

An Alternative to PegaBPM

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

by Declan Chellar

Version 1.4

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. USING PEGA BPM WITH PRPC..... 7

 3.1. SmartBPM Methodology: Project Initiation 7

 3.2. SmartBPM Methodology: Inception Phase..... 8

 3.3. SmartBPM Methodology: Elaboration Phase 11

 3.4. SmartBPM Methodology: Construction Phase 14

 3.6. SmartBPM Methodology: Transition Phase..... 16

 3.8. SmartBPM Methodology: Go Live..... 18

4. REVIEW OF SMARTBPM CRITICAL FLAWS..... 19

 4.1. Not insisting on establishing business objectives during Project Initiation..... 19

 4.2. Using the Inception phase to investigate current processes..... 19

 4.3. Not addressing the mitigation of key technical risks..... 19

 4.4. Not addressing the need to build testing environments during Construction..... 19

 4.5. Iterating between Elaboration and Construction 19

5. AN ALTERNATIVE TO SMARTBPM 21

 5.1. Key principles 21

 5.2. Roles..... 22

 5.3. Project Initiation 23

 5.3.1. Project Initiation Phase Objectives (High Level Tasks)..... 23

 5.3.2. Project Initiation Minimum Pre-requisites 23

 5.3.3. Project Initiation Medium Level Tasks 23

 5.4. Inception Phase..... 26

- 5.4.1. Inception Phase Objectives (High Level Tasks) 26
- 5.4.2. Inception Phase Minimum Pre-requisites..... 26
- 5.4.3. Inception Phase Medium Level Tasks 26
- 5.5. Elaboration Phase 30
- 5.5.1. Phase Objectives (High Level Tasks) 30
- 5.5.2. Pre-requisites 30
- 5.5.3. Medium Level Tasks 30
- 5.6. Construction Phase 34
- 5.6.1. Phase Objectives (High Level Tasks) 34
- 5.6.2. Example Pre-requisites 34
- 5.6.3. Medium Level Tasks 34
- 5.7. Transition Phase 37
- 5.7.1. Phase Objectives (High Level Tasks) 37
- 5.7.2. Example Pre-requisites 37
- 5.7.3. Medium Level Tasks 37
- 5.8. Go Live..... 39
- 5.8.1. Phase Objectives (High Level Tasks) 39
- 5.8.2. Example Pre-requisites 39
- 5.8.3. Medium Level Tasks 39

1. INTRODUCTION

1.1. Purpose

The author aims to provide an appraisal of the Pega Developer Network articles “[What is the SmartBPM Methodology?](#)”¹ and “[Using Pega BPM with PRPC](#)”², based on his experiences working on Pega projects in the field. Furthermore, the author will provide an alternative guide, which will address any perceived shortcomings of the PDN article and the SmartBPM methodology.

1.2. Scope

This document addresses the SmartBPM (now known as PegaBPM) 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 PEGA 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 Pega. 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 PEGA 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)
- Certified Pega Business Architect (7.2)
- Pegasystems Certified Methodology Black Belt
- BPMN 2.0 Master Class Certified

Declan writes about his business analysis experiences on his blog “[Analysis Fu](#)”³ and tweets about business analysis as [@AnalysisFu](#).

¹ <https://pdn.pega.com/implementation-methodology-overview/implementation-methodology-overview#WhatIsSB>

² <https://pdn.pega.com/using-pega-bpm-prpc/using-pega-bpm-prpc>

³ <http://www.chellar.com/AnalysisFu>

2. PERCEIVED AREAS FOR IMPROVEMENT

Note that Pega clearly states that the SmartBPM methodology “is **not** a concrete prescriptive process” and that it is adaptable to any size project. Therefore some of the author’s suggestions are nothing more than an adaptation 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 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. 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.



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. USING PEGA BPM WITH PRPC⁴

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 criticality 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

3.1.1. General Comments on Project Initiation

- Missing activity: going through a Phase Readiness checklist to ensure everything is in place to begin Project Initiation.
- Missing activity: going through a Phase Completion checklist to ensure everything is in place to exit Project Initiation.
- Considering the importance Pega rightly places on understanding and capturing Business Objectives in their “Business Analysis Essentials” training course, the author would have expected a review of the Business Case and Business Objectives to be a key activity of the Project Initiation phase, yet it is not. **The author regards Pega’s lack of emphasis on establishing business objectives during Project Initiation to be a key weakness.**
-

3.1.2. Project Initiation Suggested Activities

- *Preliminary discussions regarding the scope of the over-all program (or project)*
- *Accomplish walk-throughs of the current process; include all parties and review all applicable systems*
- *Determine quick wins for the client and decide which slivers are priorities*
- *Conduct knowledge transfer sessions and begin enablement training*

3.1.3. Comments on Project Initiation Suggested Activities:

Accomplish walk-throughs of the current process

Comment:

- Pega’s assumption seems to be that process improvement will happen during software development. However, PEGA 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, PEGA 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.

⁴ <https://pdn.pega.com/using-pega-bpm-prpc/using-pega-bpm-prpc>

Determine quick wins for the client and decide which slivers are priorities

Comment:

- The phrase “quick wins” can lead people to prioritise what can be developed quickly, which is often the easy, low value, business functionality. Instead, you should be thinking in terms of “business priorities” and “high business value”.
- 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, this task does not belong in the Project Initiation phase. In fact, it is only *after* Inception that we have enough information to be able to plan slivers.

3.1.4. Project Initiation Key Deliverables

- *Proof of Concepts*
- *Workshops with demos of current state versus targeted process improvements and process re-engineering*
- *PEGA Enablement training courses*
- *Project Roadmap planning; identify critical milestones, staffing model proposals, risks and assumptions*

3.1.5. Comments on Project Initiation Key Deliverables

Proof of Concepts

Comment:

- *It is odd that there is no suggested activity that would deliver a PoC.*
- *It would be hard to produce a PoC when only **preliminary** scope discussions have been had and when requirements have not been discussed yet.*

Workshops with demos of As Is versus To Be

Comment:

- “Workshops” are not deliverables. They are a medium for performing activities that produce deliverables.

Project Roadmap planning; identify critical milestones, staffing model proposals, risks and assumptions

- *“Project Roadmap” is a deliverable, planning is an activity.*

3.2. SmartBPM Methodology: Inception Phase

3.2.1. General Comments on Inception Phase

- Missing activity: going through a Phase Readiness checklist to ensure everything is in place to begin Inception.
- Missing activity: going through a Phase Completion checklist to ensure everything is in place to exit Inception.
- Despite Pega’s definition, the Inception phase is not the appropriate time for process improvement. Formal process improvement results in a To Be business architecture, which should be in place before PEGA is even chosen as a technical solution. When a business change initiative is so immature that the client does not know what its To Be business architecture 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, PEGA 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.**

- 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.
- If a Proof of Concept is required, it can only be done after the Inception Phase (once high-level requirements are understood) and before the Elaboration Phase.

3.2.2. Inception Phase Suggested Activities

- *Define the scope of the over-all program, project, or specific sliver*
- *Define the business requirements and organize them into projects and then logical slivers of deliverables*
- *Confirm the sliver and targeted deliverables*
- *Capture the high-level requirements, building out the use cases and work types for the sliver*
- *Perform operational walk-throughs to identify key process improvements with measurable targets*
- *Estimate the effort with the teams involved*

3.2.3. Comments on Inception Phase Suggested Activities

<p><i>Define the scope of the over-all program, project, or specific 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 the business requirements and organize them into projects and then logical slivers of deliverables</i></p> <p>Comment:</p> <ul style="list-style-type: none"> • The business requirements should be defined by the Business before Pega was even chosen as a technology. The task here should be to <i>record</i> those requirements. • Slivers can only be defined once the program/project scope has been understood, i.e., after Inception.
<p><i>Confirm the sliver and targeted deliverables</i></p> <ul style="list-style-type: none"> • Slivers can only be defined once the program/project scope has been understood, i.e., post Inception.

Document high-level requirements, build use cases [Specifications] and work types [Case Types]

Comment:

- There are three activities here.
- Use Case level should be defined here (there are four levels):
 - Business UC (includes manual and non-Pega activities)
 - Software UC (includes only interactions between the User and the Pega Application)
 - Atomic UC (individual atomic tasks in a Pega Flow)
 - Technical UC (how the LSA proposes to implement a particular SUC or AUC)
- Use Cases are not appropriate for BRE projects.
- Case Types are actually part of the high-level design, since identifying work types affects the class structure. Make sure this task is assigned to the LSA and not the LBA.

Perform operational walk-throughs to identify key process improvements with measurable targets

Comment:

- 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.

Estimate the effort with the teams involved

Comment:

- Project Managers of tackle this task alone. A RACI would be 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.

3.2.4. Inception Phase Key Deliverables

- *Inventory and analyze what artifacts already exist, identifying reuse and leveraging existing artifacts*
- *Initiate DCO sessions (conduct either prep & review, white board & review, or real-time capture DCO sessions)*
- *Enter the DCO session artifacts directly into the Application Profiler for each sliver*
- *As the Application Profile is built-out, use the Sizing Tool in concert to size the effort*
- *Review the Application Profile & and project sizing results with stakeholders*
- *Leverage the built-in Project Plan Template for a high-level draft of the initial project plan*
- *Establish the Project Management Framework (PMF) and Test Management Framework (TMF) for use with the project*
- *Update the Resource Plan and the Risk and Assumptions with any changes*
- *Complete a Phase Readiness checklist*
- *Perform a Methodology Alignment Workshop, if blending Pega BPM best practices and DCO with a client mandated methodology*
- *Hold a project kick-off meeting with resulting drafts of the Application Profile, project plan, resource plan and project sizing*

3.2.5. 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 reuse*
- *Initiate DCO sessions*
- *Enter artifacts into the Application Profiler*
- *Use the sizing tool*

3.3. SmartBPM Methodology: Elaboration Phase

3.3.1. General Comments on Elaboration Phase

- Missing activity: going through a Phase Readiness checklist to ensure everything is in place to begin Elaboration.
- Missing activity: going through a Phase Completion checklist to ensure everything is in place to exit Elaboration.
- 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.**
- If a Proof of Concept is required, it can only be done after the Inception Phase (once high-level requirements are understood) and before the Elaboration Phase.

3.3.2. Elaboration Phase Suggested Activities

- *Complete the detailed DCO sessions (conduct either prep & review, white board & review, or real-time capture DCO sessions)*
- *Capture requirement details, further building out the Application Profile with atomic use cases and work types*
- *Work iteratively, cycling through Elaboration and Construction phases*
- *Build out the foundation of the proposed application completing the Application Profile and running the Application Accelerator*
- *Environment setup, planning and execution*
- *Clarify and establish a governance model if it is not already present*
- *Establish the Deployment Plan/ Process, the Test Plan, and Issue Tracking Process*
- *Use the Pre-Flight and PAL tools*

3.3.3. Comments on Elaboration Phase Suggested Activities

Complete the detailed 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.

Capture requirement details, further building out the Application Profile with 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.

Build out the foundation of the proposed application completing the Application Profile and running the Application Accelerator

Comment:

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

Environment setup, planning and execution

Comment:

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

Clarify and establish a governance model if it is not already present

Comment:

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

Establish the Deployment Plan/Process, the Test Plan, and 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.

3.3.4. Elaboration Phase Key Deliverables

- *Complete the Application Profile with the atomic use cases detailed*
- *Produce the automated document generation from the Application Profile and update the sizing effort*
- *Run the Application Accelerator to establish the application foundation*
- *Create the draft flows and draft UI*
- *Establish the links between the requirements, cases and rules*
- *Engage PMF and TMF established in the previous phase*
- *Create the detailed project plan*
- *Develop a testing plan, migration plan and establish standards for executing the plans if they do not yet exist*
- *Lead a change management review to determine if requirements or priorities have changed*
- *Updated Deployment Plan and Test Plan*
- *Work iteratively and review work to-date*

3.3.5. Comments on Elaboration 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. 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

3.4.1. General Comments on Construction Phase

- Missing activity: going through a Phase Readiness checklist to ensure everything is in place to begin Construction.
- Missing activity: going through a Phase Completion checklist to ensure everything is in place to exit Construction.
- 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.**

3.4.2. Construction Phase Suggested Activities

- *Assign application configuration tasks using the iterative model*
- *Unit test each component*
- *Build test scripts for automated testing*
- *Run application test scripts and begin the system test process*
- *Complete integration system testing*
- *Fully use PMF and TMF established during Elaboration to track progress and triage defects*
- *Hold a Design review or User Experience review*
- *Review PMF and start preparations for the next sliver*
- *Lead a change management review to determine if requirements or priorities have changed*
- *Use the Pre-Flight and PAL tools*

3.4.3. Comments on Construction Phase Suggested Activities

<i>Assign application configuration tasks using the iterative model</i>
Comment: <ul style="list-style-type: none"> • Pega has traditionally assigned 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 each component</i>
Comment: <ul style="list-style-type: none"> • Pega has traditionally assigned this task to the developer role and the QA role. However, the QA team should not be responsible for unit testing.
<i>Build test scripts for automated testing</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 application test scripts and begin the 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>Fully use PMF and TMF established during Elaboration to track progress and triage defects</i>
Comment: <ul style="list-style-type: none"> • Inadequate separation of activities and tools.

Hold a Design review or User Experience review

Comment:

- This is not a single activity.
- The purpose of these reviews should be stated.
- The design review should be held during Elaboration.
- User Experience reviews should begin during Elaboration.

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.

Lead a change management review to determine if requirements or priorities have changed

Comment:

- This should be a regular activity throughout Elaboration and Construction at the end of each iteration.

Use the Pre-Flight and PAL tools

Comment:

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

3.4.4. Construction Phase Key Deliverables

- *Using PMF, assign the work from the DCO sessions and tasks from the project plan*
- *Resources complete the assigned work/ unit tests*
- *Build a Release Candidate meeting the requirements of the sliver*
- *Complete Unit/ Integration testing and associated Issue tracking Process*
- *Complete all iterative automated testing*
- *Engage TMF to test, report on and ensure rule quality*
- *Build and test migration plans*
- *Design review and user experience review with stakeholders*
- *End-user training, production deployment and Support guides, as needed*
- *Update any requirements and regenerate documentation using the Application Profile*
- *Performance, Pre-Flight, PAL, and log file analysis*
- *Updated project plan*
- *Complete a Phase Readiness checklist*

3.4.5. Comments on Construction 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. 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*

3.6. SmartBPM Methodology: Transition Phase

3.5.1. General Comments on Transition Phase

- Missing activity: going through a Phase Readiness checklist to ensure everything is in place to begin Transition.
- Missing activity: going through a Phase Completion checklist to ensure everything is in place to exit Transition.

3.5.2. Transition Phase Suggested Activities

- *Environment migrations completed for QA and UAT*
- *QA and UAT testing completed*
- *Automated testing cycles completed*
- *Issue Tracking process completed with TMF review*
- *Participate in the system testing and UAT process to ensure the application is acceptable to stakeholders and business users*
- *Ensure solution is maintainable by IT, including Support*
- *Review PMF and finalize plans for the next sliver*
- *Formalize stakeholder sign-offs and readiness*
- *Revisit target metrics and measurements for success*
- *End-user training, rollout and support plans completed*

3.5.3. Comments on Transition Phase Suggested Activities

Ensure solution is maintainable by IT, including Support

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).

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.

3.5.4. Transition Phase Key Deliverables

- *Participate in Go/ No-Go status update review after formalized stakeholder sign-offs*
- *Migration Delivery documents and production roll-out run book completed*
- *Production Support plan and Support training completed*
- *End-user training completed*
- *Issue tracking and resolutions should be signed-off*
- *Review reports, what metrics will be gathered and how follow-up will occur*
- *Completed QA/automated testing and UAT testing*
- *Performance, Pre-Flight, PAL, and log file analysis should be signed-off*
- *Complete a Phase Readiness checklist*

3.5.5. 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*

3.8. SmartBPM Methodology: Go Live

3.5.6. General Comments on Go Live

No comments.

3.5.7. Go Live Suggested Activities

- *Data migrated into Production*
- *Application deployment*
- *Go-Live support activities*

3.5.8. Comments on Go Live Suggested Activities

No comments.

3.5.9. Go Live Key Deliverables

- *Deploy final release candidate*
- *Maintain application with enhancements, fixes, and change requests by production support and business users*
- *Production Support/ Documentation updates, as needed*
- *Monitor system using reporting tools, identify and capture process improvements and related requirements*
- *Plan and deliver a status update to all stakeholders after an established period for metrics to be adequately reported on*

3.5.10. Comments on Go Live Key Deliverables

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*

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.

4.1. Not insisting on establishing business objectives during Project Initiation

The author has worked on Pega projects where not only were business objectives not documented in Pega but no member of the development team could say what they were nor where they were documented. Since Pega's DCO tools do not make it mandatory to capture business objectives, nor mandatory to assign a primary business objective, it is left to team members to exercise due diligence, which they often do not.

4.2. 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 architecture, which should be in place before PEGA is even chosen as a technical solution. When a business change initiative is so immature that the client does not know what its To Be business architecture 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 Pega is not a process re-engineering tool; it is a process implementation tool.

4.3. 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".

4.4. 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.

4.5. Iterating between Elaboration and Construction

Elaboration and Construction are distinct phases in a project and each has its own objectives (see sections 5.5.1 and 5.6.1). 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 to work iteratively, they must continuously cycle between those two phases. This misunderstanding seems to be represented in one of Pega's illustrations⁵, which is represented here:

⁵ <https://pdn.pega.com/implementation-methodology-overview/implementation-methodology-overview#WhatIsSB>

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

This illustration misrepresents the nature of iteration.

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.



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 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 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 • 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 • Pega specialist
PM	Project Manager	This role may be in charge of an entire programme cutting across technology teams or in charge of the Pega effort only.
LSA	Lead Systems Architect	A Pega-certified Lead Systems Architect
SA	System Architect	A developer conversant with Pega
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

5.3.1. Project Initiation Phase Objectives (High Level Tasks)

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

5.3.2. Project Initiation Minimum Pre-requisites

- Business case
- SMART Business objectives⁶
- High-level To Be business process models (low-level, tested models of the lifecycle of each “Case Type” would be better, preferably using BPMN 2.0)
- A clear, unambiguous, internally consistent business taxonomy
- Business role definitions
- High-level technical solution architecture

5.3.3. Project Initiation Medium Level Tasks

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				-			-
Present business objectives	Ensure all stakeholders understand what success looks like	None specified	• The presentation itself	A/R				-			-
Present the To Be business processes	Ensure all stakeholders have a vision of the processes to be implemented	None specified	• The presentation itself	A/R				-			-

⁶ See: “Documenting Strategic Goals” <http://www.chellar.com/AnalysisFu/?p=2223>

PROJECT INITIATION												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Present the high-level technical solution architecture	Ensure all stakeholders understand what is to be implemented using PEGA and what interfaces will be required	None specified	<ul style="list-style-type: none"> The presentation itself 	A/R	I	I	I	-	I	I	-	
Present methodology	Ensure all stakeholders understand the methodology to be used	None specified	<ul style="list-style-type: none"> The presentation itself 	I	I	A/R	I	-	I	I	-	
Enable training (e.g., Pega, process modelling, use case modelling, data modelling)	Ensure all project participants have the necessary skills	None specified	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Staffing model 	C	C	A/R	C	-	C	-	-	
Plan Inception workshops	Ensure necessary time for Inception is planned and participants are available	None specified	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Requirements Traceability Model Requirements Traceability Matrix 	A	R	I	I	I	I	-I	-	

PROJECT INITIATION											
ACTIVITIES				RACI							
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint
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	-
Complete phase completion checklist	Ensure phase tasks have been satisfactorily completed	None specified	Phase readiness checklist	I	R	A/R	R	-	-	-	-

5.4. Inception Phase

5.4.1. Inception 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)

5.4.2. Inception Phase Minimum Pre-requisites

- All Project Initiation tasks listed in section 5.3.3 are complete.

5.4.3. Inception Phase Medium Level Tasks

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 • 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 (Specification) Rule • 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 case types	Enable the LSA to define candidate Pega case types	None specified	<ul style="list-style-type: none"> • Candidate business case types 	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	
Define candidate Pega case types	Gain an initial understanding of the Pega class structure	<ul style="list-style-type: none"> • Pega Case Type 	<ul style="list-style-type: none"> • Documented Pega Case Types 	I	I	A/I	R	I	I	I	I	
Identify interface constraints	Ensure technical risks can be identified	<ul style="list-style-type: none"> • 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"> • 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 business priorities	Ensure business need is understood	<ul style="list-style-type: none"> • Requirement Rule • Use Case (Specification) Rules • Word • Excel • Industry-standard requirements tools 	<ul style="list-style-type: none"> • Documented business priorities 	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 technical risks	Ensure technical risks are understood	<ul style="list-style-type: none"> • Requirement Rule • Use Case (Specification) Rules • 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 back to stakeholders	Confirm that business needs have been understood	<ul style="list-style-type: none"> • Word • PowerPoint • Pega Application Profiler 	<ul style="list-style-type: none"> • Documented business needs • Pega 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	-	-	-	-	

INCEPTION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Estimate testing effort	Contribute to project plan	<ul style="list-style-type: none"> • Pega’s sizing tool • Any industry-standard estimation tools 	• Testing estimates	I	-	A/I	-	-	R	-	-	
Estimate deployment effort	Contribute to project plan	<ul style="list-style-type: none"> • Pega’s sizing tool • Any industry-standard estimation tools 	• Deployment estimates	I	-	A/I	R	-	-	-	-	
Estimate training effort	Contribute to project plan	None specified	• Training estimates	I	-	A/I	I	-	-	R	-	
Complete phase completion checklist	Ensure phase tasks have been satisfactorily completed	None specified	• 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 	• 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 	• Go/No Go decision	C	I	A/R	I	I	I	I	I	

5.5. Elaboration Phase

5.5.1. 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

5.5.2. Pre-requisites

- All Inception tasks listed in section 5.4.3 are complete.

5.5.3. 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 (Specification) 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	-	-	-	

ELABORATION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Produce initial Physical Data Model	Establish initial data architecture	None specified	• Physical Data Model	-	-	A/I	R	I	-	-	-	
Build Database	Establish initial data architecture	None specified	• Database	-	-	A/I	R	I	-	-	-	
Define design principles	Ensure robust and consistent development	None specified	• Design principles	-	-	A/I	R	I	-	-	-	
Establish technical governance model	Ensure robust and consistent development	None specified	• Governance model	-	-	A/I	R	I	-	-	-	
Conduct design review	Ensure solution meets satisfies customer’s design principles and satisfies business needs	None specified	• 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	• Business critical functionality	C	C	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” 	• Business critical functionality	C	C	A/I	I	R	-	-	-	
Build critical technical requirements	Early risk mitigation.	None specified	• Technically critical functionality	-	-	A/I	R	R	-	-	-	

ELABORATION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
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” 	• 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” 	• Test results	I	C	A/R	I	R	R	-	-	
Define testing strategy	Prepare for writing test scripts and test environment building.	None specified	• Testing strategy	C	-	A/I	C	-	R	-	-	
Define deployment strategy	Ensure strategy is defined in advance of deployment to other environments.	None specified	• Deployment strategy	C	-	A/I	R	I	-	-	-	
Start producing training material	Ensure training material is ready in time for Transition phase.	None specified	• Draft training material	C	C	A/I	-	-	-	R	-	
Re-estimate remaining analysis effort	Contribute to project plan	None specified	• Analysis estimates	I	R	A/I	I	-	-	-	-	
Re-estimate remaining development effort	Contribute to project plan	None specified	• Development estimates	I	-	A/I	R	-	-	-	-	
Re-estimate remaining testing effort	Contribute to project plan	None specified	• Testing estimates	I	-	A/I	-	-	R	-	-	
Re-estimate deployment effort	Contribute to project plan	None specified	• Deployment estimates	I	-	A/I	R	-	-	-	-	

ELABORATION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Re-estimate remaining training effort	Contribute to project plan	None specified	• Training estimates	I	-	A/I	I	-	-	R	-	
Complete phase readiness checklist	Ensure phase tasks have been satisfactorily completed	None specified	• 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 	• Project plan	I	C	A/R	C	I	C	C	-	
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 	• Go/No Go decision	C	I	A/R	I	I	I	I	I	

5.6. Construction Phase

5.6.1. 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

5.6.2. Example Pre-requisites

- All Elaboration tasks listed in section 5.5.3 are complete.

5.6.3. 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 (Specification) 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 	C	C	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” 	• All business functionality	C	C	A/I	I	R	-	-	-	
Build remaining technical requirements	Complete system build	None specified	• 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” 	• 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” 	• Test results	C	C	A/I	I	R	R	-	-	
Build automated testing harness	Preparation for Transition phase test execution	• Any approved automated testing tools	• Automated testing harness	-	C	A/I	C	I	R	-	-	
Build test scripts for automated testing	Preparation for Transition phase test execution	• Any approved automated testing tools	• Automated test scripts	-	C	A/I	I	I	R	-	-	
Build testing and training environments	Preparation for Transition phase test execution	None specified	• Testing and training environments	I	-	A/I	R	R	C	-	-	

CONSTRUCTION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Complete training materials	Ensure training material is ready in time for Transition phase.	None specified	• Training material	C	C	A/I	-	-	-	R	-	
Re-estimate remaining testing effort	Contribute to project plan	None specified	• Testing estimates	I	-	A/I	-	-	R	-	-	
Re-estimate deployment effort	Contribute to project plan	None specified	• Deployment estimates	I	-	A/I	R	-	-	-	-	
Re-estimate remaining training effort	Contribute to project plan	None specified	• Training estimates	I	-	A/I	I	-	-	R	-	
Complete phase readiness checklist	Ensure phase tasks have been satisfactorily completed	None specified	• Phase readiness checklist	I	R	A/R	R	C	R	R	-	
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 	• 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 	• Go/No Go decision	C	I	A/R	I	I	I	I	I	

5.7. Transition Phase

5.7.1. 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

5.7.2. Example Pre-requisites

- All Construction tasks listed in section 5.6.3 are complete.

5.7.3. 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 	• Test results	C	C	A/I	C	C	R	-	-
Track and fix defects	Quality assurance	<ul style="list-style-type: none"> • Test Management Framework • Any approved testing tools • “Show and Tell” 	• Test results	C	C	A/I	I	R	R	-	-
Deliver training to end users	Business readiness	None specified	• The training itself	C	-	A/I	-	-	-	R	

TRANSITION PHASE												
ACTIVITIES				RACI								
Task	Objective	Tool Options	Deliverable	Cust	BA	PM	LSA	SA	QA	Train	Maint	
Deliver training to maintenance team	System support readiness	None specified	• The training itself	C	-	A/I	R	R	-	-	I	
Complete phase readiness checklist	Ensure phase tasks have been satisfactorily completed	None specified	• 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 	• 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 	• Go/No Go decision	C	I	A/R	I	I	I	I	I	

5.8. Go Live

5.8.1. Phase Objectives (High Level Tasks)

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

5.8.2. Example Pre-requisites

- All Transition tasks listed in section 5.7.3 are complete.

5.8.3. 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	• Migrated data	C	-	A/I	R	R	-	-	I
Deploy application into Production environment	Production readiness	None specified	• Deployed application	C	-	A/I	R	R	-	-	I
Perform handover to maintenance team	Production readiness	None specified	• All related business and technical documentation	I	-	A/I	R	R	-	-	I
Monitor system to identify process improvements, enhancements, etc.	Ongoing improvement	None specified	• Improvement proposals	C	C	-	-	-	-	-	A/R
Maintain application	Ongoing improvement	None specified	• Enhancements, fixes, etc.	C	-	-	-	-	-	-	A/R