
5.26.	Medium Level Tasks	30



1. Introduction

1.1. Purpose

The author aims to provide an appraisal of the Pega Developer Network article “Getting Started with SmartBPM”¹, based on his experiences working on Pega projects in the field. Furthermore, the author will provide an alternative “Getting Started” guide, which will address any perceived shortcomings of the PDN article and the SmartBPM methodology.

1.2. Scope

This document addresses the SmartBPM methodology only and not the Pega Scrum methodology. Furthermore, this document does not address techniques specific to any discipline (e.g., use case modelling).

1.3. Intellectual property

The original PDN article is the intellectual property of Pegasystems. It is quoted and paraphrased in this document for reference purposes only, since to appraise the specifics of the article would not be possible without reference to the article itself. References to the original PDN material appear in blue, italicised font. Non-italicised, black (or sometimes red) font indicates the intellectual property of the author of this document. **This document may be copied but may not be sold.**

1.4. Assumptions

The author assumes that the reader is familiar with PRPC terminology and acronyms.

1.5. Future versions of this document

This document will be updated to reflect the following:

- Feedback from readers’ experiences
- Changes to Pega’s SmartBPM methodology
- The author’s future experiences

1.6. About the author

Declan Chellar is a Senior Business Analyst based in the European Union. He provides consultancy on business change initiatives, particularly those using Pegasystems’ PRPC. He has worked in IT since 1996 and primarily as an analyst since 1998 for companies such as Accenture, Knowledge Rules, Tata Consulting Services, Virtusa, Capgemini and Electronic Data Systems. Declan also has full systems lifecycle experience and has successfully delivered software as a PRPC analyst, developer and team manager. He is a member of the British Computer Society and the International Institute of Business Analysts (IIBA™) and holds the following relevant qualifications:

- ISEB Diploma in Business Systems Development (specialising in Business Analysis)
- Pegasystems PRPC v6.1 Certified Business Architect
- Pegasystems Certified Methodology Black Belt
- Certified BPMN 2.0 Professional

Declan writes about his business analysis experiences on his blog “Analysis Fu”² and tweets about business analysis as @AnalysisFu.

¹ http://pdn.pega.com/expandedAudience/methodology_smartBPM.asp

² <http://www.chellar.com/blog>



2. PERCEIVED AREAS FOR IMPROVEMENT

Note that Pega clearly states that the SmartBPM methodology “is **not** a concrete prescriptive process”³, that it is adaptable and that the phase information is an outline only⁴. Therefore some of the author’s suggestions are nothing more than an adaption of the methodology and others merely add detail to what is currently in outline form. However, other suggestions will be the result of perceived problems with the original PDN article.

2.1. Divergence from the Rational Unified Process

SmartBPM is based on the Rational Unified Process, yet diverges (mistakenly, in the author’s opinion) from the RUP in two key ways:

- SmartBPM does not address the mitigation of key technical risks
- Pega implies that we should iterate across the Elaboration and Construction phases

These points are dealt with in more detail in section 4.

2.2. Confusion of activity with tool use

When activities are confused with the tools that can help you perform those activities, focus is lost and people can (and do) think they have done their job simply because they have produced something using a particular tool. While it is entirely appropriate for Pega to promote the use of tools such as the Application Profiler and the Project Management Framework (PMF), the use of the tools is often presented as if it were an end in itself. The alternative guide (see section 5) will separate activities from suggested tools.

2.3. Key Deliverables not always correspond to Activities

The activities and key deliverables of each phase in the article do not always correspond, with the result that the activity which produces a deliverable is not explained or the deliverable produced by an activity is not made clear. The alternative guide (see section 5) will ensure that activities and deliverables correspond.

2.4. Confusion of activity, outcome and deliverable

Pega’s tables in the PDN article are split into Activities and Key Deliverables. However, sometimes activities appear in the Key Deliverables tables and vice versa. Moreover, outcomes also appear in both tables. The alternative guide (see section 5) will separate activities from outcomes.

2.5. Imprecise use of grammar

Some may regard this as mere pedantry; however, incorrect use of grammar can cause confusion in the reader. In order to maximise clarity, imperative verbs should be used for activities, the past participle should be used for outcomes and nouns should be used for deliverables. This is not applied consistently in the Pega article. The alternative guide (see section 5) will be grammatically precise and consistent.

³ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp>

⁴ http://pdn.pega.com/expandedAudience/methodology_smartBPM.asp



2.6. Too few roles

Not defining enough roles can result in activities being assigned to the wrong role or the merging of roles and, therefore, the blurring of responsibilities. The alternative guide (see section 5) will define more roles.

2.7. Assignment of responsibilities

The author does not always agree with Pega's assignment of activities to roles. Moreover, the original Pega article simply indicates which roles are involved, but not how they are involved. A RACI format would be more useful. The alternative guide (see section 5) will suggest a re-assignment of activities in RACI format.

2.8. Lack of phase pre-requisites

The author feels that the pre-requisites of each phase should be made clear. By defining pre-requisites, it is then easier to draft phase-readiness checklists. The alternative guide (see section 5) attempts to do this.



3. GETTING STARTED WITH SMARTBPM

Note that throughout this section, the author's comments will appear in un-italicised, black font. However, unitalicised, red text is used to highlight what the author considers to be critical flaws in the SmartBPM methodology (see section 4 for a review of these perceived flaws). Text in italicized, blue font summarises, paraphrases or quotes the original PDN article for reference purposes only.

Pega tells us that the SmartBPM methodology as an agile and adaptable process framework. The introduction stresses the importance of business analysis throughout the phases, particularly in relation to:

- *Change management*
- *Stakeholder management*
- *Reporting on application metrics and productivity gains*
- *Process improvement*

3.1. SmartBPM Methodology: Project Initiation

*Project Initiation Activity*⁵

- *Kick-off project*
- *Set project expectations*
- *Transfer knowledge*
- *Agree common requirements terminology*

General comments:

- In order to initiate a project, the client business should have developed their To Be business model. However, this is often not the case and projects are initiated without the client's knowing what To Be state they are aiming for.
- On enterprise-wide projects, the client business may well have a dedicated technical solutions team which will decide which elements of the To Be business model will be implemented in PRPC. This may happen before or during project initiation.

3.1.1. Comments on Project Initiation Suggested Activities

Performing walk-throughs of the As Is process

Comment:

- Pega's assumption seems to be that process improvement will happen during software development. However, PRPC is first and foremost a process implementation tool, not a process re-engineering tool. The project team should be more concerned with the To Be process. If a company has not already gone through an exercise of process improvement and produced a To Be model before choosing the technology, then there will be little or no time for process improvement. Moreover, PRPC might not even be the right technology to implement the needed improvements.
- However, an understanding of the current process is useful in helping the team understand the improvements the client is looking to implement.

⁵ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



Identifying “quick wins” for the client and prioritising requirements into “slivers”

Comment:

- The phrase “quick wins” can lead people to prioritise what can be developed quickly, which is often the easy, low value, business functionality. The phrase “business priorities” should be used.
- However, it should be noted that elsewhere in the PDN, Pega emphasises “greatest returns”.
- Pega often uses its own terminology where standard words already exist in the industry. “Sliver” is an example. A “sliver” is essentially the software to be developed for a single release.
- It is only during Inception that we gather enough information to understand the scope and size of the project. Therefore it is only after Inception that we can plan slivers. This task does not belong in this phase.

3.1.2. Comments on Project Initiation Key Deliverables

Comment:

Most of the listed deliverables are not deliverables at all but are either activities or the mechanism through which something is delivered, for example:

- *Proofs of concept*
- *Workshops*
- *Training courses*
- *Identification of milestones*

3.2. SmartBPM Methodology: Inception Phase

Inception Phase⁶

- *Define scope*
- *Capture requirements*
- *Identify process improvements*
- *Define slivers*
- *Capture specifications using the Application Profiler*
- *Create an Application Profile*
- *Estimate development effort*

General comments:

- Despite Pega’s definition, the Inception phase is not the appropriate time for process improvement. Formal process improvement results in a To Be business model, which should be in place before PRPC is even chosen as a technical solution. When a business change initiative is so immature that the client does not know what their To Be business model is by the time Inception starts, then the project is already on the path to failure. While project staff should be looking to improve business processes throughout the project, for example, by merging or changing the proposed order of process steps, PRPC is not a process re-engineering tool; it is a process implementation tool. **The author regards Pega’s guidance on this matter to be a critical weakness in SmartBPM.**
- The Application Profiler, also known as the Application Profile Wizard (APW), should not always be used. For example, where the project is to implement the CPM framework with minimal enhancement, a GAP analysis should be done and existing Pega Rules amended, whereas the

⁶ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



Application Profiler is for building a bespoke application. Moreover, the Application Profiler was designed to capture requirements about business processes, workflows in particular. It is not suited to business rules engine (BRE) projects where no processes are to be implemented. It can even be more complicated than that, when a project is part workflow, part BRE, part data retrieval/display and part non-workflow process. You will find the APW does not work as smoothly as you expected if you apply it to your project with a broad brush. **The author regards Pega's failure to make clear the limitations of the APW to be a critical weakness in SmartBPM.**

- At the end of Inception, the PM should establish the initial project plan overall and the iteration plan for the Elaboration phase. Pega omits this.
- Pega often uses its own terminology where standard words already exist in the industry. "Sliver" is an example. A "sliver" is essentially the software to be developed for a single release.
- Pega terminology often masks industry-standard concepts and confuses tasks with tools. For example, "DCO session" actually means "requirements workshop". The activity should not be confused with the tools Pega wants us to use.

3.2.1. Comments on Inception Phase Suggested Activities

<p><i>Define scope for the programme, project or "sliver"</i></p> <p>Comment:</p> <ul style="list-style-type: none">• Slivers can only be defined once the program/project scope has been understood, i.e., post Inception.
<p><i>Define business requirements and organising them into "slivers"</i></p> <p>Comment:</p> <ul style="list-style-type: none">• The business requirements should be defined by the Business before PRPC was even chosen as a technology. The task here should be to record those requirements.• Slivers can only be defined once the program/project scope has been understood, i.e., post Inception.
<p><i>Document high-level requirements, build use cases and work types</i></p> <p>Comment:</p> <ul style="list-style-type: none">• There are three activities here.• Use Case level should be defined here (there are four levels).• Use Cases are not appropriate for BRE projects.• Work Types are actually part of the high level design, since identifying work types affects the class structure. This task should be assigned to the LSA but Pega assigns this task to the BA.
<p><i>Identify process improvements and targets</i></p> <p>Comment:</p> <ul style="list-style-type: none">• The client should already have gone through an exercise of process improvement and produced a To Be model before choosing the technology. The task should simply be to understand the desired process improvements and targets.• Moreover, this task should be assigned to everyone on the project team.
<p><i>Estimate the effort</i></p> <p>Comment:</p> <ul style="list-style-type: none">• The PDN article makes this a task for the Project Manager alone. A RACI would be more useful here, since all team leaders are responsible for estimating their team's effort. This is not simply a task for the PM.• Be aware that Pega's sizing metrics might not coincide with those of your company.



3.2.2. Comments on Inception Phase Key Deliverables

Most of the listed deliverables are not deliverables at all but are either activities or the mechanism through which something is delivered, for example:

- *Identify re-use*
- *Initiate DCO sessions*
- *Enter artifacts into the Application Profiler*
- *Use the sizing tool*

3.3. SmartBPM Methodology: Elaboration Phase

*Elaboration Phase*⁷

- *Build the foundation of the implementation using the Application Accelerator*
- *Expand requirements*
- *Perform process discovery*
- *Design user interface*
- *Establish application standards*
- *Draft testing and migration plans*

General comments:

- SmartBPM is based on the Rational Unified Process (RUP). In the RUP, one of the major purposes of Elaboration is to **mitigate key technical risks**. Pega does not address this at all and it is an important omission. The author has met Pega Certified Lead Systems Architects who did not understand the importance of mitigating the key technical risks during Elaboration, with the result that they tackled the “easy” stuff first for the “quick win”, but were left at the end of the project with little time left to tackle the “showstoppers”. **The author regards Pega’s failure to address the mitigation of technical risks during Elaboration as the critical weakness in SmartBPM.**
- Since the the Application Accelerator requires an Application Profile as an input, the AA will not be used in circumstances where there is no need for an AP (see section 3.2).

3.3.1. Comments on Elaboration Phase Suggested Activities

Complete “DCO sessions”

Comment:

- Pega terminology often masks industry-standard concepts and confuses tasks with tools. “DCO session” in this context actually means “analyse and document detailed requirements”. The activity should not be confused with the tools Pega wants us to use.

Elaborate on requirement by fleshing out the Application Profile using atomic use cases and work types

Comment:

- This is essentially the point of the previous activity repeated but with reference to specific DCO tools.

“Work iteratively, cycling through Elaboration and Construction phases”

Comment:

- This is normal practice in iterative development, rather than a specific task.
- We should cycle through the software lifecycle disciplines, not project phases.

⁷ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



Completing the Application Profile and run the Application Accelerator to establish the foundation of the application

Comment:

- As throughout, a RACI would be useful here to show that while the PM is accountable for this task, the LSA is responsible.
- An Application Profile is not always desirable, therefore running the Application Accelerator is not always possible.

Set up environment

Comment:

- As throughout, a RACI would be useful here to show that while the PM is accountable for this task, the LSA is responsible.

Establish governance model

Comment:

- Pega assigns this task to the BA and PM. However, each team leader will have input into the governance model.

Establish Deployment Plan, Test Plan, Issue Tracking Process

Comment:

- The iteration plans for the Construction phase should also be established at the end of Elaboration.

“Use the Pre-Flight and PAL tools”

Comment:

- Using a particular tool does not fully expose what the task and objectives are.

3.3.2. Comments on Elaboration Phase Key Deliverables

Comment:

Most of the listed deliverables are not deliverables at all but are either activities or the mechanism through which something is delivered. Moreover, deliverables are often confused with the tools Pega uses. For example:

- *Complete Application Profile*
- *Run Application Accelerator*
- *Create draft flows and UIs*
- *Establish links between requirements, use cases and Rules*
- *Develop a testing plan*



3.4. SmartBPM Methodology: Construction Phase

*Construction Phase*⁸

- *Build the implementation*
- *Assign configuration tasks*
- *Unit test components are unit tested*
- *Develop test scripts*

General comments:

- An omission here is that by the end of the Construction phase, all testing environments must be built; otherwise the Transition phase will be delayed. **The author considers this omission to be a critical flaw in SmartBPM.**

3.4.1. Comments on Construction Phase Suggested Activities

<i>Assign application configuration tasks</i> Comment: <ul style="list-style-type: none">• Pega assigns this task to the PM. However, the LSA should be responsible. The PM is unlikely to have the technical knowledge and should not be micro-managing.
<i>Unit test components</i> Comment: <ul style="list-style-type: none">• Pega assigns this task to the developer role and the QA role. However, the QA team should not be responsible for unit testing.
<i>Build automated testing scripts</i> Comment: <ul style="list-style-type: none">• This activity should start during Elaboration as the requirements become more detailed. However, it might not finish until Construction.
<i>Run automated testing test scripts and begin system test process</i> Comment: <ul style="list-style-type: none">• System testing belongs to the Transition phase.
<i>“Complete integration system testing”</i> Comment: <ul style="list-style-type: none">• This activity belongs to the Transition phase.
<i>Use PMF and TMF to track progress and manage defects</i> Comment: <ul style="list-style-type: none">• Inadequate separation of activities and tools.
<i>Conduct Design and User Experience reviews</i> Comment: <ul style="list-style-type: none">• This is not a single activity.• The purpose of these reviews should be stated.• The design review should be held during Elaboration.

⁸ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



“Review PMF and start preparations for the next sliver”

Comment:

- The purpose of activity should be stated; otherwise it is just an instruction to use a particular tool.

Conduct a change management review

Comment:

- This should be a regular activity throughout Elaboration and Construction.

Use the Pre-Flight and PAL tools

Comment:

- Using a particular tool does not fully expose what the task and objectives are.

3.4.2. Comments on Construction Phase Key Deliverables

Comment:

Most of the listed deliverables are not deliverables at all but are either activities or the mechanism through which something is delivered. Deliverables are often confused with the tools Pega uses. Moreover, some deliverables do not correspond to a previously listed activity. For example:

- *Using PMF, assign work*
- *Complete Unit/ Integration testing*
- *Build a Release Candidate*
- *Complete all iterative automated testing*
- *End-user training*

3.5. SmartBPM Methodology: Transition Phase

Transition Phase⁹

- *Deploy the implementation to the testing teams*
- *Ensure the quality of the application and its readiness for deployment to production*

General comments:

- In addition, it is during the Transition phase that business readiness activities take place, such as training end users in new processes and software and training the maintenance team.

3.5.1. Comments on Transition Phase Suggested Activities

Ensure solution is maintainable

Comment:

- Does “maintainable” mean that the solution has been designed for ease of maintenance or that the maintenance team is prepared to take over?
- If the former, then this task should be completed when the solution is defined (during Elaboration).
- Surely the LSA is responsible for this task.

“Review PMF and finalize plans for the next sliver”

Comment:

- The purpose of activity should be stated; otherwise it is just an instruction to use a particular tool.

⁹ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



3.5.2. Comments on Transition Phase Key Deliverables

Most of the listed deliverables are not deliverables at all but are either activities or activity outcomes. For example:

- *Participate in Go/ No-Go status update review*
- *Production Support plan and Support training completed*
- *End-user training completed*
- *Review reports and metrics*
- *Complete a Phase Readiness checklist*

3.6. SmartBPM Methodology: Go Live

Go-Live Activity¹⁰

- *Occurs when a sliver is launched into production*
- *Business users begin to use the live application.*

3.6.1. Comments on Go Live Suggested Activities

No comments.

3.6.2. Comments on Go Live Key Deliverables

Comment:

Most of the listed deliverables are not deliverables at all but are activities. For example:

- *Deploy final release candidate*
- *Maintain application*
- *Monitor system*
- *Deliver status update to stakeholders*

¹⁰ <http://pdn.pega.com/DevNet/Overviews/MethodologyOverview.asp#WhatIsSB>



4. REVIEW OF SMARTBPM CRITICAL FLAWS

While the author acknowledges that Pega designed the SmartBPM methodology to be adaptable and that the PDN articles are outlines only, there are several critical flaws which need to be addressed.

3.7. Using the Inception phase to investigate current processes

Despite Pega's definition, the Inception phase is not the appropriate time for process improvement. Formal process improvement results in a To Be business model, which should be in place before PRPC is even chosen as a technical solution. When a business change initiative is so immature that the client does not know what their To Be business model is by the time Inception starts, then the project is already on the path to failure. While project staff should be looking to improve business processes throughout the project, for example, by merging or changing the proposed order of process steps, it should be remembered that PRPC is not a process re-engineering tool; it is a process implementation tool.

3.8. Not highlighting the limitations of the Application Profiler

Pega stress the use of the Application Profiler throughout its guidance on SmartBPM. However, the Application Profiler, also known as the Application Profile Wizard (APW), should not always be used. For example, where the project is to implement the CPM framework with minimal enhancement, a GAP analysis should be done and existing Pega Rules amended, whereas the Application Profiler is for building a bespoke application. Moreover, the Application Profiler was designed to capture requirements about business processes, workflows in particular. It is not suited to business rules engine (BRE) projects where no processes are to be implemented. It can even be more complicated than that, when a project is part workflow, part BRE, part data retrieval/display and part non-workflow process. You will find the APW does not work as smoothly as you expected if you apply it to your project with a broad brush.

3.9. Not addressing the mitigation of key technical risks

SmartBPM is based on the Rational Unified Process (RUP). In the RUP, one of the major purposes of Elaboration is to mitigate key technical risks. Pega does not address this at all and it is an important omission. The author has met Pega Certified Lead Systems Architects who did not understand the importance of mitigating the key technical risks during Elaboration, with the result that they tackled the "easy" stuff first for the "quick win", but were left at the end of the project with little time left to tackle the "showstoppers".

3.10. Not addressing the need to build testing environments during Construction

An omission in SmartBPM is that by the end of the Construction phase, all testing environments must be built; otherwise the Transition phase will be delayed.

3.11. Iterating between Elaboration and Construction

Elaboration and Construction are distinct phases in a project and each has its own objectives (see sections 5.12 and 5.16). Once a phase is complete, we move into the next phase. However, we do not move backwards and forwards between phases.

The Elaboration phase is so called because one of the key objectives is to elaborate on the requirements. However, the name sometimes confuses people who have only ever used a Waterfall



methodology and causes them to think that it is only about elaborating requirements. The Construction phase is named for its main activity: construction of components of code. This causes some people to think that it is only about construction and they assume therefore that in order to work iteratively, they have to continuously cycle between those two phases. This misunderstanding seems to be represented in one of Pega's illustrations¹¹, which is represented here:

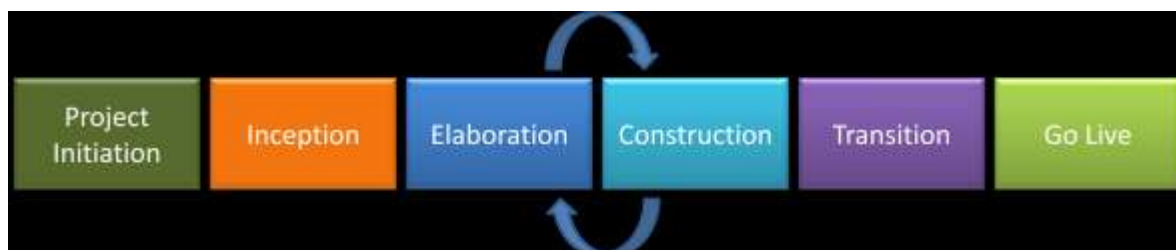


Figure 1: Pega's illustration of iteration from the PDN

This illustration misrepresents the nature of iteration. Moreover, in the PDN, Pega explicitly tell us that we should be *“cyclling through Elaboration and Construction phases”*.

The terms **“iteration”** and **“iterative”** refer to the fact that we constantly cycle through the project **disciplines** (not phases) as we deal with each increment of work to be completed. For example, we might go through the analysis, design, construction and function testing of a single, critical path through a use case, then do the same for a critical path of a **different** use case, before cycling back to complete the first use case. Therefore if we were to visually represent the concept of iteration, it would look more like the following:

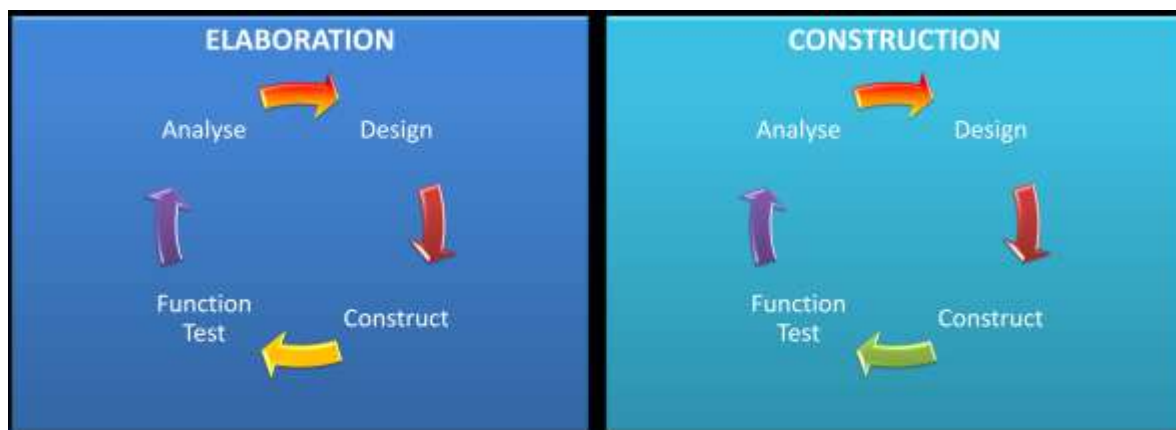


Figure 2: The author's illustration of iteration

Some business-critical use case paths will go through a full cycle during elaboration. Use case paths of low business value might not even start analysis until sometime during Construction. Most use case paths will start their lifecycle during Elaboration and complete it during Construction.

¹¹ http://pdn.pega.com/ba_landing.asp



5. AN ALTERNATIVE TO SMARTBPM

The following is one possible alternative to the SmartBPM guide. You should take it and use it in a way that best fits your project needs. You may find it easier to manage in a spreadsheet, rather than a document table.

5.1. Key principles

Work collaboratively	Co-locate teams and have staff from different disciplines working together rather than handing documentation off to each other
Be lean	Do not let the production of documentation hold up progress in software development – much documentation is actually only needed for maintenance purposes, and so can be produced late in the project, once the software itself has been built
Model visually	Process flows are more easily and coherently investigated and explained using diagrams. Textual explanations of processes should be high-level and for introductory purposes only.
Work iteratively	Tackle high-risk (high-priority) business scenarios first and take them through the full software development lifecycle. Later cycle back to deal with lower risk scenarios. This mitigates risks early and reduces the chance of encountering “show-stoppers” late in the project.
Deliver incrementally	Produce software in smaller increments (starting with high-value increments) so that the customer gains some practical benefit early on. Avoid the so-called “quick win” mentality, which focuses the mind on that which can be delivered most easily, rather than what is of the highest value to the customer.
Test early and often	Encountering critical defects late does not mitigate risk early and increases the chance of encountering “show-stoppers” late in the project.
Delivering software is more important than generating documents	Documentation exists as a substitute for personal communication. By working collaboratively, especially between BAs and developers, we reduce the need for documentation in order to build software. However, documentation is still important in handing over an application to maintenance and live support teams. This means documentation still needs to be produced, but perhaps later in the project when it is not an obstacle to software development.

5.2. Roles

The following roles are used in the RACI matrices below. Note that you should adapt the list of roles to suit the nature of your project and organisation. Moreover, your project-specific RACI will probably list some tasks at a lower level of granularity.

ROLE	DESCRIPTION	NOTES
Cust	Customer	This represents the customer in the broad sense, including: <ul style="list-style-type: none">• Business sponsors• Business stakeholders



		<ul style="list-style-type: none"> • Business enterprise architects • Business technical architects
BA	Business Analyst	Represents any BAs working on the project. There may be two types of BA on your project: <ul style="list-style-type: none"> • Business specialist • PRPC specialist
PM	Project Manager	This role may be in charge of an entire programme cutting across technology teams or in charge of the PRPC effort only.
LSA	Lead Systems Architect	A Pega-certified Lead Systems Architect
SA	System Architect	A developer conversant with PRPC
QA	Quality Assurance	This represents both the Testing Manager and the testing team.
Train	Training	This represents the team responsible for rolling out the change initiative to the end users.
Maint	Maintenance	This represents the team responsible for system maintenance after deployment to production. This team may include several of the roles listed above

5.3. Project Initiation

5.4. Phase Objectives (High Level Tasks)

- Ensure that the development team gains an initial understanding of the business needs and design constraints
- Ensure that the business stakeholders gain an initial understanding of PRPC
- Ensure that all participants understand the methodology to be used
- Introduce all participants to a common project vocabulary

5.5. Pre-requisites

- Business case
- High level statements of requirement
- High-level To Be business process models
- Logical Data Model (to ensure that the business understands its own data needs)
- High level solution design

5.6. Medium Level Tasks

PROJECT INITIATION												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	



PROJECT INITIATION												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Present summary of business case	Ensure all stakeholders understand strategic objectives and key risks	None specified	• The presentation itself	A/R	I	I	I	-	I	I	-	
Present high level statements of requirement	Ensure all stakeholders understand the project scope	None specified	• The presentation itself	A/R	I	I	I	-	I	I	-	
Present the To Be business process	Ensure all stakeholders have a vision of the process to be implemented	None specified	• The presentation itself	A/R	I	I	I	-	I	I	-	
Present the high level solution design	Ensure all stakeholders understand what is to be implemented using PRPC and what interfaces will be required	None specified	• The presentation itself	A/R	I	I	I	-	I	I	-	
Develop a Proof of Concept where necessary	Demonstrate that PRPC is the appropriate tool	None specified	• Proof of Concept	C	C	A/I	R	-	-	-	-	
Present methodology	Ensure all stakeholders understand the methodology to be used	None specified	• The presentation itself	I	I	A/R	I	-	I	I	-	
Enable training (e.g., PRPC, process modelling, use case modelling, data modelling)	Ensure all project participants have the necessary skills	None specified	• The training itself	A/R	I	I	I	I	I	I	I	
Produce a project roadmap	Ensure all stakeholders understand the likely lifetime of the project and the key milestones	None specified	• Project roadmap	C	I	A/R	I	I	I	I	-	
Produce a high-level staffing model	Ensure all stakeholders understand what staffing is likely to be needed	None specified	• Staffing model	C	C	A/R	C	-	C	-	-	
Plan Inception workshops	Ensure necessary time for Inception is planned and participants are available	None specified	• Inception workshop plan	C	R	A/C	I	I	I	I	I	
Define requirements traceability model	Ensure all stakeholders understand how requirements are to be traced	None specified	• Requirements Traceability Matix	A	R	I	I	I	I	-I	-	
Document project risks	Ensure all stakeholders understand project risks and their mitigations	None specified	• Risk log	C	C	A/R	C	-	C	C	-	
Document assumptions	Ensure assumptions are clear to all stakeholders	None specified	• Assumptions log	C	C	A/R	C	-	C	C	-	

5.7. Inception Phase



5.8. Phase Objectives (High Level Tasks)

- Define the scope of the project
- Estimate the size of the work
- Identify business priorities and technical risks
- Produce an initial project plan based on business priorities and technical risks (dividing the work into several releases if necessary)

5.9. Pre-requisites

- All Project Initiation tasks listed in section 5.6 are complete.



5.10. Medium Level Task

INCEPTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Document high level requirements	Formally establish scope and ensure HLRs are held in a central location for ease of reference	<ul style="list-style-type: none"> Requirement Rule Application Profile Wizard Word Excel Industry-standard requirements tools 	<ul style="list-style-type: none"> Requirements artefact (see Tool Options) 	C	R	A/I	I	I	I	I	I
Identify candidate software use cases (SUCs), including Actors, for high level functional requirements	Establish placeholders for further requirements analysis	<ul style="list-style-type: none"> Use Case Rule Application Profile Wizard Word Industry-standard UC tools 	<ul style="list-style-type: none"> Documented SUCs (high level) Documented Actors 	C	R	A/I	I	I	I	I	I
Identify candidate business work types	Enable the LSA to define candidate PRPC work types	None specified	<ul style="list-style-type: none"> Candidate business work types 	C	R	A/I	I	I	I	I	I
Define candidate PRPC work types	Gain an initial understanding of the PRPC class structure	<ul style="list-style-type: none"> PRPC Work Type Application Profile Wizard 	<ul style="list-style-type: none"> Documented PRPC Work Types 	I	I	A/I	R	I	I	I	I
Identify interface constraints	Ensure that technical risks can be identified	<ul style="list-style-type: none"> Application Profile Wizard Word Excel 	<ul style="list-style-type: none"> Documented interfaces (high level) 	C	I	A/I	R	I	I	I	I
Identify reporting requirements	Ensure reporting needs are understood at the high level	<ul style="list-style-type: none"> Application Profile Wizard Word Industry-standard requirements tools 	<ul style="list-style-type: none"> Documented reporting requirements (high level) 	C	R	A/I	I	I	I	I	I
Identify correspondence requirements	Ensure correspondence needs are understood at the high level	<ul style="list-style-type: none"> Application Profile Wizard Word Industry-standard requirements tools 	<ul style="list-style-type: none"> Documented correspondence requirements (high level) 	C	R	A/I	I	I	I	I	I



INCEPTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Identify business priorities	Ensure business need is understood	<ul style="list-style-type: none"> Requirement Rule Use Case Rules Application Profile Wizard Word Excel Industry-standard requirements tools 	<ul style="list-style-type: none"> Documented business priorities 	C	R	A/I	I	I	I	I	I
Identify technical risks	Ensure technical risks are understood	<ul style="list-style-type: none"> Requirement Rule Use Case Rules Application Profile Wizard Word Excel Industry-standard risk-management tools 	<ul style="list-style-type: none"> Documented technical risks 	C	I	A/I	R	I	I	I	I
Present business needs overview to stakeholders	Confirm that business needs have been understood	<ul style="list-style-type: none"> Word Application Profile Wizard 	<ul style="list-style-type: none"> Documented business needs PRPC Application Profile document 	C	R	A/I	I	I	I	I	I
Estimate analysis effort	Contribute to project plan	<ul style="list-style-type: none"> Pega's sizing tool Any industry-standard estimation tools 	<ul style="list-style-type: none"> Analysis estimates 	I	R	A/I	I	-	-	-	-
Estimate development effort	Contribute to project plan	<ul style="list-style-type: none"> Pega's sizing tool Any industry-standard estimation tools 	<ul style="list-style-type: none"> Development estimates 	I	-	A/I	R	-	-	-	-
Estimate testing effort	Contribute to project plan	<ul style="list-style-type: none"> Pega's sizing tool Any industry-standard estimation tools 	<ul style="list-style-type: none"> Testing estimates 	I	-	A/I	-	-	R	-	-



INCEPTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Estimate deployment effort	Contribute to project plan	<ul style="list-style-type: none"> • Pega's sizing tool • Any industry-standard estimation tools 	<ul style="list-style-type: none"> • Deployment estimates 	I	-	A/I	R	-	-	-	-
Estimate training effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> • Training estimates 	I	-	A/I	I	-	-	R	-
Complete phase readiness checklist	Ensure that phase tasks have been satisfactorily completed	None specified	<ul style="list-style-type: none"> • Phase readiness checklist 	I	R	A/R	R	-	-	-	-
Produce initial project/release plan	Prepare for "Go/No Go" decision for the project	<ul style="list-style-type: none"> • Any project planning tool • PMF 	<ul style="list-style-type: none"> • Project plan 	I	C	A/R	C	I	C	C	-
Present project/release plan to stakeholders	Obtain "Go/No Go" decision for the project	<ul style="list-style-type: none"> • Any project planning tool • PMF 	<ul style="list-style-type: none"> • Go/No Go decision 	C	I	A/R	I	I	I	I	I

5.11. Elaboration Phase

5.12. Phase Objectives (High Level Tasks)

- Define low level requirements for high and (most) medium business priorities
- Mitigate key technical risks
- Establish technical architecture
- Develop and test key, high priority, business functions
- Define testing strategy
- Define deployment strategy
- Start preparing training materials

5.13. Pre-requisites

- All Inception tasks listed in section 5.10 are complete.



5.14. Medium Level Tasks

ELABORATION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Elaborate requirements	Define 80% of requirements by end of Elaboration	<ul style="list-style-type: none"> Workshops “Show and Tell” Use Case Rule Requirement Rule Visio Word Excel 	<ul style="list-style-type: none"> Approved low-level requirements 	C	R	A/I	C	C	C	I	-	
Set up development environment	Ensure work can proceed following “Go” decision	None specified	<ul style="list-style-type: none"> Development environment 	-	-	A/I	R	I	-	-	-	
Define Class Structure	Establish technical architecture	None specified	<ul style="list-style-type: none"> Class structure 	-	-	A/I	R	I	-	-	-	
Produce Physical Data Model	Establish technical architecture	None specified	<ul style="list-style-type: none"> Physical Data Model 	-	-	A/I	R	I	-	-	-	
Build Database	Establish technical architecture	None specified	<ul style="list-style-type: none"> Database 	-	-	A/I	R	I	-	-	-	
Define design principles	Ensure robust and consistent development	None specified	<ul style="list-style-type: none"> Design principles 	-	-	A/I	R	I	-	-	-	
Establish technical governance model	Ensure robust and consistent development	None specified	<ul style="list-style-type: none"> Governance model 	-	-	A/I	R	I	-	-	-	
Conduct design review	Ensure solution meets satisfies customer’s design principles and satisfies business needs	None specified	<ul style="list-style-type: none"> Approved design 	C	C	A/I	R	I	I	-	-	
Mitigate key technical risks	Ensure “show-stoppers” are dealt with early	None specified	Depends on the nature of the risks.	-	-	A/I	R	R	-	-	-	
Build business critical functionality (including supporting features, e.g., interfaces)	Early risk mitigation	None specified	<ul style="list-style-type: none"> Business critical functionality 	-	-	A/I	R	R	-	-	-	
Unit and Function test business critical functionality	Early risk mitigation and quality assurance	<ul style="list-style-type: none"> Test Management Framework Any standard testing tools “Show and Tell” 	<ul style="list-style-type: none"> Business critical functionality 	-	-	A/I	I	R	-	-	-	



ELABORATION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Build critical technical requirements	Early risk mitigation.	None specified	<ul style="list-style-type: none"> Technically critical functionality 	-	-	A/I	R	R	-	-	-
Unit and Function test critical technical requirements	Early risk mitigation and quality assurance.	<ul style="list-style-type: none"> Test Management Framework Any standard testing tools "Show and Tell" 	<ul style="list-style-type: none"> Technically critical functionality 	-	-	A/I	I	R	-	-	-
Track and fix defects	Quality assurance	<ul style="list-style-type: none"> Test Management Framework Any approved testing tools "Show and Tell" 	<ul style="list-style-type: none"> Test results 	I	C	A/R	I	R	R	-	-
Define testing strategy	Prepare for writing test scripts and test environment building.	None specified	<ul style="list-style-type: none"> Testing strategy 	-	-	A/I	-	-	R	-	-
Define deployment strategy	Ensure strategy is defined in advance of deployment to other environments.	None specified	<ul style="list-style-type: none"> Deployment strategy 	-	-	A/I	R	I	-	-	-
Start producing training material	Ensure training material is ready in time for Transition phase.	None specified	<ul style="list-style-type: none"> Draft training material 	C	C	A/I	-	-	-	R	-
Re-estimate remaining analysis effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Analysis estimates 	I	R	A/I	I	-	-	-	-
Re-estimate remaining development effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Development estimates 	I	-	A/I	R	-	-	-	-
Re-estimate remaining testing effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Testing estimates 	I	-	A/I	-	-	R	-	-
Re-estimate deployment effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Deployment estimates 	I	-	A/I	R	-	-	-	-
Re-estimate remaining training effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Training estimates 	I	-	A/I	I	-	-	R	-
Complete phase readiness checklist	Ensure that phase tasks have been satisfactorily completed	None specified	<ul style="list-style-type: none"> Phase readiness checklist 	I	R	A/R	R	C	R	R	-
Produce revised project/release plan	Prepare for "Go/No Go" decision for the Construction phase	<ul style="list-style-type: none"> Any project planning tool PMF 	<ul style="list-style-type: none"> Project plan 	I	C	A/R	C	I	C	C	-



ELABORATION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Present revised project/release plan to stakeholders	Obtain "Go/No Go" decision for the Construction phase	<ul style="list-style-type: none"> Any project planning tool PMF 	<ul style="list-style-type: none"> Go/No Go decision 	C	I	A/R	I	I	I	I	I

5.15. Construction Phase

5.16. Phase Objectives (High Level Tasks)

- Complete requirements definition for medium and low business priorities
- Complete system build
- Build testing and training environments
- Complete preparing training materials

5.17. Example Pre-requisites

- All Elaboration tasks listed in section 1.1 are complete.

5.18. Medium Level Tasks

CONSTRUCTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Complete requirements definition	Define 100% of requirements by end of Construction	<ul style="list-style-type: none"> Workshops "Show and Tell" Use Case Rule Requirement Rule Visio Word Excel 	<ul style="list-style-type: none"> Approved low-level requirements 	C	R	A/I	C	C	C	I	-
Build remaining business functionality (including supporting features, e.g., interfaces)	Complete system build	None specified	<ul style="list-style-type: none"> All business functionality 	-	-	A/I	R	R	-	-	-



CONSTRUCTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Unit and Function test remaining business functionality	Complete system build	<ul style="list-style-type: none"> Test Management Framework Any standard testing tools "Show and Tell" 	<ul style="list-style-type: none"> All business functionality 	-	-	A/I	I	R	-	-	-
Build remaining technical requirements	Complete system build	None specified	<ul style="list-style-type: none"> All technical functionality 	-	-	A/I	I	R	-	-	-
Unit and Function test remaining technical requirements	Complete system build	<ul style="list-style-type: none"> Test Management Framework Any standard testing tools "Show and Tell" 	<ul style="list-style-type: none"> All technical functionality 	-	-	A/I	I	R	-	-	-
Track and fix defects	Quality assurance	<ul style="list-style-type: none"> Test Management Framework Any approved testing tools "Show and Tell" 	<ul style="list-style-type: none"> Test results 	C	C	A/I	I	R	R	-	-
Build automated testing harness	Preparation for Transition phase test execution	<ul style="list-style-type: none"> Any approved automated testing tools 	<ul style="list-style-type: none"> Automated testing harness 	-	C	A/I	C	I	R	-	-
Build test scripts for automated testing	Preparation for Transition phase test execution	<ul style="list-style-type: none"> Any approved automated testing tools 	<ul style="list-style-type: none"> Automated test scripts 	-	C	A/I	I	I	R	-	-
Build testing and training environments	Preparation for Transition phase test execution	None specified	<ul style="list-style-type: none"> Testing and training environments 	I	-	A/I	R	R	C	-	-
Complete training materials	Ensure training material is ready in time for Transition phase.	None specified	<ul style="list-style-type: none"> Training material 	C	C	A/I	-	-	-	R	-
Re-estimate remaining testing effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Testing estimates 	I	-	A/I	-	-	R	-	-
Re-estimate deployment effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Deployment estimates 	I	-	A/I	R	-	-	-	-
Re-estimate remaining training effort	Contribute to project plan	None specified	<ul style="list-style-type: none"> Training estimates 	I	-	A/I	I	-	-	R	-
Complete phase readiness checklist	Ensure that phase tasks have been satisfactorily completed	None specified	<ul style="list-style-type: none"> Phase readiness checklist 	I	R	A/R	R	C	R	R	-



CONSTRUCTION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Produce revised project/release plan	Prepare for “Go/No Go” decision for the Transition phase	<ul style="list-style-type: none"> Any project planning tool PMF 	<ul style="list-style-type: none"> Project plan 	I	C	A/R	C	C	C	C	-
Present revised project/release plan to stakeholders	Obtain “Go/No Go” decision for the Transition phase	<ul style="list-style-type: none"> Any project planning tool PMF 	<ul style="list-style-type: none"> Go/No Go decision 	C	I	A/R	I	I	I	I	I

5.19. Transition Phase

5.20. Phase Objectives (High Level Tasks)

- Deploy system into testing and training environments
- Complete system-wide testing (e.g., performance, integration, user acceptance)
- Complete preparing training materials
- Deliver training to end users
- Deliver training to maintenance team

5.21. Example Pre-requisites

- All Construction tasks listed in section 5.18 are complete.

5.22. Medium Level Tasks

TRANSITION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Deploy to testing and training environments	Enable test execution and training	None specified	Software deployed to testing and training environments	I	-	A/I	R	R	I	I	
Complete system-wide testing	Quality assurance	<ul style="list-style-type: none"> Any approved testing tools 	<ul style="list-style-type: none"> Test results 	C	C	A/I	C	C	R	-	-



TRANSITION PHASE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Track and fix defects	Quality assurance	<ul style="list-style-type: none"> • Test Management Framework • Any approved testing tools • “Show and Tell” 	<ul style="list-style-type: none"> • Test results 	C	C	A/I	I	R	R	-	-
Deliver training to end users	Business readiness	None specified	<ul style="list-style-type: none"> • The training itself 	C	-	A/I	-	-	-	R	
Deliver training to maintenance team	System support readiness	None specified	<ul style="list-style-type: none"> • The training itself 	C	-	A/I	R	R	-	-	I
Complete phase readiness checklist	Ensure that phase tasks have been satisfactorily completed	None specified	<ul style="list-style-type: none"> • Phase readiness checklist 	I	I	A/R	R	R	R	R	-
Produce revised project/release plan	Prepare for “Go/No Go” decision for Go Live	<ul style="list-style-type: none"> • Any project planning tool • PMF 	<ul style="list-style-type: none"> • Project plan 	I	C	A/R	C	I	C	C	-
Present revised project/release plan to stakeholders	Obtain “Go/No Go” decision for Go Live	<ul style="list-style-type: none"> • Any project planning tool • PMF 	<ul style="list-style-type: none"> • Go/No Go decision 	C	I	A/R	I	I	I	I	I

5.23. Go Live

5.24. Phase Objectives (High Level Tasks)

- Deploy system into production environment
- Complete handover to maintenance team
- Monitor production system

5.25. Example Pre-requisites

- All Transition tasks listed in section 5.22 are complete.



5.26. Medium Level Tasks

GO LIVE											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
Migrate data into Production environment	Production readiness	None specified	<ul style="list-style-type: none">Migrated data	C	-	A/I	R	R	-	-	I
Deploy application into Production environment	Production readiness	None specified	<ul style="list-style-type: none">Deployed application	C	-	A/I	R	R	-	-	I
Perform handover to maintenance team	Production readiness	None specified	<ul style="list-style-type: none">All related business and technical documentation	I	-	A/I	R	R	-	-	I
Monitor system to identify process improvements, enhancements, etc.	Ongoing improvement	None specified	<ul style="list-style-type: none">Improvement proposals	C	C	-	-	-	-	-	A/R
Maintain application	Ongoing improvement	None specified	<ul style="list-style-type: none">Enhancements, fixes, etc.	C	-	-	-	-	-	-	A/R