

Pega Certified DCO Architect

Test Competencies and Topics

Test Domains	% of Exam
Project Planning and Management	10%
Methodology Adoption and Selection	24%
DCO Tools and Best Practices	32%
Conducting DCO Sessions	20%
Project Governance and Oversight	14%
Total	100%

Figure 1 From the DCO Architect Blueprint, August 2015

Table of Contents

- Project Planning and Management..... 3
 - Prepare an iterative program roadmap 3
 - Staff a Pega project..... 3
 - Prepare and conduct kickoff meetings 5
- Methodology Adoption and Selection 6
 - Knows the importance of selecting a methodology that fits 6
 - Knows when and how to use PegaBPM 8
 - Identify the phases of a Scrum project..... 8
 - Knows when and how to use Scrum 10
 - Identify the phases of a Scrum project..... 11
- DCO Tools and Best Practices 15
 - Knows how to use the principles of “Continuous DCO” 15
 - Knows how to gather and document requirements using DCO 15
 - Knows how to write meaningful specifications and iterate their implementation using DCO sessions 16
 - Can provide iterative project sizing estimates 17
- Conducting DCO Sessions 20
 - Knows the best practices for conducting a DCO session 20
 - Conduct DCO:..... 21
 - Knows how to Conduct a “Prep and Review” DCO session 22
 - Can identify the key deliverables of a DCO session 23
- Project Governance and Oversight..... 25

Project Planning and Management

Prepare an iterative program roadmap

- Starting with a well-written, clearly defined problem statement, vision, and project scope ensures that the project team and stakeholders have a common understanding of what is expected.
- Successful programs share some common traits.
- Start with a well-written problem statement and vision.
- Then, establish clear, measurable set of objectives such as “reduce customer complaints by 10%” or “reduce processing time by 2 days.”
- A clearly defined scope is another marker for success.

Program -> Projects -> Release(s)

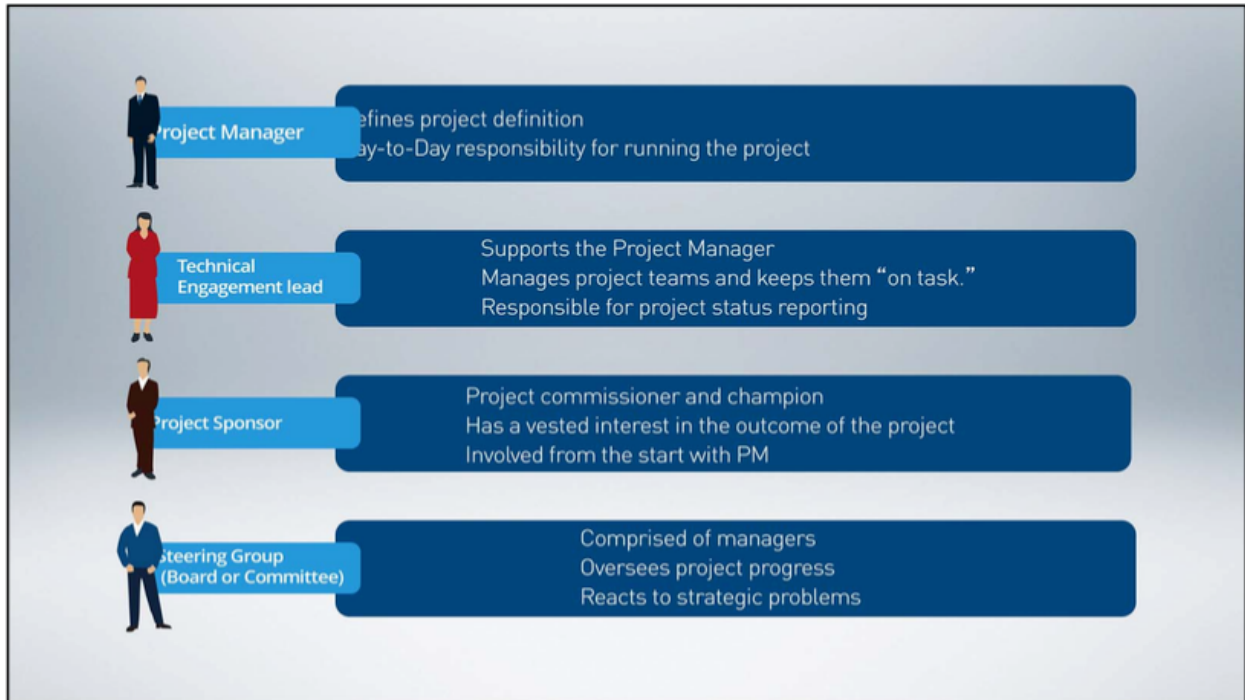
- As a best practice, we want to limit our releases to a length of 90-180 days, or 3-6 months.
- This is the point at which we start to lose the ability to adapt to changes
- It is important to note that if our project can be completed within 6 months, there is no need to divide it into releases.
- This roadmap sequences our projects and releases, and illustrates dependencies and the opportunity for concurrent effort.
- Evaluate the project’s complexity and business visibility
- The intersection of project complexity and business visibility influences our project selection.

Best Practices:

- Organize releases so as to maximize the value delivered to the business.
- We want to identify the functionality that provides the greatest opportunity for reuse or process improvement.
- Keep the delivery cycle short for releases. This allows us to implement functionality before our requirements change too much and begin to erode the project value.
- Define the critical assumptions for the project and establish risk factors. This helps us to identify potential issues before they become problems.
- Training stakeholders in methodology and PRPC tools

Staff a Pega project





- **Process Owners**-senior user form the business area who makes all the key business process decisions
- **SMEs**-consummate knowledge of the operational mechanics of the as-is process and an appreciation for transforming process in order to leverage PRPC. One SME per each major impacted business area.
- **Business Architect**-Ensures that the business requirements, use cases, and objectives are addressed through the implementation lifecycle. Organizes and Schedules DCO Sessions

- **Lead Business Architect**-Facilitates DCO sessions and helps business resources prepare for DCOs. Manages the tasks and priorities of all BAs and ensures artifact quality.

Prepare and conduct kickoff meetings

- Typically, a kickoff meeting lasts the entire workday, with a break for lunch.
- The first part of the meeting – which should last about an hour – represents the proverbial “ribbon cutting ceremony.” The entire team, including senior executives from the business and regional and account executives from Pega and any involved partner, should attend this part of the meeting.
- The second part of the meeting focuses more on the project itself, and need only be attended by the project participants.
- The first topic for the meeting should be a review of the project by the executive sponsor. This allows the sponsor to outline the project’s objectives, business value, and scope to the team – in short, to share the sponsor’s vision with the team.
- Next, we want to introduce the project stakeholders, and outline their roles within the project. This establishes an understanding of the parties responsible for each facet of the project. Be sure to provide contact information for each of the stakeholders as part of the agenda.
- Outline the project structure. This includes a discussion of all recurring meetings such as status and governance meetings – including their schedules and expected participants. We will also want to discuss the project timeline and any major milestones.
- Next, we will want to discuss the implementation methodology we’ll use on the project.
- Discuss the DCO process – especially the process of capturing requirements and use cases, as this is the first part of DCO that the business encounters. This establishes an understanding of DCO tools and practices, and is essential to reinforcing the adoption – and use – of DCO.



- Last, we want to discuss the next steps on the project. This typically includes the set up of the development environment and other necessary systems, establishing any necessary standards and best practices, and the beginning of requirements gathering.
- **The Big Picture**
 - Walkthrough As-Is process, identifying areas for process improvement
 - Once we understand the process, we can begin capturing high-level specifications.
 - We will also want to take an inventory of existing requirements and specifications. If what we need to do is already implemented, we can save valuable time by taking advantage of these existing resources by reusing them.

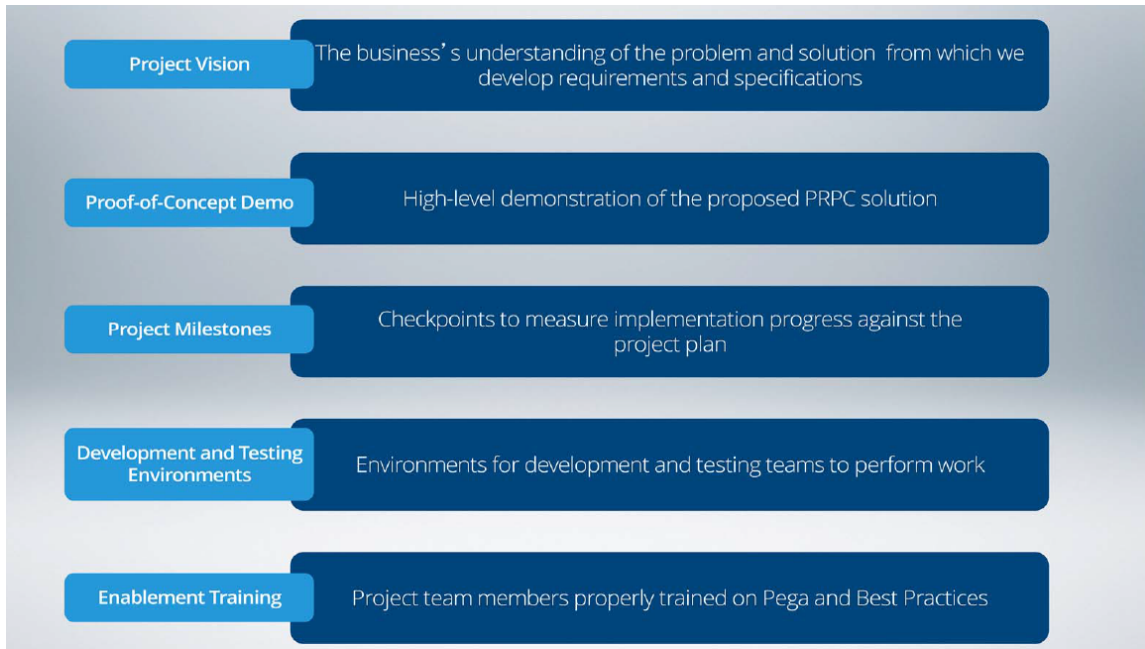


Figure 2 Key Deliverables of the Project Planning Effort

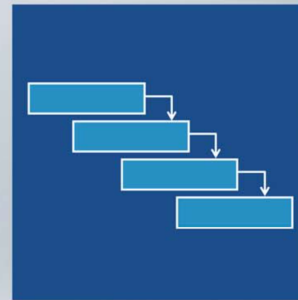
Methodology Adoption and Selection

Knows the importance of selecting a methodology that fits

- The goal of any methodology is to achieve greater efficiency and effectiveness through consistent use of repeatable processes.
- Following a methodology ensures consistency and consistency, while maybe not a guarantee, can certainly “help” produce a more successful project on time, on budget, and with much less risk of failure.
- The goal of any methodology is to achieve greater efficiency and effectiveness through consistent use of repeatable processes.



Loosely defined requirements,
or requirements subject to
frequent change



Detailed requirements

Rigorous change
management necessary

Fully qualified roadmap of
project life-cycle

- Use criteria to select the most appropriate methodology
- First and foremost, when selecting a methodology, choose one that fits the application, the team, and the enterprise.
- This means that some **projects** are best suited to one methodology, while other projects are a better fit with another.
- **Choosing one methodology over another does not necessarily mean that the project risk goes up or down**, or that the project is doomed to fail or destined to succeed.
- It is the interaction between the team and the methodology that will help generate a successful implementation.
- Deploy the most Agile methodology that your implementation team can adopt without increasing risk to an intolerable level.

- **A solution that is not well-defined upfront and has many changing requirements may be better suited for a Scrum/Agile methodology.**
 - This type of methodology aligns well with this type of project because it is designed to handle and adapt to change easily.
 - Use Scrum if your team and organization are open to new experiences and fully understand the disciplines associated with Scrum
- Pega BPM introduces your team to many of the Agile concepts associated with Scrum, while tolerating some of the overhead associated with non-Scrum implementations.
- Waterfall methodologies can handle change, but they are not as agile or receptive to that change. The requirements tend to be locked down upfront, and change is handled through a more rigorous change management process.
 - Enterprises, or solutions, that require more detail up front and a fully qualified roadmap of how to deliver the specifications over the lifecycle of the project – or through multiple projects – tend to follow this type of methodology.

SUPPORTING FACTORS FOR SUCCESS

FACTOR	WHY IT'S IMPORTANT
Enabled Teams	A critical success factor. All team members should be trained on both the technology with which they will be implementing the solution and the implementation methodology
Enterprise Support	Teams that simply take step-by-step direction tend to deliver solutions that may appear to meet the requirements, but don't produce the intended results. In contrast, providing enterprise support empowers your teams to deliver better results. It encourages them to question the intent of requirements in order to ensure their validity and implementation.
Collaborative Development	Teams that work in an isolated manner miss the value of interpretive discussions with and perspectives of others. "Just send over your requirements and I will build it" mentalities have shown the danger of this approach. Collaborative teams work more cohesively towards common goals. The "he said, she said..." excuse-driven process gives way to a "we have decided as a team to..." collaborative approach.
Enterprise Factors	Consider the nature of your business processes when choosing a methodology. Some enterprises may require heavy processes in order to ensure strict compliance levels. While seemingly inefficient or burdensome, they may be necessary to ensure a high quality product. For example, a lifesaving drug or device requires a rigorous testing cycle with little tolerance defects or miscalculations.
Tolerance for Change	Some methodologies and organizations are better suited to adjust to real-time change. Change is inevitable; the methodology and its guiding process dictate how you handle it. Projects that account for change have a greater probability for success than those that do not.

Knows when and how to use PegaBPM

- With Pega BPM, implementations are divided into small scale releases that measure less than 180 days in length, and preferably no more than 90 days.
- Long release cycles result in lost value to the organization and increase the risk of adding more requirements.
- The Pega BPM iterative approach releases slivers of functionality at frequent intervals to incrementally maximize the value gained by the project.
- If team is familiar with Water fall and wants to become more agile
- when the business, and other interested, resources want to stay engaged and play an active role throughout the entire implementation lifecycle.

Identify the phases of a PegaBPM project

Phase	Description
Vision	Assess backlog of strategic initiatives for Risk and Business Value, then prioritize the initiatives, align in Rank Order for delivery, and create an estimated roadmap
Project Initiation	From the highest ranking initiatives, develop program and project list Organize program/projects by scope/budget/timelines into



Participants

Business sponsors
Business architects
SMEs
Executives

Best Practices

Do your homework
Focus on business value and risk
Do the right amount of effort

increments that can be delivered using an Agile/Iterative methodology
Project sizing

Participants	Key Activities
Business sponsors	Operational walkthrough
Business architects	Prioritized project backlog
SMEs	Budgetary estimates for projects
Technical lead	
Executives	

Inception

Focus on an individual project
 clearly define the main objectives, requirements, and high-level specifications of the proposed solution, and confirm the budgetary estimate.
 We will also work to configure the initial stage-based case design.
 Application artifact is created

Best Practices	Key Outcomes
Keep to level-1 application definition	Project assessed
Capture what is needed to understand scope	Operational walkthrough
Don't go too far	Level 1 application
No UI and draft flows only	Application profile documented
Specifications are at level 1 detail	Level of Effort estimate
Go far enough	Phase readiness assessment
Requirements adequately represent business need	
Initial case design (stages and steps)	

Elaboration/Construction

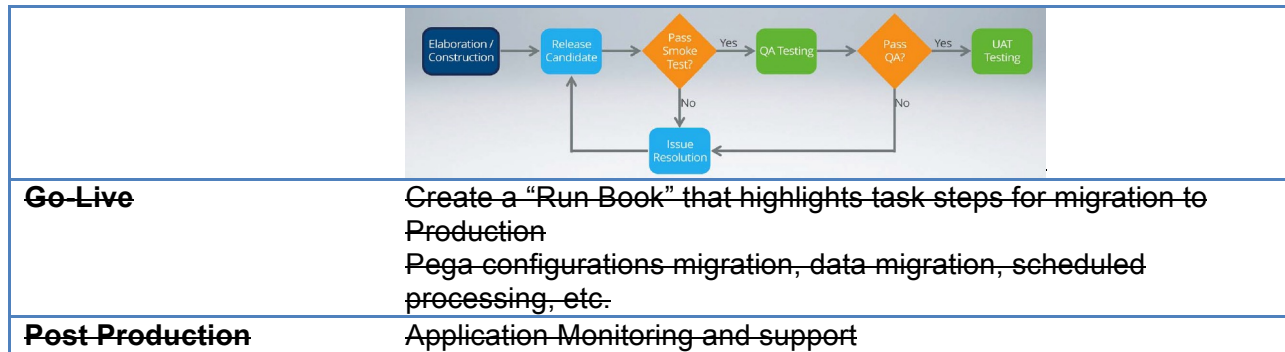
Iterative Design, Build, Test (D/B/T) with Continuous DCO
 Organize specifications into work streams—categorize workstreams [Core, Important, Nice to Have]; then prioritize work streams
 Create iteration plan for D/B/T [Core, then Important, then Nice To Have]



Identify a “Champion” for each work stream and a roadmap for each iteration
 As each work stream goes through DBT, use DCO Elaboration sessions with cross-functional teams to gather more detailed requirements and Playback progress

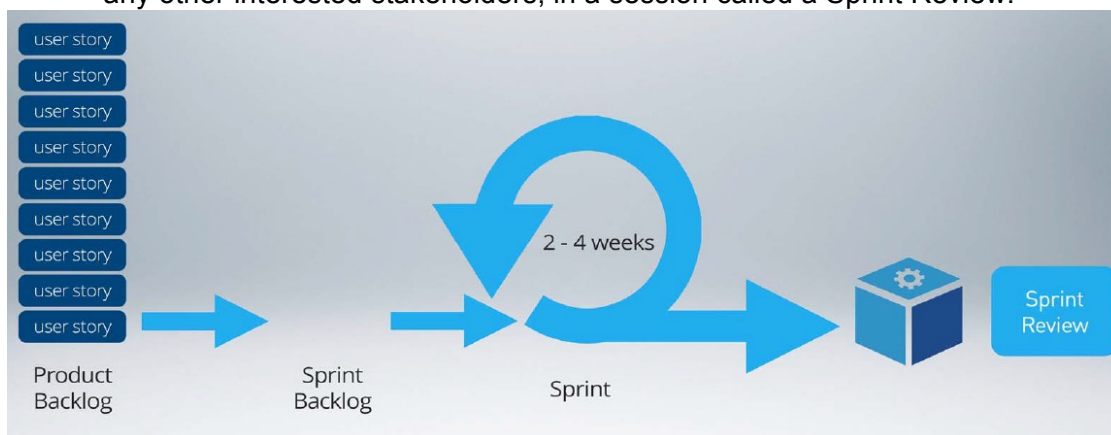
Transition

Fully integrated System, Performance, QA and UAT testing
 Focus on app quality, consistency, standards compliance, and usability



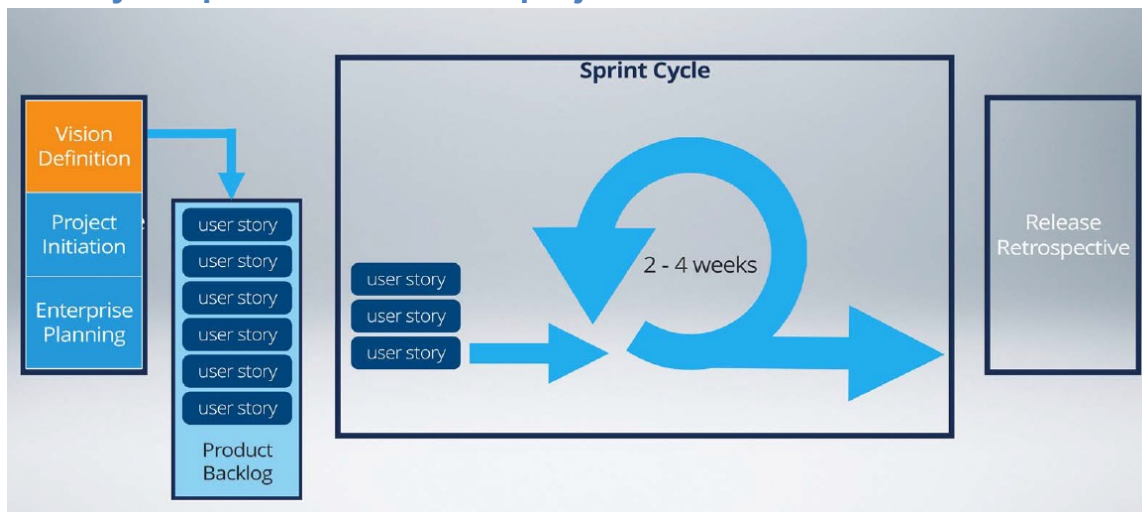
Knows when and how to use Scrum

- Uptake of system features is typically low (64% of features rarely or never used)
- **Scrum is a project management framework that is applicable to any project with aggressive deadlines, complex requirements and a degree of uniqueness.**
- The scrum team is self-organizing and self-managing, and should be completely autonomous.
- **The recommended team size is between 5 – 9 resources.**
- the team works together to produce the effective activities necessary to provide what is called the “Done” for a user story.
- A Scrum project consists of a number of time-boxed iterative periods called sprints, **which are usually 2 – 4 weeks long.**
- The sprint consists of a number of components - called user stories - that are selected from a repository of requirements - called a product backlog.
- **The team delivers a production-quality configuration at the end of each sprint.**
- After the sprint is complete, the team demonstrates the result to the product owner, and any other interested stakeholders, in a session called a Sprint Review.

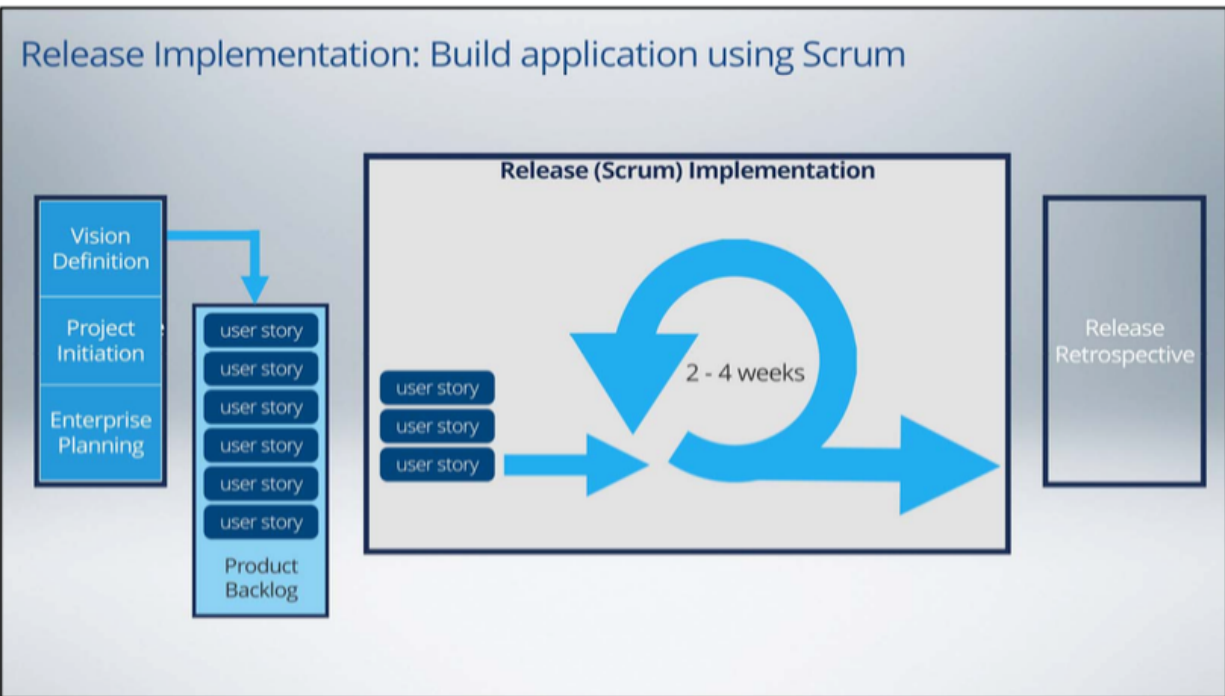




Identify the phases of a Scrum project



- We have found that the bigger picture of Scrum – the product roadmap and the high-level product backlog – and the retrospectives often fade into the background, obscuring the long-range vision and strategic goals.
- To address this, we split this initial big picture planning stage into three separate stages, each concentrating on a particular element; adding two additional stages to cover the iterative definition, build, test, and rollout processes.
- At the end of each sprint cycle - after the release retrospective - you revisit the vision and goals at the beginning of each sprint to retain strategic focus.

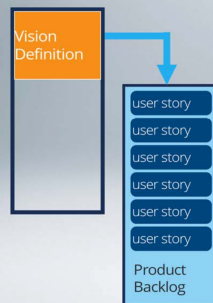


Agile Project Phase

Vision Definition

- Create project roadmap and high-level product backlog
- Long-term and short-term strategies for releasing business value in short bursts of delivery
- Clearly demonstrated business value
- Intersection of project complexity and business visibility influences your project selection
- Identify releases against each goal - or set of goals - with defined and measurable business benefits associated.
- Break down releases in projects, create Epics in associated product backlog
- Apply releases and projects to estimate delivery and create roadmap

Vision Definition: Develop understanding of big picture



Clearly demonstrated business value

Organize initiatives into releases

Prioritize releases based on ROI

Organize releases into projects

Create product backlogs

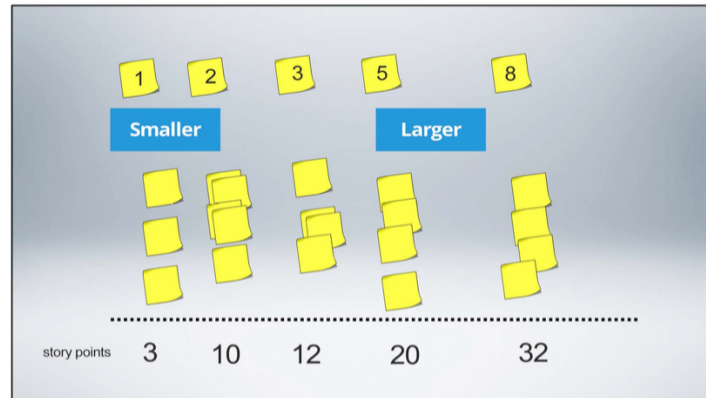
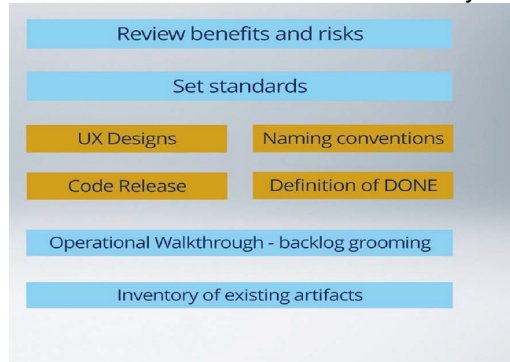
Create product roadmap

Project Initiation

- This stage focuses on setting expectations with the project team and project stakeholders, **defining the first project, and creating or enhancing**

the product backlog.

- In this stage, the Scrum teams and stakeholders review the benefits, assumptions, risks and costs associated with implementing a Scrum project.
- One of the more important activities in this stage is setting the standards that will be used to guide this project through completion.
- “Definition of Done”
- For the initial project, define the scope and kickoff/set expectations
- Create/enhance the product backlog
- Operational walkthrough- capture high-level user stories (or Epics) by understanding existing system and its challenges
- Identify what existing resources can be re-used
- Affinity sizing helps organize backlog into groups for delivery- use Fibonacci numbers to arbitrarily organize stories into sprints



The team assigns a number to each item to create arbitrary initial boundaries and, respecting the physical scale, groups the items. For example, if you are using the Fibonacci scale, place the 5 further away from 3 than 3 is from 2 on the spectrum.

After the product owner has reviewed and discussed any contentious items with the team, they register relative story points against each user story.

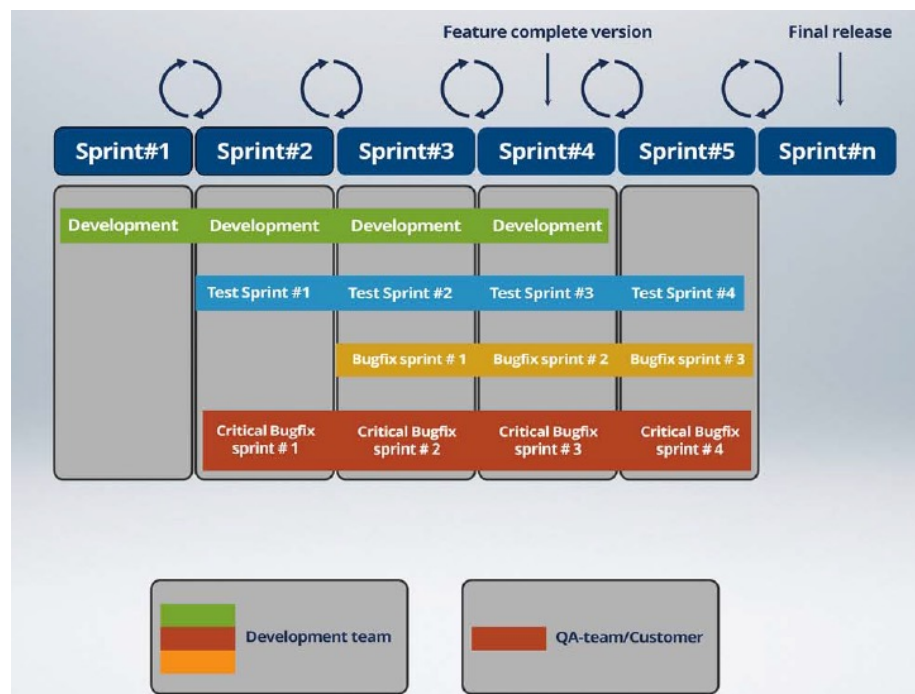
Enterprise Planning

- Plan infrastructure and reusable components for future iterations
- Design Enterprise Class Structure to maximize reuse across sprints, projects and applications
- Define/Amend Test Strategies

Release (Scrum) Implementation

- Application is built using sprint cycles using Scrum methodology:
- Backlog grooming prep
 - Product Owner validates Prioritization of backlog
 - Business value validation
 - Mitigate Dependencies
 - Confirm user story level of detail is correct

- Backlog Grooming
 - Validate user story is ready for implementation
 - User stories decomposed or aggregated
 - “Definition of Done” is settled
 - Sizing of user stories
- Sprint Planning- final product backlog preparation
 - Commit to Definition of Done
 - Final backlog preparation
 - Discuss dependencies
 - Review backlog for priorities
 - Determine Sprint duration (typically two-to-four weeks)
- Sprint Planning- Sprint backlog preparation
 - Create Sprint backlog
 - Determine user story points based on resources and velocity (how many points per sprint)
 - What/How/When work will be done (and what does “Done” mean)
- Sprint Execution
 - Team is self-regulating
 - Daily standup
 - Task updates from team members (completed and to-be done)
 - groom the Product Backlog during the Sprint, adding details and acceptance criteria.
- Sprint Review
 - Present “Done” user stories
 - Product owner decides if “done” is met
 - Accepted user stories go into the next release
 - Rejected user stories go back into the product backlog



SCRUM/Release Retrospective the team reviews their work, evaluates it and makes adjustments to improve their process

What did we do well?	What should we do differently?
What did we learn?	What still puzzles us?

DCO Tools and Best Practices

Knows how to use the principles of “Continuous DCO”

- “DCO Sessions” are the most effective way to bridge the gap between business users and technologists when building business applications.
- gather and capture the detail related to a fixed set of specifications. **This fixed set of specifications is typically called a workstream**
- Well chosen business goals and objectives
- Think big, start small (using iterations)
- Avoid automating unnecessary or ineffective processes
- Educate stakeholders in Pega methodologies (Roles, Process, Deliverables)
- Identifying Case Types is first step
 - A case type defines the tasks and decisions needed to complete a business transaction.
 - Case types are comprised of data elements, UI screens, processes and sub cases, decisions and integrations used to implement the tasks for a specific case.
 - Stages define the lifecycle of the Case
- Best practices for Stage Based Case Design
 - Represent a specific business transaction
 - Named after the business transaction
 - Use relevant names
 - Singular in context
 - Consider subcases where appropriate
 - Steps define how a case moves from one stage to the next (steps can be skipped, but must be resolved in order for case to move forward)
 - Actors and Participants:
 - Actors use applications or process work
 - Participants help Build applications
 - Collaborators help define/review requirements specs etc.

Knows how to gather and document requirements using DCO

- **an agreement between the customer and the build team on what the application will do.**
- A requirement uses business language to describe “what” we need our application to do to meet our business needs.
- Requirements can range from high-level abstract statements of services, to more detailed functional specifications.

- Requirements can also provide benchmarks against which the application can be tested.
- an inventory of events, conditions, or functions that must be satisfied and tracked in a development project.

Type	Description	Example
Business Rule	Related to the business processes in the specific application. Is usually associated with a specific use case or step in a process, and identifies the system behavior at that step.	First Name should not be longer than 20 characters
Change Control	Related to the needs of making modifications in the application and system, and managing those changes.	System shall support two-digit version numbers
Enterprise Standard	Identifies an enterprise- or industry-wide standard to which the application or system must adhere.	Routing Transit Number must be 9 digits
Functional	Identifies what the system has to do. Typically used to identify a function that will be used in the application, such as a data transformation or calculation, rather than describing system behavior.	Remaining budget must be calculated to two decimal points
Non-functional	Identifies qualities the overall system must have, typically related to how the system performs.	System needs to have 2-3 seconds screen to screen interaction

- Requirements should be categorized. At Pega, requirements are typically categorized into one of five types:
 - We use a “Business rule” category to identify those requirements usually associated with a specific use case or step in a process.
 - A “Change control” type of requirement is used to identify how we will manage changes in the application.
 - The “Enterprise standard” category is used to identify those requirements that can be applied across the enterprise, or are an “industry standard” that all applications must adhere to.
 - A “Functional” type of requirement – while similar to a business rule type – identifies a function that will be used in the application, such as calculations or data manipulation.
 - Finally, the “Non-functional” category identifies performance metrics, such as screen-to-screen interaction times.
- Requirement writing best practices:
 - Written in business terms
 - Clear and concise
 - Requirements should be consistently written
 - Verifiable- written so it can be tested by inspection analysis or demonstration
 - Requirements should be atomic

Knows how to write meaningful specifications and iterate their implementation using DCO sessions

- Specifications use business language to describe the steps needed to satisfy, or meet, a requirement.
- Specifications answer “how” are they going to do that?
- Specs can satisfy more than one requirement; one requirement may need several specs to be satisfied

- Specification Writing Best Practices:
 - Written in business terms
 - Complete and unambiguous
 - No change of ownership within the specification
 - Can be implemented and tested
 - Traceable- every specification linked to a requirement

Can provide iterative project sizing estimates

- Projects are most often sized in categories of small, medium and large.
- Depending on the size of the project, a timeline is projected – the assumption being that smaller projects take less time than larger projects
- Environmental variables such as the skills and competency of the project team can have a dramatic impact on the timeline.
- If we define the specifications needed for a given project and make it a point to accurately identify the type of specification, and are honest about the effort it will take to implement each specification, we can take advantage of Pega's Sizing wizard to help calculate a sizing estimate for the application based on its contents
- While understanding the project scope is a "must consider factor" for project planning, to develop an effective sizing estimate, it is necessary to identify variables such as the experience and competency of the team, and how much time they have dedicate to the project, and how much time it might take to implement one specification versus another. These variables help create thresholds upon which size is ultimately determined and applied.
- The initial sizing is based on the number of specifications and the complexity
- Once initial estimate complete, we factor in team availability and skillsets

➤ Pega Sizing Wizard

- The initial sizing is based on the number of specifications and the complexity value assigned to each.
- Once we have this initial estimate, we can start to factor in other environmental variables, such as team members availability to the project and their competency.

Initial estimate based on number specifications and complexi

Scrum Summary Work Effort Estimates						
Effort Category	Estimate hrs	Design Cnclg hrs	Technology Cnclg hrs	Human Factors Cnclg hrs	Testing hrs	Total
User Story Effort Hrs	280.0	-	-	-	-	280.0
Interfaces Hrs Effort Hrs	-	-	-	-	-	-
Reporting User Hrs Effort Hrs	60.0	-	-	-	-	60.0
Infrastructure Effort Hrs	-	-	-	-	-	-
Total Hrs:	340.0	0.0	0.0	0.0	0.0	340.0

Summary Work Effort								
	DURATION (WEEKS)	Baseline EFFORT hrs	Additional Effort Hrs	Testing hrs	Total hrs	Risk Contingency hrs	Additional Consulting Services hrs	Infrastructure Consulting Services hrs
RECEPTION	1.0	60.0	-	-	60.0	-	-	-
ELABORATION	-	-	-	-	-	-	-	-
CONSTRUCTION	-	-	-	-	-	-	88.0	-
TRANSITION	-	-	-	-	-	-	-	-
POST PROD SUPPORT	1.0	60.0	0.0	0.0	60.0	0.0	244.0	0.0
Total Work Hrs								

Custom scaling allows for fine-tuning estimated effort

Custom Scaling Settings				
General Project	Design/Test Factors	Use Count B	Baseline B	Baseline C
	f	125%	20%	20%
	f	100%	10%	10%
	f	60%	20%	20%
Baseline Effort	Screen Hours	Baseline Effort for Code/Case Hrs		
	Hardware Hours	4.00		
		2.00		
Work Estimates		Elaboration and Construction phases only		
	Complexity	Screen Count	Complexity Coefficient	Baseline Effort (hrs)
	New Spec/Requirement	1.0	1.0	1.0
	New User/Requirement	2.0	2.0	2.0
	New Screen/Requirement	3.0	3.0	3.0
	Existing Assets/Requirement	1.0	0.5	0.5
	Existing Existing Specifications	1.0	0.5	0.5
	Existing Existing Specifications	1.0	0.5	0.5
	New Work Type	1.0	1.0	1.0
	New Work Type	1.0	1.0	1.0
	Existing Work Type	1.0	1.0	1.0
	Existing Existing Work Type	1.0	1.0	1.0
	Existing Existing Work Type	1.0	1.0	1.0

- The Application Sizing Tool is used to estimate effort based upon Complexity (as defined in specifications) and resources (which are preset based on complexity values and then adjusted in the spreadsheet)

Attach Project Sizing to Application	Click to locate a sizing spreadsheet file on your local system and upload it to the wizard. The system updates the View Related Project Timelines values, and attaches the spreadsheet to the Attachment tab on the selected Application rule form.
---------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Types of Documents:

Application Profile	Vision and Direction
	Business Objectives
	Implementation Approach
	Duration and Timelines
	Primary Focus: Direction
Application Document	Implementation Details
	Specifications
	Application Assets
	Current State
Specification Document	Primary Focus: Progress
Specification Document	Specifications

➤ **Application Profile.**

The primary focus of the application profile is on vision and direction.

The application profile is typically **generated near the beginning of the application development** effort, as a summary of the proposed solution, and includes business objectives, the planned implementation approach – be it Scrum, Pega BPM, or something else - and expected duration and timelines.

Project managers and business analysts use the application profile as a solution proposal to submit to project ownership for review and approval. The application profile provides the team with direction – the “what” it is that we want to do.

➤ **Application Document.**

This document provides much of the same information provided in the application profile.

However, the primary **intent of the application document is to provide information on “how” the application is implemented.**

The application document is typically produced **iteratively** as specifications are entered and elaborated on.

As the application build effort progresses, detailed information about application data and other implementation assets becomes readily available.

The application document provides the team with configuration details – “how” we implemented the “what” we wanted to do.

➤ **Specification Document.**

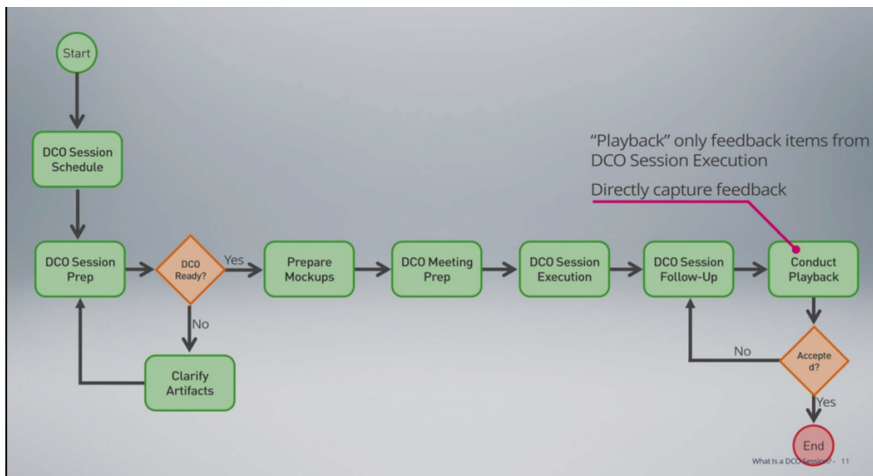
The primary focus of this document is....well....specifications.

Business analysts and project managers define specification sets and can then publish a specification document for each set.

For the CERTIFICATION: Need to know what happens when the Sizing document is part of the application document...I could not get this to work in our Dev environment

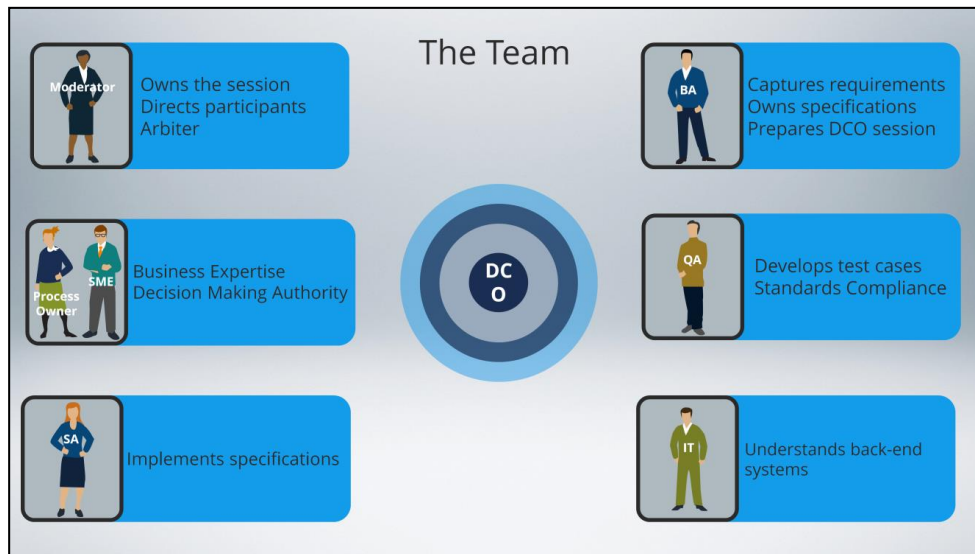
Conducting DCO Sessions

Knows the best practices for conducting a DCO session

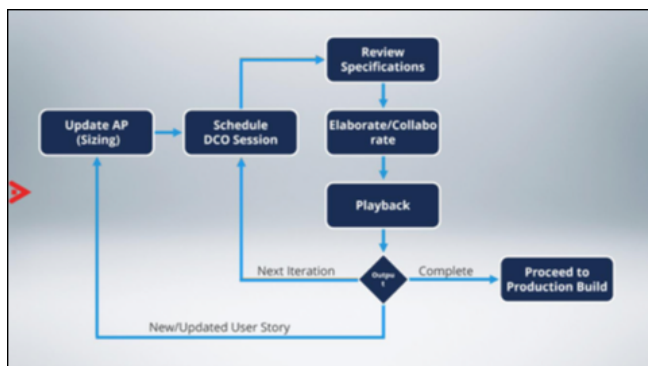


- During DCO Prep, work to make artifacts “DCO Ready”
- Once “DCO ready”, then draft versions of the flows and UI screens are mocked up and made ready to present to the session team
- DCO Session :
 - The process owner walk through the business flow using requirements, specifications, draft flows, and UI components as the canvas for discussion.
 - The feedback from the team should be directly captured back in the system as updates to the requirements, specifications, the draft flows and UI notes.
 - Normally updates to flows and UI are not done in real-time unless there is no impact to the meeting.
- During the “playback” the scope of artifacts to be reviewed is reduced to only the feedback items that were captured in the previous DCO session.

Conduct DCO:



- **moderator** - responsible for managing the DCO session and all participants included in the session. Their main focus is to drive through the specification set and to ensure that any takeaways are captured and that the session stays on track; ensuring that the proper client SME's attend the session and that all physical resources required for the DCO session have been reserved. They also play an arbitrator role and control scope and level of discussions.
- **Process owner** or other **subject matter expert** - intimate knowledge of the business process being designed, and should have the authority to make decisions regarding the business process during the session.
- A **system architect** is responsible for making sure that everything they need to build the solution is captured. This includes helping build the case map so this resource must have expert level experience with case design and user interfaces.
- **business architect** is responsible for capturing any changes to the requirements or specifications discussed in the session. They also participate in the DCO session preparation activities.
- **QA representative** is responsible for developing the test plans to test the features and functionality described by the requirements and specifications. They are also responsible for addressing the quality of the requirements being presented to ensure they meet corporate standards and are being addressed at an appropriate level to ensure success.
- **Representative from IT** has knowledge of any back-end systems that might be affected by the specifications. This resource must have the correct level of experience with the existing systems and any new systems that are being put in place and have experience with the technologies used.



Every D - C - O session, regardless of the type, follows the same basic pattern. There is always some level of review – so we know what it is we are up against.

Then, depending on the type of D - C - O session there will be some amount of work to “flesh out” the specifications into working prototypes – drafts.

The last step is the playback. Again, this “playback” provides the business line with an opportunity to preview the business process before it is fully developed as a production-ready asset.

An effective D - C - O session should always produce some result. We evaluate this result to determine our next steps.

- **Prepare for DCO**
 - Engage stakeholders early and keep them engaged
 - Educate Team Members, Set Expectations, Prepare schedule in advance, send invites early, ID Process Owners, ID Session Roles
 - Organize Level 1 Specs into Workstreams
 - DCO sessions are not vision sessions, and that the to- be-built process and UI should be identified and scoped before the session begins
 - Prepare upfront DCO session schedule and make sure that **each role is represented and can be present at each session.**
 - Make sure that **each business process is assigned a Process Owner.** This includes identifying subject matter experts and ensuring they are the “decision maker.”

- **DCO Sessions**
 - tightly moderated meetings with a **fixed and clearly defined set of resources** whose sole purpose is the gathering and capturing of detail related to a **fixed set of Use Cases**
 - At the meeting, ensure attended by required resources and ensure that the SME is the decision maker.
 - Parking Lot Items:
 - Keep the list of open, **unresolved issues to 5-10 items.** If more than 10 we may be missing an SME.
 - Never add items to the parking lot that are not related to the DCO session.
 - Each item assigned a resource - and delivery date prior to end of DCO session
 - Whenever possible, update the specifications and requirements during the session, rather than afterward, to avoid confusion or inaccuracy.
 - The goal of any DCO session is the “next iteration”

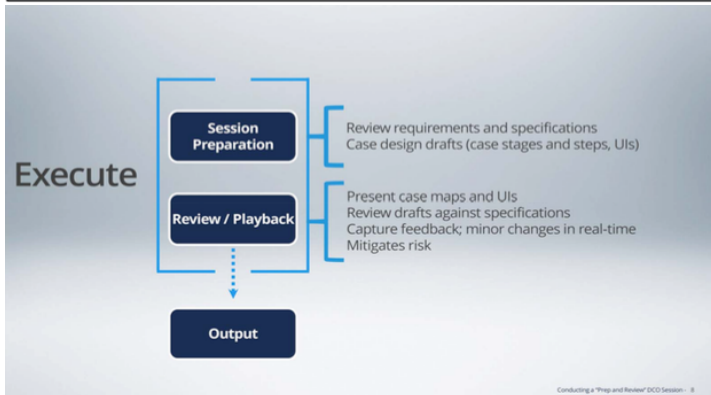
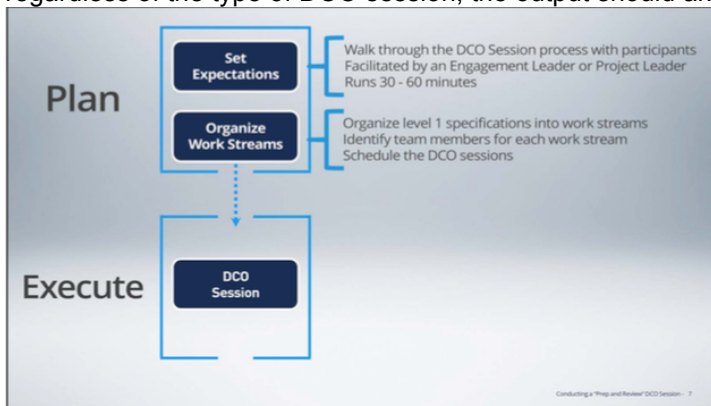
- **After DCO**
 - Begin implementation Work as soon as possible
 - Follow up on parking lot items and ensure deadlines are met
 - Focus on delivering next iteration

Knows how to Conduct a “Prep and Review” DCO session

“Prep and Review” DCO session,

- The first step is to prepare all the artifacts (requirements, specifications, mockups) for the DCO session.
- The team reviews the requirements and specifications and begins designing the case maps.

- Once this work is completed, the DCO session itself is held.
- During the DCO session the case maps are presented to the full team for review.
- The process owner walks through the relevant steps in the case providing full details on exactly what each step is doing and how each step directly maps to a specification.
- If any changes are required, they are either made in real-time if there is little to no impact to the meeting, or notes are taken in the specifications themselves that document the changes that are needed.
- At the end of the session the goal will be to have a fully reviewed and accepted case design draft that all parties agree meet the requirements of the business process that was targeted
- regardless of the type of DCO session, the output should always be the same



Other Types of DCO Sessions (Whiteboard and Real-Time Capture)

- Work streams are defined in advance; DCO sessions scheduled
- During sessions, Case Types, Reqs, Specs, Mockups are identified, designed collaboratively in realtime

Can identify the key deliverables of a DCO session

- DCO sessions cycle through Review, Build, Playback until the workstream is complete
- There is always some level of review – so we know what it is we are up against.
- Then, depending on the type of D - C - O session there will be some amount of work to “flesh out” the specifications into working prototypes – drafts.
- The last step is the playback. Again, this “playback” provides the business line with an opportunity to preview the business process before it is fully developed as a production-ready asset.
- An effective D - C - O session should always produce some result. We evaluate this result to determine our next steps.
- For example, during the first few D - C - O sessions, the focus is on fleshing out the details of the application, so **each D - C - O session will most likely produce the “next iteration” as its**

output, and then we would prepare for, and hold, the next D - C - O session.

- We would repeat this process for as many sessions as it takes to complete the work stream.
- When the business line is OK with the case design, we can proceed to a production build effort.
- However, **if during the playbacks it is determined that a user story needs to be updated – or new ones need to be added – we would update the application profile - and most likely reset sizing estimates as well.**

Project Governance and Oversight

Knows the importance of governance

- A properly functioning governance program focuses
 - first on doing the right projects and programs constrained by the organizations capacity to undertake the work;
 - Secondly, that the processes and methodologies used to implement them are effective and repeatable; and that those performing those activities are accountable for their actions.
- The balance of project governance focuses around creating the environment that generates the capability to deliver projects and programs effectively.
- Governance provides for effective sponsorship, effective staff development, effective and flexible processes and procedures, simple but accurate reporting and good early warning systems to identify issues, problems and projects no longer creating value.
- Governance over Pegasystems programs and projects ensures conformance to your policies, procedures, and processes to deliver great systems.
- The key activity in Governance involves conducting effective regular governance meetings.

Can identify the types of governance programs



- **Executive Governance:** Executive management establishes this governance program to focus on the activities of stakeholders to achieve corporate goals. The intent is to impact projects from the top down and to provide a strong level of interaction and support between various organizational divisions.
- **Program Governance** focuses on the relationship between business and IT. It has a profound influence on the way solutions are implemented - and their ultimate success.
- It is important to note that well-defined program governance can pave the way for successful projects. Poorly designed, or non-existent, program governance often leads to barriers or silos between business and IT, and a high failure rate.
- Program governance is usually divided into two levels.
 - **Strategic level program governance** focuses on defining the roadmaps and timelines, and providing the business segment or operating group direction to implement transformational initiatives.
 - **Tactical level program governance** focuses on enabling delivery teams and business

- sponsors to implement these initiatives through various delivery groups and methods.
- Then, there is project governance.
 - Project governance defines the processes and methodology that need to be in place for successful solution delivery.
 - Project governance focuses on establishing solid lines of communication to ensure that the business benefit is delivered and project scope, time, and budget are controlled.
 - A well-defined project governance program is essential to the proper functioning of a Pega development team – and absolutely critical to the success of a Pega project.
- A solid governance model ensures that issues are resolved in a timely manner consistent with the fast-paced, iterative nature of the methodology.

Knows the best practices and principles for effective governance

- Executive governance sponsorship establishes and funds the initiatives needed to implement corporate governance.
- Program governance sponsorship is typically business and IT executive management. Sponsors ensure that there is a driving need for governance activities. Because sponsors influence and remove roadblocks, they must remain connected with an open flow of communication.
- We also recommend assigning a project sponsor to act as a project champion and push initiatives through to success.
- Project sponsors often participate as part of a project board to monitor and control the outcome. The most successful champions are those that have a particular interest in the project outcome. The executive team may appoint a project sponsor, but more often project sponsors have a strong relationship with the executive team and appoint themselves to sponsor a project. Many project sponsors are senior members of the executive committee and are the chair of the project board.
- Governance Without Communication Is Not Governance
- The key activity in Governance involves conducting effective regular governance meetings.
- Effective communications and active governance are critical components of successful delivery. Governance meetings that “get things done” are a primary predictor of ultimate success.
- Governance Meetings:
 - Governance meetings are important for keeping a continual flow of information between project leadership, business sponsors, and executive leadership.
 - These meetings – which should last an hour when held every two weeks – provide bidirectional escalation paths to quickly identify and resolve issues, keeping all of the parties in sync with the project's charter and minimizing risk to the project.
 - The meetings should include the project sponsor, project manager, business leadership, QA, and IT, who review the current status of the project.
 - Don't present project demos in governance meetings. Record and publish the minutes for each meeting to provide an audit trail.

Good governance includes:



Project Steering Committee

Understand strategic, long-term impact
Own the implementation



Weekly Status Reports

Reconcile outstanding milestones
Document decisions and approved changes
Communication tool to help reduce risk



Issue Resolution

Evaluate and resolve issues



Governance Meetings:

1 hour: every two weeks
< 1 hour: weekly
> 1 hour: critical milestones approaching

Meeting (Owner)	Frequency and Format	Participants
Project Kickoff (PM and TEL)	Roundtable and conference call to ensure all pre-project items are complete and review project plan	All project team members
Regular Pulse (PM)	Daily Meeting	PM, LSA, TEL
Status Update (PM)	Weekly conference call to review status and issues, and keep project on track	VP IT, PM, LSA
Stakeholder Governance (Business Sponsor)	Bi-weekly, additional by exception	Business Sponsor, VP IT, PM, LSA

- ✓ Not optional
- ✓ Established and communicated early
- ✓ All parties agree on governance specifics
- ✓ Include milestones for each meeting
- ✓ Schedule regular, recurring meetings
- ✓ Review project status reports



Knows the best practices for effective change control


Change Control

Anticipate Change to Manage It Effectively

PLAN FOR CHANGE

DESIGN FOR CHANGE


BUILD FOR CHANGE



Change is **inevitable** so plan for it early; managed change is a **good** thing


Ensure that application design supports capabilities of both **current releases** as well as **future releases**

Implement an approach that **adapts and embraces** change but **closely controls** it



Change Control

Balancing Schedule, Resources, and Scope



We can only control **two** variables at any time
Business and developers need to **coordinate**
Quantify impact along **all three** variables

An effective strategy:

- Reduces surprises and builds trust
- Quantifies the impact of change

Without a strategy, we may:

- Need to drop features
- Slip dates or go over-budget
- Misalign with business expectations

Can implement an effective change control process

Building a Change Control Process



Plan Create a formal change-control process **early**
Establish acceptance by **all parties**

Capture Changes Clearly identify the change proposal with **full detail**
Use project management software to **manage and track the change**

Classify/Prioritize Establish whether the change is **new** or a **clarification of an existing requirement**
Prioritize the change against the existing requirements

Assess Impact Assess the impact to the **business value**
Assess the impact to the **project** (effort, timeline, budget)

Decide Review the change in the Change Control committee

- If **approved**, integrate into plan
- If **rejected**, record the decision

- These best practices help to set expectations early on for the entire team – especially the business – by clearly outlining the details of the change control process.
 - Clearly define when to use the process;
 - How to track proposed changes;
 - How the change approval process works;
 - Who participates in the change control process;
 - And, who the final arbiters of change are.
- The project manager should establish the process early during the project, and ensure that the change-control process has been communicated to the entire team.
- All of the parties involved in the project should fully understand and agree with the process.

- All change proposals should include the details necessary to clearly understand the problem and the proposed resolution.
- Ideally, these details should be captured in the project management system in use on the project.
- We should have enough information to classify the change as either a new requirement or a clarification of an existing one.
- This information allows us to prioritize the change against the other project requirements.
- Once we've prioritized the change, we can assess its impact on the delivered business value, and the implementation of the project.
- This assessment should include the anticipated effects on project staffing, timelines, and cost.
- The change, and its impact, should be reviewed by the approval committee.
- If approved, the change should be incorporated into the project plan. If it's rejected, record the reasons for the rejection. This information may affect the decision to re-assess the change at a later date.