



# Pega Certified Business Architect (PCBA) 8.7 Exam Prep

EXAM  
PREP

# Agenda

- Study Materials
- Exam Advice
- PCBA Exam Test Domains and Distribution
- Review Each Test Domain
- Sample Exam Questions
- Answer Your Questions

# Pega Certified Business Architect Exam

Certification Path:



## Pega Certified Business Architect Exam

- Consists of 50 multiple choice exam questions, 47 are graded
- 90 minutes to Complete Exam and NDA (nondisclosure agreement)
  - 120 minutes will be granted to students taking the exam from non-English speaking countries
- Click the link below to review the NDA in advance to save time during the exam:  
<https://community.pega.com/knowledgebase/documents/certificationprogramtermsofparticipation20160331>
- Passing Grade – 70 % (33 questions answered correctly)

# Pega Certified Business Architect Exam

## Test question formats:

- **Multiple Choice** — Select one option that best answers the question or completes a statement.
- **Multiple Selection/Response** — Select more than one option that best answers the question or completes a statement. The text states how many options are correct, such as Choose two.
- **Matching (Drag and Drop)** — Select an item in column 1 and associate it with the correct response in column 2.
- **Build List** — Select correct order from random shuffle of answers

# Pega Certified Business Architect Exam

- The following Exam Code is used for registration:
  - **Exam Code:** PEGAPCBA87V1
- The exam is currently in the following languages:
  - English, French, German, Japanese

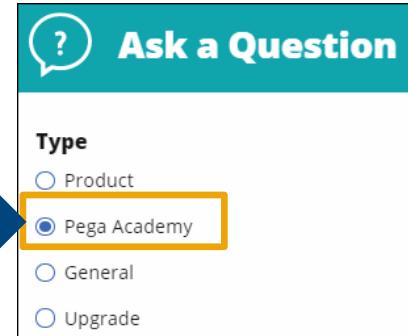
## Pega Certified Business Architect Exam

- Results made available immediately upon completion of exam
- May re-take the exam up to 3 times in a 12-month period
  - Must wait 3 calendar days from the first fail before attempting again
  - If taking the third time must wait 2 weeks
- Sign up for the exam at <http://vue.com/pegasystems>

## Study Materials

- Study all course materials and complete all the exercises
  - If you are having difficulties with the self-study exercises, please use the Pega Academy Community address these issues:
    - <https://collaborate.pega.com/node/add/question>

Be sure to check the Pega Academy radio button



- Review the PCBA exam topics and their distribution % here:
  - <https://academy.pega.com/exam/pega-certified-business-architect-3>

## Exam Advice

- Target your review using the slides in this deck.
- You can also use the Business Architect Essentials Student Guide –  
<https://v953w.app.goo.gl/pAUP>
- Book your test appointment on Pearson VUE well in advance. Seats fill quickly and availability varies by location.
- To reserve a seat at a local testing center near you, click this link:  
<http://pearsonvue.com/pegasystems>
- To schedule for and take the test from home, click this link:  
<https://home.pearsonvue.com/pegasystems/onvue>

## Exam Advice

- When taking the exam, consider the following:
  - Go for the “easy” questions in your first pass. Answer obvious questions first.
  - If you are unsure, **Mark for Review** and move on!
  - Use **Strikeout** (Right-Click) to exclude distractors (answers you believe to be wrong).
- Positive answers tend to be right. If it sounds like a feature or concept that is sensible or makes business sense, chances are it is correct.
- Apply the principal of **Occam's Razor**:
  - In layperson’s terms, “The simplest solution tends to be the best one.”
  - When presented with competing hypotheses to solve a problem, one should select the solution with the fewest assumptions.

# Answering Questions

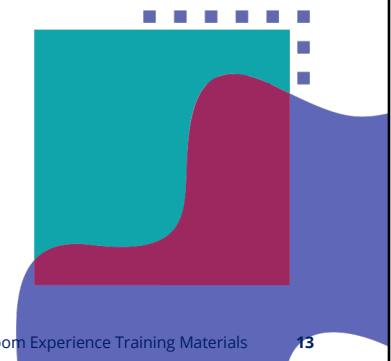
Most questions are presented as multiple choice based, with either a single correct response or multiple correct responses.

- The title bar displays the name of the exam, your name, time remaining and an indication of your progress.
- You can flag any question for later review using the Flag for Review option.
- To answer a question, select the appropriate number of correct responses.
- Click **Previous** to move back through the exam questions.  
Click **Next** to move forward through the exam questions.

The screenshot shows a digital exam interface. At the top, there's a dark blue header with the text "Pegasystems - Candidate Name A" and "Time Remaining: mm:ss". To the right of the name is a red circle containing the letter "A". Below the header, the page number "12 of 73" is displayed. On the right side of the header, there's a "Flag for Review" button with a red circle containing the letter "B" and a small icon. The main content area has a light blue background. A question is displayed with the letter "C" in a red circle above it. The question text is: "The rain in Spain falls mainly on the \_\_\_\_\_.  
A. Roof tops  
B. Plain  
C. Patio tables  
D. Streets". At the bottom of the screen, there's a dark blue footer bar with the letter "D" in a red circle, followed by "← Previous", a "Navigator" icon, and "Next →".

## Reviewing Questions

- Before you complete the exam, you are provided the option to review the exam. Items marked for review display a colored flag.
- From the Item Review screen, you can:
  - review the full exam
  - review only those questions not answered
  - review only those questions flagged for review



## Submitting the Exam

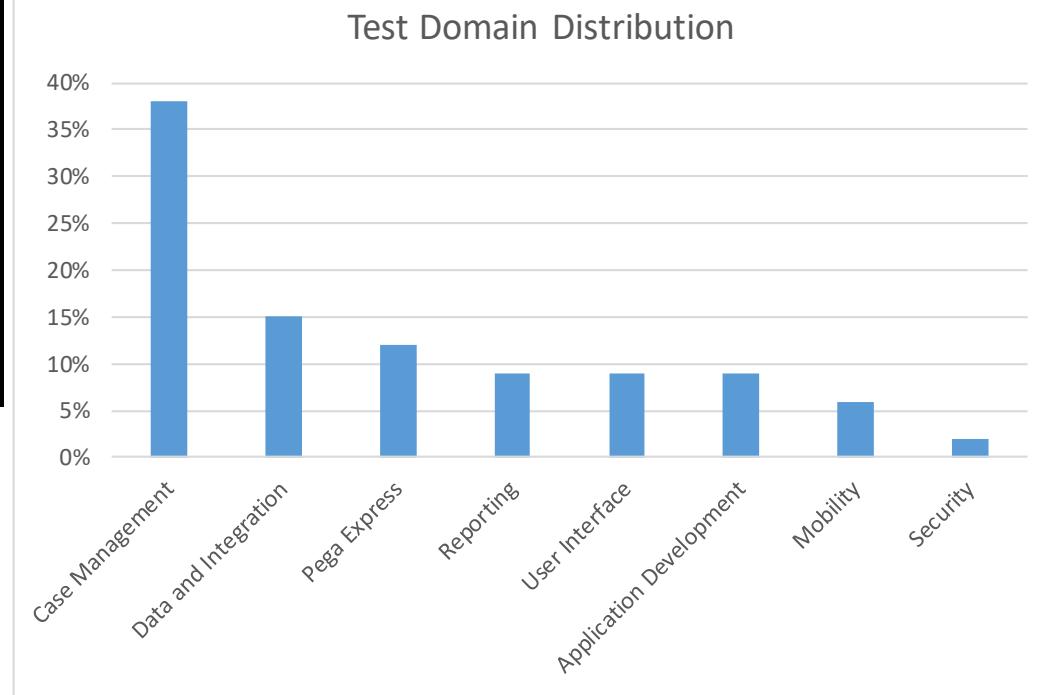
- Once you submit your exam for scoring, it cannot be recalled.
- If sitting for the exam on-site, the proctor will give you a printed Exam Summary Report. This report indicates whether you passed or failed, and the percentages of correct answers in each test domain.
- If sitting for the exam online, you be able to see your scores immediately upon completion and will receive an email from Pearson Vue with your scores.
- The exam report will not list individual questions or indicate which questions you scored correctly or incorrectly.

# PCBA 8.7 Test Domains

Requires 70% to pass

Test Domains	% of Exam	# Questions
Case Management	38%	19
Data and Integration	15%	8
Pega Express	12%	6
Reporting	9%	5
User Interface	9%	5
Application Development	9%	5
Mobility	6%	3
Security	2%	1
<b>Total</b>	<b>100%</b>	<b>50</b>

Exam Blueprint here:  
<https://academy.pega.com/exam/pega-certified-business-architect-3>



# Exam pointers

## BEFORE THE EXAM

- Challenge how you found Exam Readiness.
- Comfortably answering questions within 75 secs?  
... and getting more than 80% correct?
- Typically taking the full 75 secs?
- Getting 2 out of every 5 wrong?
- Study topics in proportion to their weighting.
- Familiarize yourself with the exam taking system provided by Pearson VUE.
- **Pearson VUE demo test** under **Related links** at:  
<https://home.pearsonvue.com/pegasystems>

## DURING THE EXAM

- Get to the end!
  - Put the “easy” marks in the bank
  - Use “Flag for Review” for anything you need to come back to
- If you think there are more correct options than there are options to pick, choose the most correct options.
- If you have seen behavior in Pega that differs from the course material, answer questions based on what the course materials say.

## Practice Exam Links

**PCBA 8.7 Randomly generated 50 question practice exam:**

<https://www.classmarker.com/online-test/start/?quiz=n7x6229096759c47>

- 90 minutes
- Answers visible at the end
- Not saved

**Self Study Mission Test:** <https://academy.pega.com/mission/business-architect/v4/test/in/35761>

# Case Management

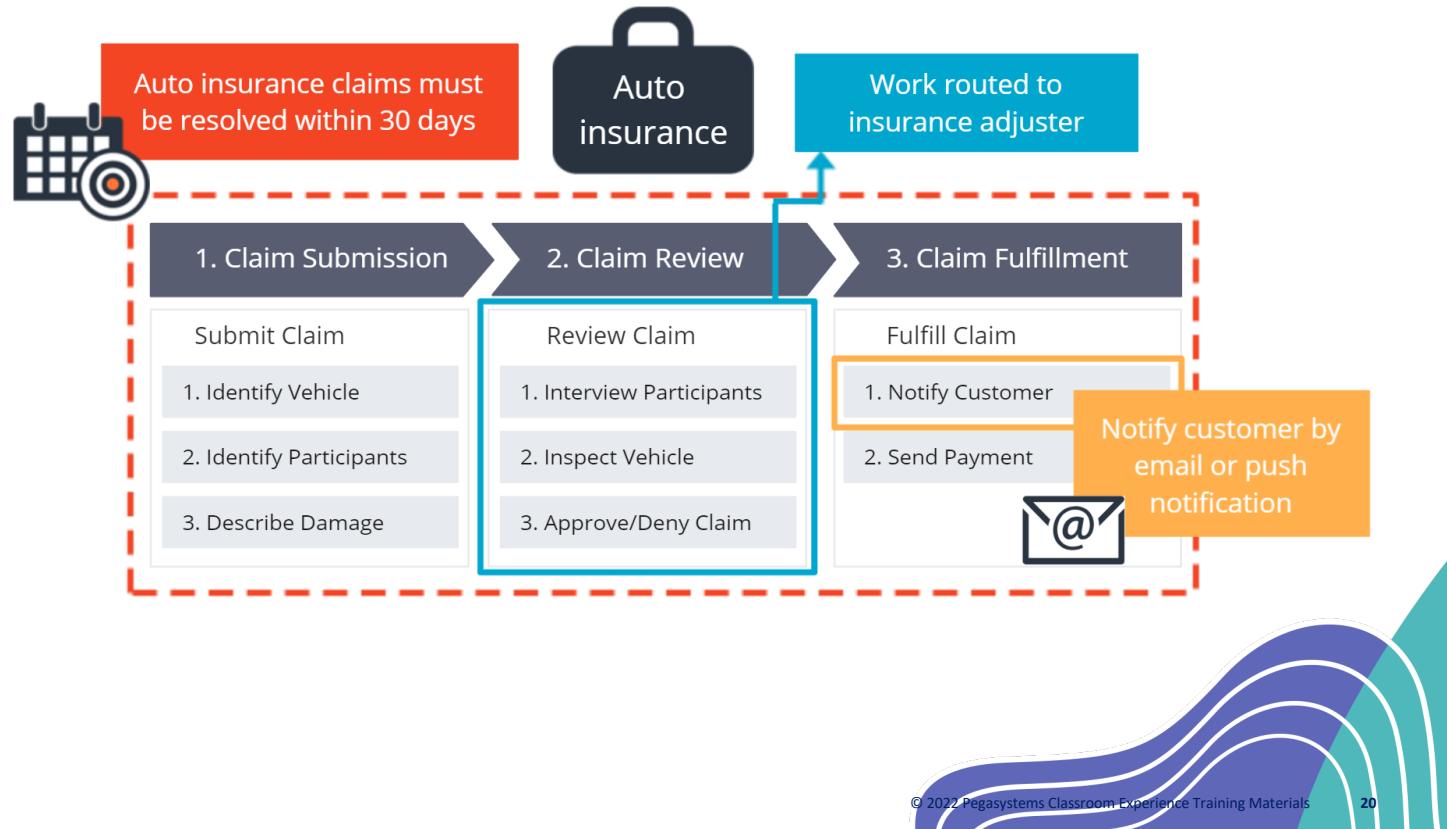
38%

## Case Management 38%

- Design a case life cycle: stages, case statuses, add instructions to assignments
- Add a service level agreement: urgency, goals, deadlines
- Route assignments to users, work queues
- Design an approval process
- Configure and send email correspondence
- Identify duplicate cases
- Add optional actions to a workflow
- Understand when to use automation shapes
- Skip a stage or process
- Create a child case
- Automate workflow decisions using conditions
- Pause and resume case processing: wait steps
- Calculating fields using decision tables

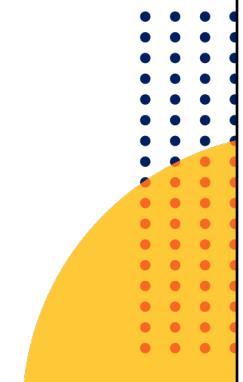
# Case Management - Model-Driven Design

- Case types
- Cases
- Service Levels



## Case Management – Case Types and Cases

- **Case type** – Definition of a Specific Business Transaction; a “template” which defines a specific type of case and process(s) tasks
- **Case** – An Instance of a Case Type, a Specific Business Transaction
- All cases generally have the following characteristics:
  - a unique identifier
  - status
  - urgency
  - follow at least one process
- Cases are represented as a composite, an object, and they allow us to associate several different elements.
  - Actors, Tasks, Data, Status, History and ultimately Resolution



# Case Management – Case Types and Cases

- A case is created for each dental claim filed by a patient.
  - **Case 1** — Teeth cleaning for John Smith on May 3
  - **Case 2** — New crown for Linda Wise on January 15
- Each case moves through the processes as defined in the case type.

**DENTAL CLAIM**

Patient Information
Name
Date of Birth
Gender
Email
Procedure Information
Procedure
Claim Date

**DC - 1**

Patient Information	
Name	John Smith
Date of Birth	March 31, 1982
Gender	Male
Email	Smith@Pega.com
Procedure Information	
Procedure	Teeth Cleaning
Claim Date	May 06, 2015

**DC - 2**

Patient Information	
Name	Linda Wise
Date of Birth	June 19, 1980
Gender	Female
Email	Wise@Pega.com
Procedure Information	
Procedure	Crown Applied
Claim Date	January 15, 2015

# Case Management- Case Type Relationships

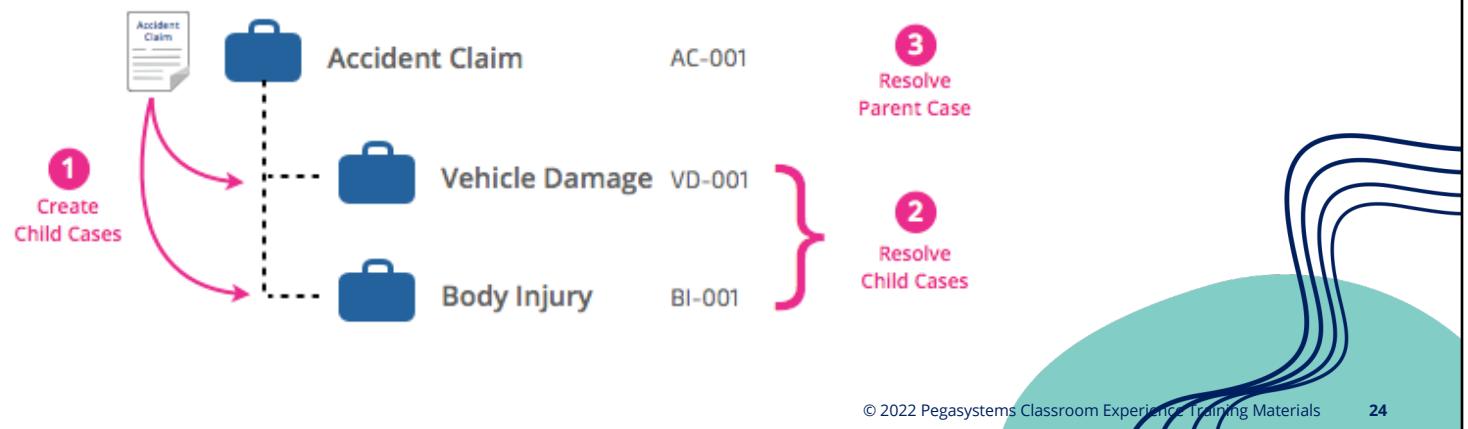
- A business transaction can be complicated, and its processing involves multiple cases that must be worked in parallel.
- You can model this parent-child relationship with a case type hierarchy that contains a top-level case type and child case type.
  - A **top-level** case type does not have any parent case type, but can cover, or become a parent of, other case types.
  - A **child** case type is covered by a parent case type. Child case types represent work that must be completed to resolve the parent.



# Case Management- Case Type Relationships

Business process: Car Accident Claim.

- Top-level case type: Accident Claim.
- Two child case types: Vehicle Damage and Bodily Injury.
- Each child case is addressed by a different parties.
- Both child cases must be addressed before the Accident Claim can close.



## Case Management- Case Type and Stages

- We define the Case Type using the Case Designer, a landing page used to view and modify the Stages, Processes, Steps, and behaviors of a case.
- A Case Type is divided into the Stages which represents the overall workflow, each Stage represents a distinct phase of the Case Type's life cycle.
- A Stage is a first-level grouping for related tasks.
- What constitutes a Stage?
  - Change in case ownership, change in case status or step belong together

# Case Management- Stages

There are three different Stage types:

- **Primary** stages – automatically or manually transition to another stage
  - Automatic – represented by the chevron on the stage shape
  - Manual – represented by a rectangle shape
- **Alternate** stages – manual transition to another stage
- **Resolution** stages – manual transition to another stage
  - Denoted by a red line under the stage name – does not automatically update status

## Case Management- Stages

Stage best practices:

- Start with stages – not the details
- Separate happy path and exceptions
- Try to maintain limit of 2 nouns per stage name
- Use resolution stages for clarity and understanding

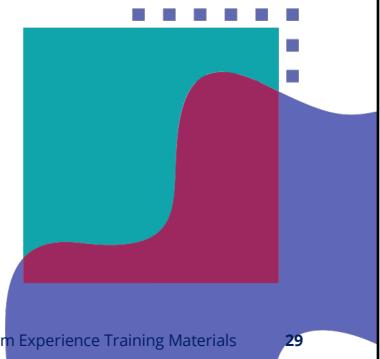
## Case Management- Stages, Processes, and Steps

- Each Stage can be broken down into one or more Processes
- Each Process can be broken down into one or more Steps
  - Each step represents a distinct action like an assignment, or the creation of a subcase.
- By default, each Process occurs in sequence however we can modify
  - Make the Process available to the user when our case enters the stage
  - Make the Process conditionally available to the end user

# Case Management- Processes and Steps

Process and Steps Best Practices:

- Use a verb or action phrase to describe each process
- Use an iterative approach
- Sets the expected order of tasks
- Universally understood
- Easily communicated

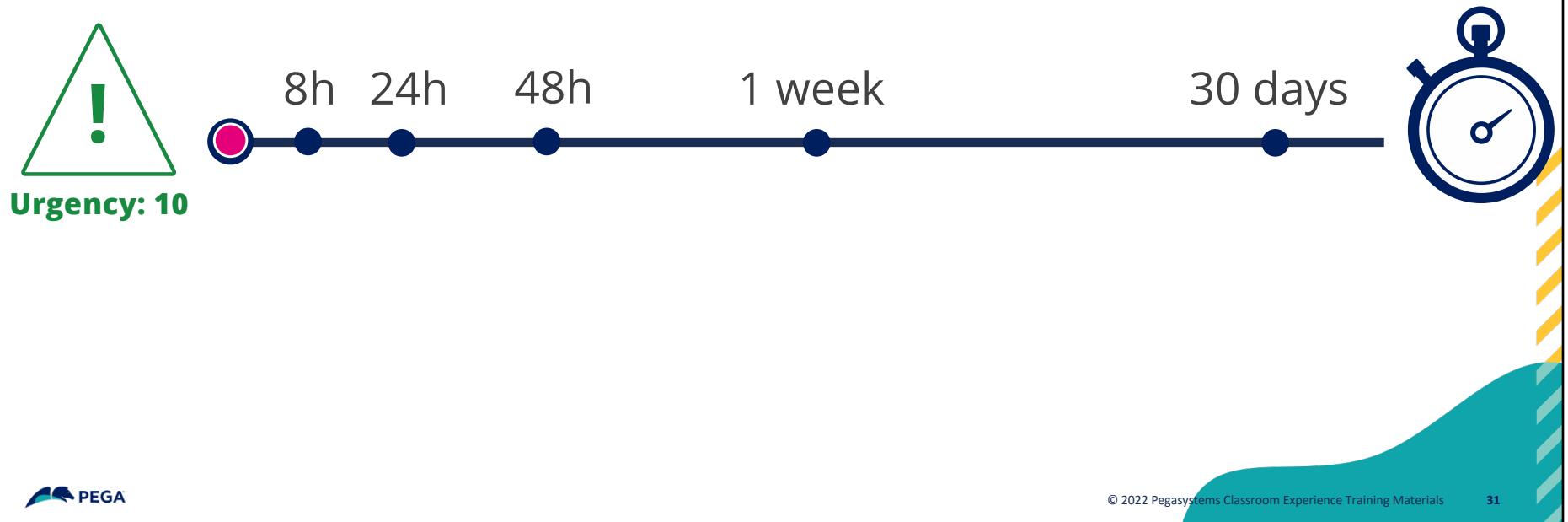


## Case Management- Service Levels

- Service Level Agreements or SLAs are used to ensure work is completed within a certain time frame.
- We define service levels using 3 milestones:
  - Goal – how long it **should** take – measured from assignment start
  - Deadline – how long it **may** take – measured from assignment start
  - Passed Deadline – how long after the deadline before taking additional action – measured from the end of the deadline – repeatable

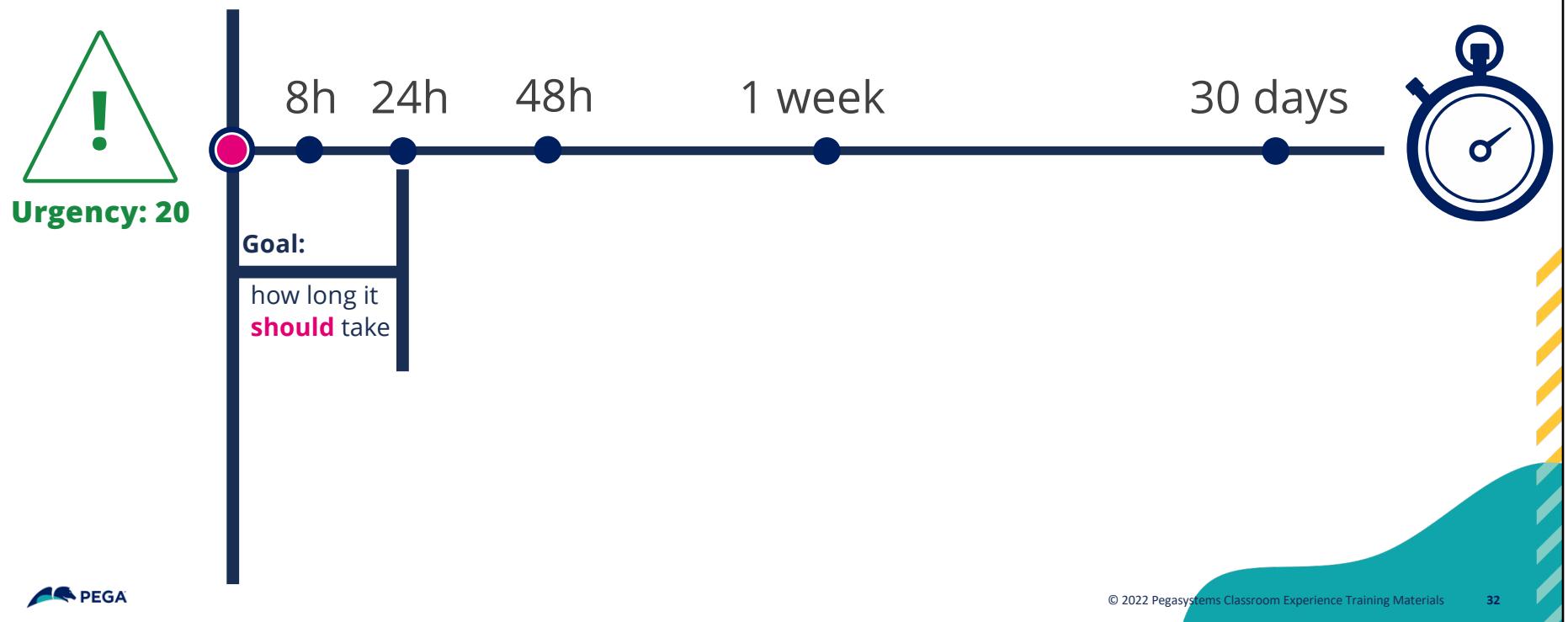
# Case Management– Service Levels

## Service Level Agreements



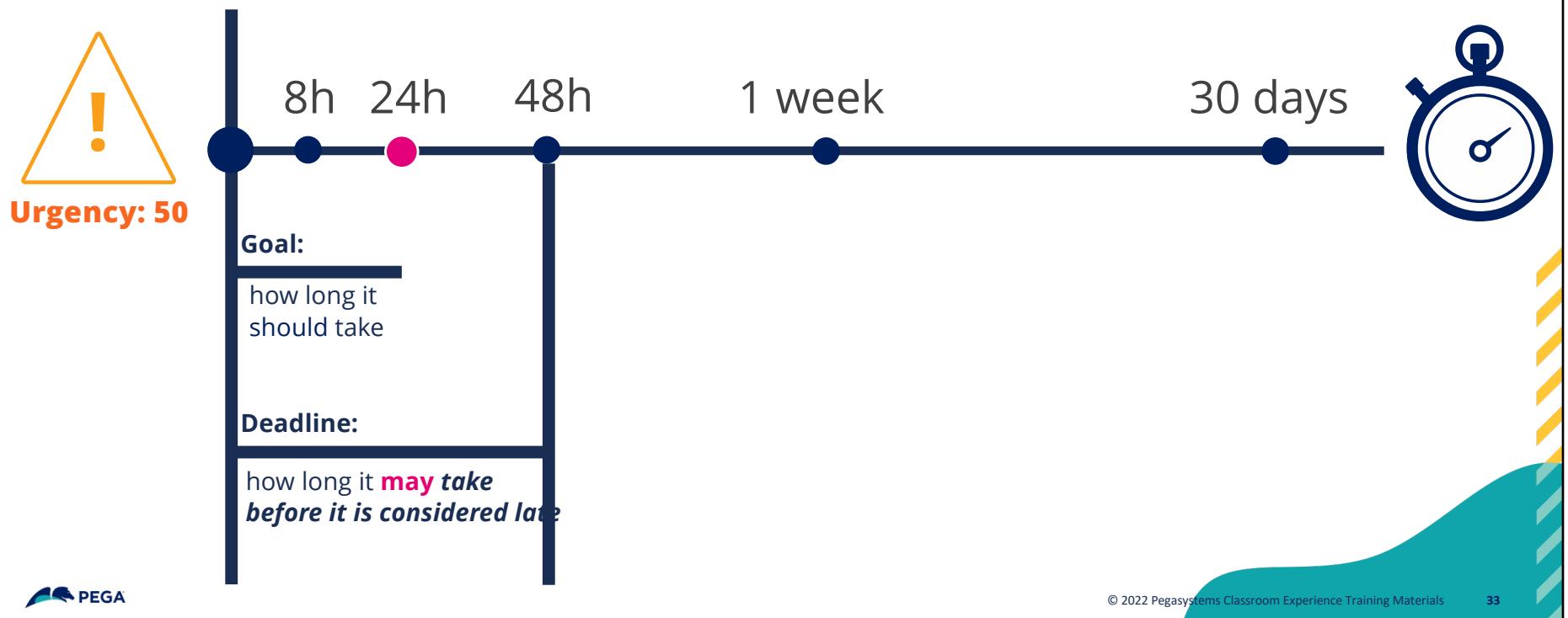
# Case Management – Service Levels

## Service Level Agreements



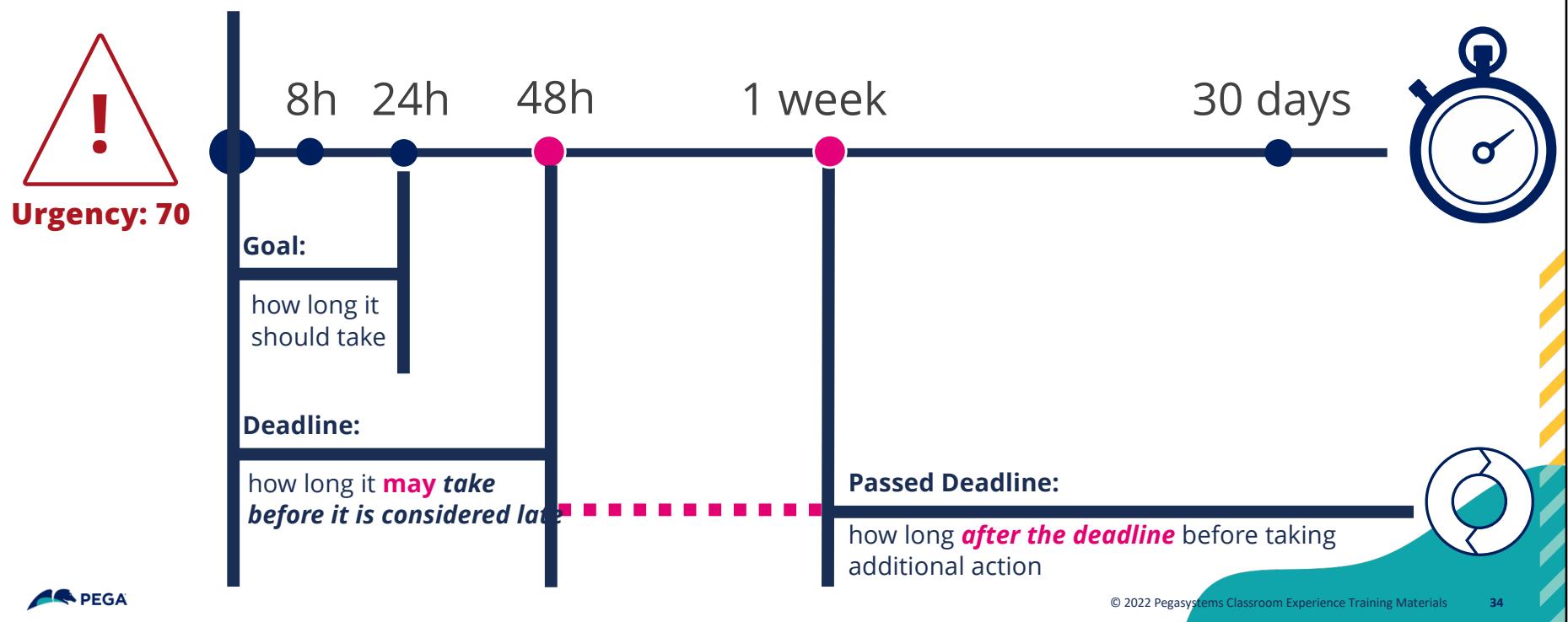
# Case Management – Service Levels

## Service Level Agreements



# Case Management – Service Levels

## Service Level Agreements

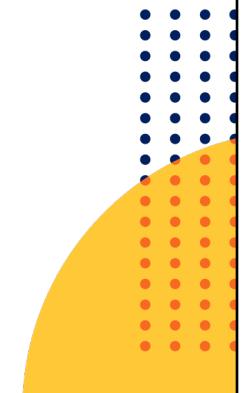


## Case Management – Service Levels

- Service levels can be set for both the **Case Type** and **Assignment** Shapes as well as **Stages** and **Processes**.
- To set on a **Case Type** use the Settings tab on the Case Designer, modify the setting by editing the Goals and Deadlines.
- To set on an **Assignment** Shape, open the assignment properties and reference a Service Level rule.
- Typically, we increase the Urgency at each milestone.
  - Urgency has a fixed range: 0-100
- **Escalation Actions** can be configured at each milestone.

## Case Management – Controlling the flow

- A Case's Work Status can be changed as the Case progresses through its life cycle from Creation through Resolution.
- The status of a case can be set on stage entry.
- Flow shapes in a Flow Rule can be configured to change the Status of a Case.
- All Cases begin with a “**New**” Work Status and are closed when they reach a “**Resolved**” Work Status.



## Case Management – Controlling the flow

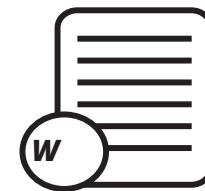
All internal Work Statuses in Pega should begin with one of the following keywords:

- **New**
- **Open**
- **Pending**
  - ex. Pending-Approval
- **Resolved**
  - ex. Resolved-Completed
  - ex. Resolved-Rejected

# Case Management – Routing

Route cases to a Worklist or a Work Queue.

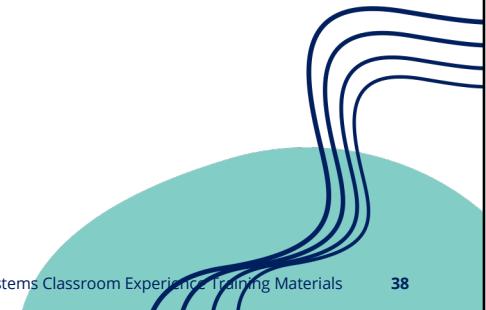
- **Worklist** – a list of all the assignments for a particular operator
- **Work Queue** – routing queue where assignments are stored; multiple operators can access them



Worklist

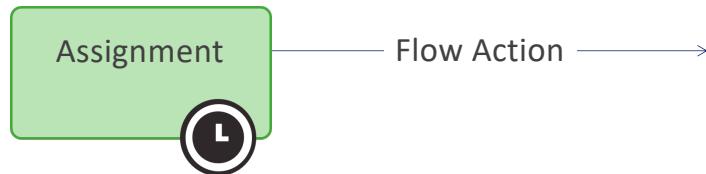


Work Queue



## Case Management – Routing

- Routing is configured in assignment-based steps in the case designer and in the Assignment shape in a flow rule:
  - Determines who works on the case (work list or work queue)
  - The flow action connector(s) determine what they see on the screen

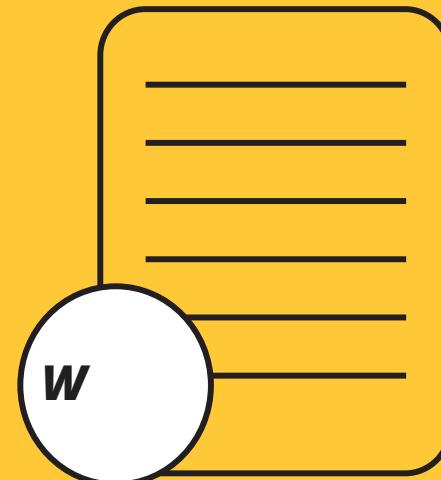


- When using the Case Designer, if routing information is added via the properties panel of a Collect Info or Approve/Reject step, the associated Assignment shape in that process flow is configured automatically.

## Case Management – Routing

Case routing to an Operator's **Worklist**

- A case is routed to a user
- System creates an assignment object in memory
- The assignment appears in the operator's worklist
- When the operator completes the action, the assignment is resolved



Worklist

## Case Management – Routing

- Case routing to a **Work Queue**
  - A case is routed to a queue
  - System creates an assignment object in memory
  - The assignment appears in the work queue list
  - It stays there until it gets assigned to an operator
  - The assignment moves to the operator's worklist
  - When the operator completes the action, the assignment is resolved

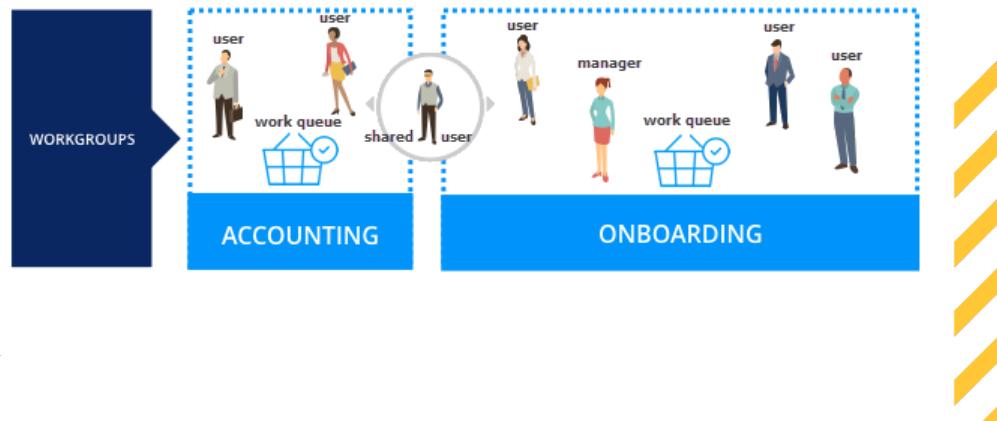


Work Queue

# Work group

Definition and description

- A cross-functional team that contains a manager, users (operators), and a work queue.
- The work queue contains shared work and resources across the business.
- Operators associated with a work group can share work among operators in different business units.
- A work group identifies one operator as the work group manager. Managers have the option to assign, monitor, and report on work performed in each work group.
- Work groups can contain authorized managers to help transfer assignments.
- The **case manager portal** refers to work groups as **teams**.



## Case Management – Routing

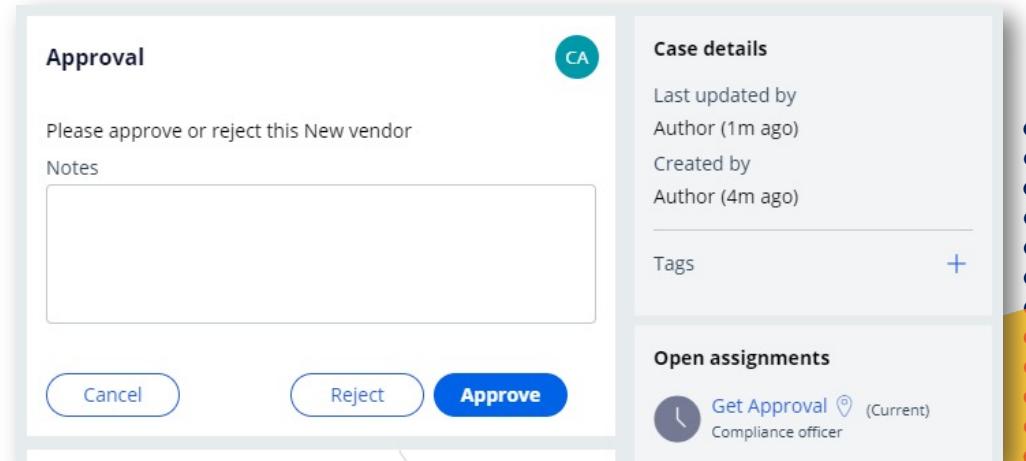
Some Standard Routing Activities available out of the box in Pega 8:

- ToWorklist
- ToWorkbasket (this is still a valid router – to be renamed ToWorkQueue)
- ToDecisionTable
- ToDecisionTree
- ToSkilledGroup
- ToLeveledGroup
- ToOrgUnitManager
- ToCostCenterManager

# Case approvals

## Definition

- Case approvals are decision points at which one or more users decide whether to approve or reject a case.
- Approvals can be added to a business process to seek consent from users with different roles or levels of expertise.
- Email and mobile push notifications can be configured for approvals.

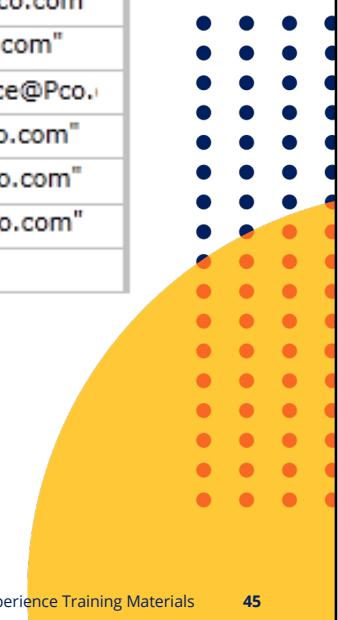


# Approval flow type

## Description

- Single level
  - Receive approval from a single user by indicating a specific user, work queue or business logic.
- Cascading
  - Configure reporting structure or authority matrix
  - Receive approval from people on different levels of your organizational chart or in different parts or departments of your organization.

Conditions	Actions
○ <b>TotalPRCost &gt;=</b>	○ <b>Approver ID</b>
○ when 0	→ "CSM@Pco.com"
○ when 25000	→ "VP@Pco.com"
○ when 75000	→ "VPFinance@Pco.com"
○ when 100000	→ "SVP@Pco.com"
○ when 500000	→ "CFO@Pco.com"
○ when 500001	→ "CEO@Pco.com"
otherwise	→ ""

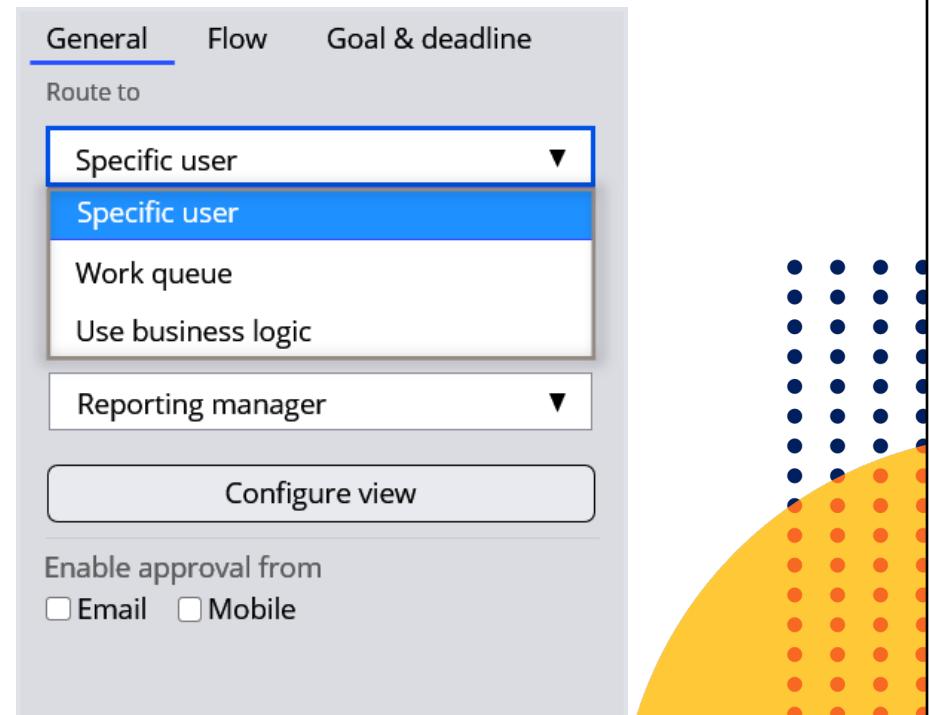


© 2022 Pegasystems Classroom Experience Training Materials 45

# Single level approval

## Description

- Assign approvals to the worklist of a specific user, a work queue, or use business logic to assign work based on custom conditions.
- **Specific user**
  - A single user will be assigned the task of approving or rejecting an action.
- **Work queue**
  - A list of approvals for a group of users, usually managers, assigned the task of approving or rejecting an action.
- **Business logic**
  - Create custom routing options to ensure approvals are routed to the most appropriate user, usually a manager.



# Cascading Approval Models

## Definition

- **Authority matrix** – Used when a set of rules directs the approval chains to individuals both inside and outside the organization of the submitter
- Implemented as a page list property of the case type
- A property of the page list class is specified as containing the approver ID
- The page List is used as the source of approvers for a cascading approval

If Amount Is Billable Time?	Route To
	Yes Accounts Payable
>\$50 USD	Manager
>\$500 USD	Director
>\$2000 USD	VP



Property: Approver List [Available]  
CL: WIND-Auto-Work-EvaluateAndSellAVehicle ▾ ID: ApproverList

General Advanced Specifications History

Property type

Page List

Page definition ★ WIND-Auto-Data-ApproverDetails

ApproverList  
ApproverDepartment  
**ApproverID**

- **Reporting structure** – Approvals always move up the reporting structure of the submitter or another defined list



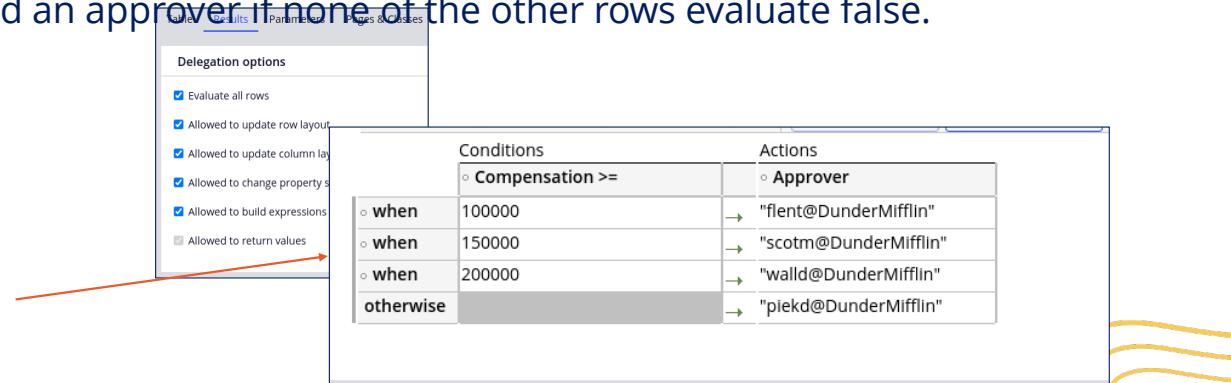
# Authority Matrix - Configuring Decision Table

Implementation

## Set Evaluate all rows option and configure the logic

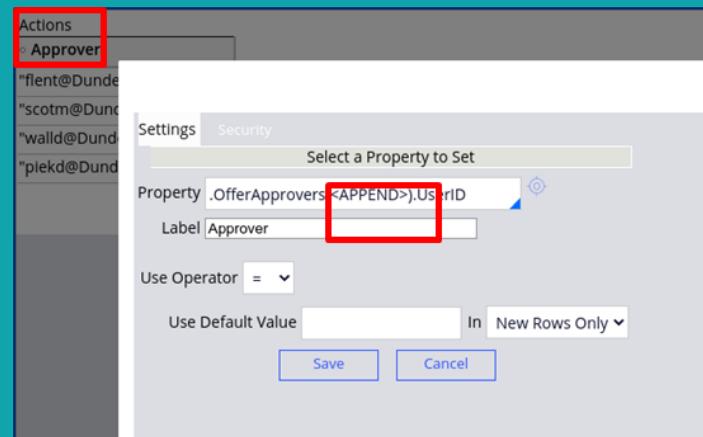
- The first page of the authority matrix page list will be populated with the Action value of the first true row from the decision table each subsequent true row will add another page with the given approver.
- The **otherwise** row will only add an approver if none of the other rows evaluate false.

Conditions	Actions
○ Compensation >=	Return
○ if	200000 → A
○ else if	150000 → B
○ else if	100000 → C
otherwise	→ D



# Authority Matrix - Decision Table and Symbolic Keywords

## Description



- The **Actions** column needs to be configured to append a new page to the list and set appropriate property for each true row.
- Pega provides a set of **symbolic indexes** to access items in a page list without using an explicit index number

<b>&lt;APPEND&gt;</b>	To add an element to the end (highest index value) a Value List or Page List property
<b>&lt;CURRENT&gt;</b>	In that context, the <CURRENT> index identifies the index value for the current iteration
<b>&lt;INSERT&gt;</b>	Use the <INSERT> keyword followed by an integer to insert a new element and its value into a Page List property at a numeric index position. Elements with same or higher index value are "pushed down" by one
<b>&lt;LAST&gt;</b>	To set or retrieve an element value from the end (highest index value) a Data relationship property
<b>&lt;PREPEND&gt;</b>	Use the <PREPEND> keyword insert a new element and its value into a Page List property as the first element. All existing elements are "pushed down" by one

# Reporting structure

## Definition

- Type of cascading approval that allows receiving approval from people on different levels of an organizational chart.
- Determines which operator is the starting point in the reporting structure.
  - Reporting manager
    - Reporting manager of current user
  - Workgroup manager
    - Manager of the current user's default workgroup

The screenshot shows two overlapping windows. The top window is titled 'Reporting Structure' and displays an organizational chart. The bottom window is titled 'Step' and shows configuration options for a cascading approval flow.

**Reporting Structure:**

- Legend:
  - Yellow box: Highlighted operators shows reporting structure
  - Blue box: Highlighted operators are not available to receive work
- Operators listed:
  - David Wallace, CEO
  - Toby Flenderson, Head of Human Resources
  - Stanley Hudson, Sales Rep
  - Kelly Kapoor, Customer Service Representative
  - HRBot1, Robot
  - HRBot1, Robot
  - Phyllis Vance, Sales Rep
  - Michael Black, Regional Manager
  - Michael Scott, Regional Manager
  - Pam Beesly, Secretary
  - Jim Halpert, Salesman
  - Pega Student
  - Jan Levinson, Regional Manager
  - Creed Bratton
  - Suvarchala
  - Case Management Admin

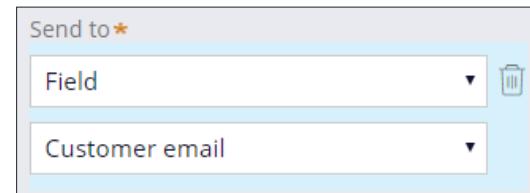
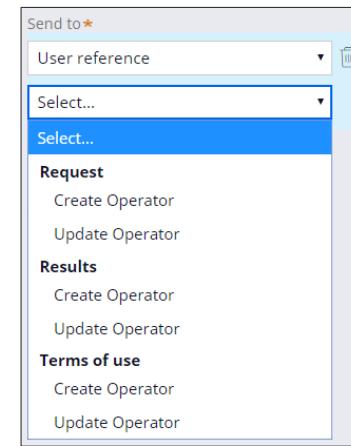
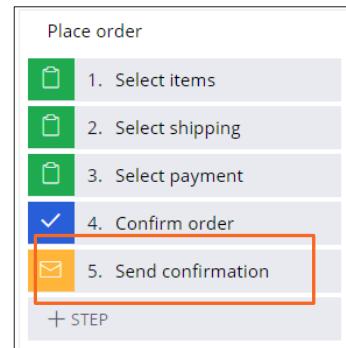
**Step Configuration:**

- Step tab: General
- Approval flow type: Cascading
- Approval based on: Reporting structure
- Approval to be completed by:
  - Workgroup manager (selected)
  - Reporting manager
  - Workgroup manager (highlighted in blue)
- Enable email approval
- Configure view button

# Correspondence - Identify who

## Definition

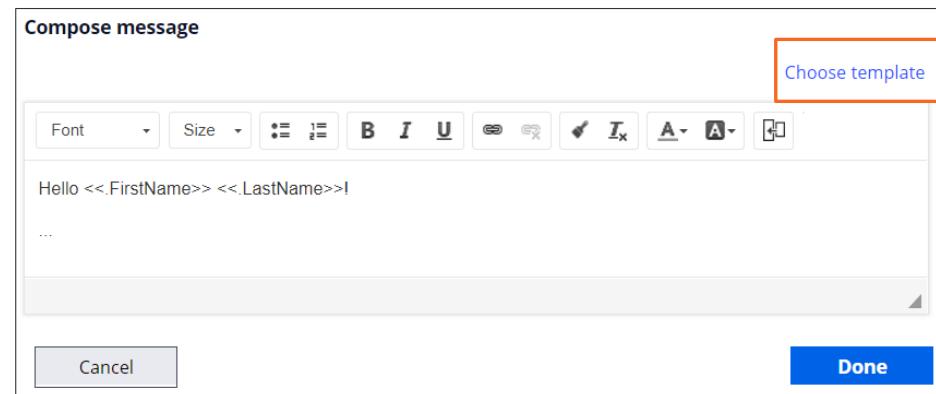
- When configuring correspondence, first determine, ***With whom do I need to communicate?***
- Sending to:
  - Email address**
    - Must update the application if email address changes.
  - Field**
    - A property with email address as the value.
  - User reference**
    - A list of existing users in an application.
  - Participants**
    - people, businesses, and organizations that are involved in a case.



# Correspondence - Identify how

## Definition

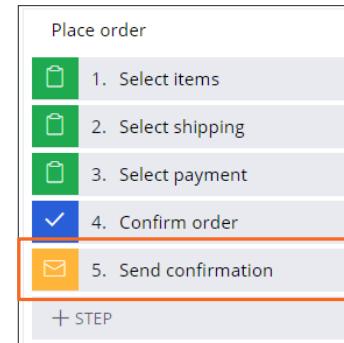
- Determine ***How will the communication be composed?***
- Pega Platform provides a rich text editor to create formatted email correspondence.
- The editor allows reuse of case data in the email.
- Pega Platform provides out-of-the-box templates that use case processing data, such as the case ID and case status.  
Templates such as:
  - Confirmation and thank you notices
  - Past deadline and goal notifications
  - Rejection and resolution details
  - Status updates



# Correspondence - Identify when

## Definition

- Determine ***When does the communication need to be sent?***
- Pega Platform simplifies sending correspondence by allowing configuration of the **Send email automation step** to a case.
  - Sends an email automatically to the selected parties.
- Another option is to configure email notifications for case level notifications.
  - Automatically sends a notification when an assignment in the case is routed to a user worklist.
  - Automatically configured to send to the user associated with the worklist.



Notifications  
Manage the events that send notifications to users

Email notifications

Email user when assignment is routed to worklist

Email  
 Custom       Use existing

Subject

If left blank, default will be used. Default: .pyLabel + "(" + .pyID + ")" moved to your worklist

Message  
[Compose message](#)      Message preview will appear here

# Correspondence types

## Definition

Pega Platform provides 4 correspondence types to send automated communication with users:

- **Email**

- Outgoing message may contain HTML, attachments or only text.
- System must contain an email account data instance and connect to a mail server.



- **SMS phone text**

- Short outgoing text messages to be sent to digital cell phones.



- **Fax**

- Outgoing letters to be sent by fax transmission.

- **Printed letter**

- Outgoing postal letters to be printed.

# What is a Duplicate Case?

## Definition

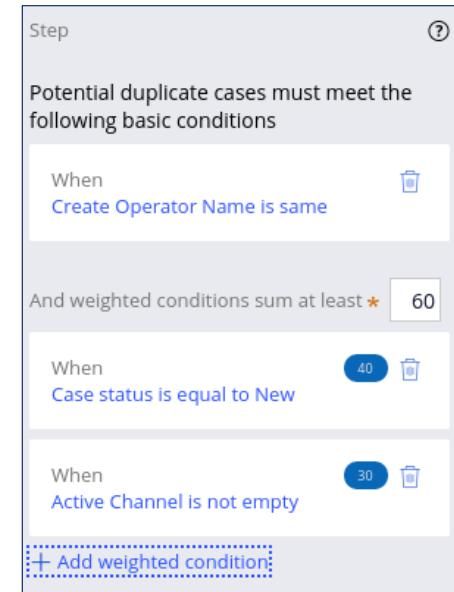
- A user may create a case which has the same data values as another case in the system.
- For instance, two purchase requests may have the same request date, the same items, or the same customer's name. However, if a specific combination of data values match, the new case is possibly a **duplicate case**.
- Pega provides the **search duplicate cases step** to help users identify and resolve duplicate cases.
- This process is implemented in the case life cycle as a **Search duplicate cases step**.



# Duplicate Search - From Case Designer

## Implementation

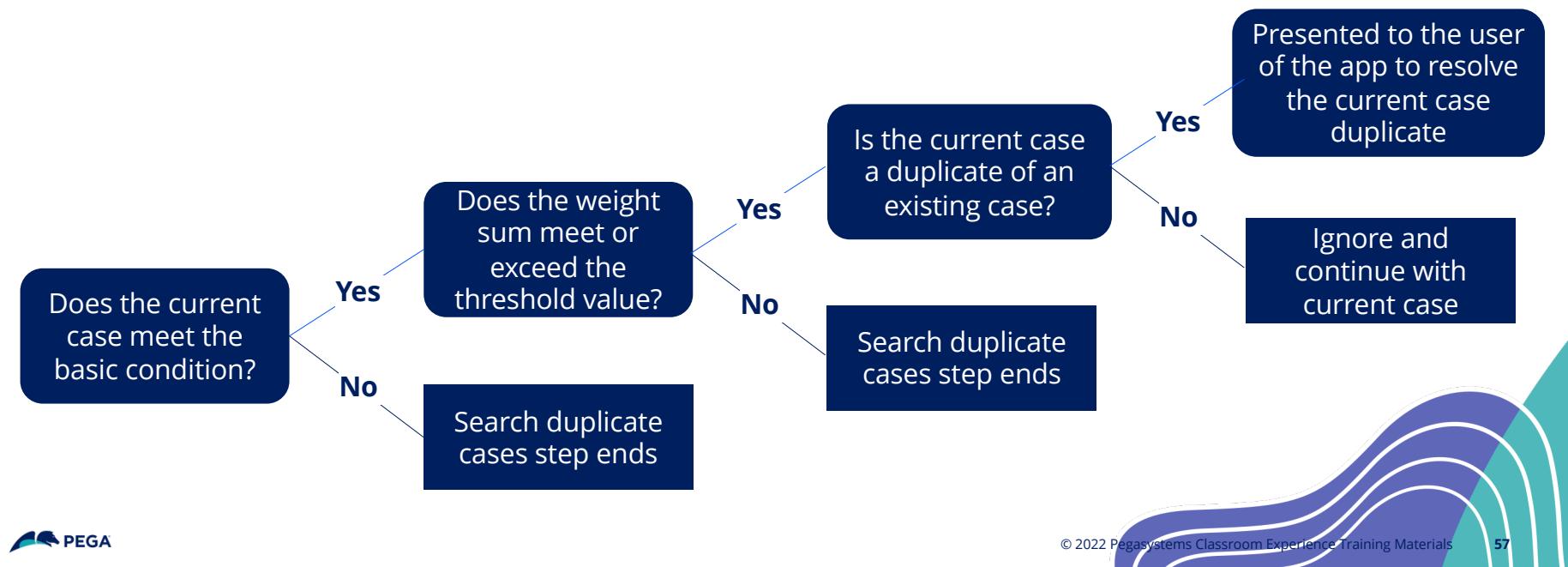
- Configure basic and weighted conditions.
- Basic conditions specify criteria that duplicates must meet.
  - Useful for reducing the number of cases to be matched and improving performance
- The sum of the weighted conditions must be equal to, or greater than a threshold.



# Functionality of the Duplicate Search

## Description

When a case enters the **Search Duplicate Cases** step, the system uses **basic conditions** and **weighted conditions** to compare specific property values with cases already present in the system.



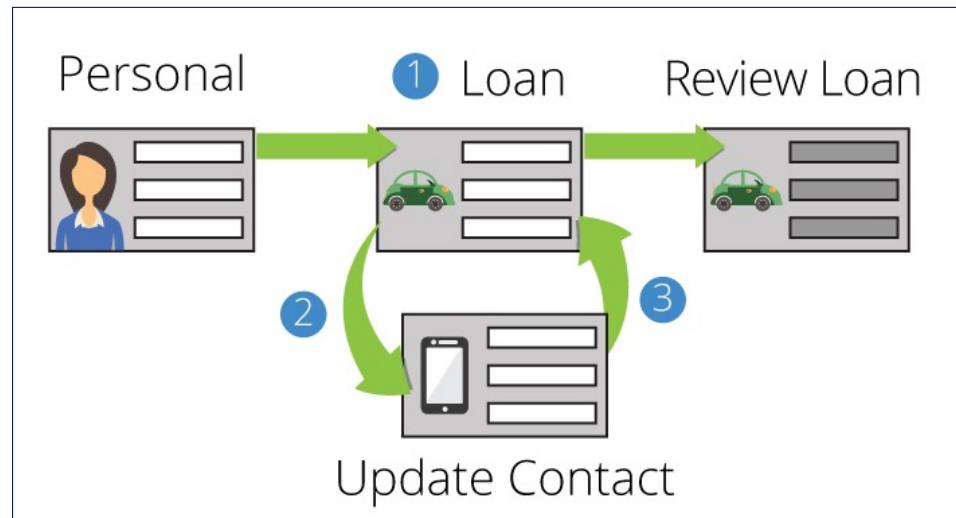
## Optional Action - Use Case

- Use Optional Actions to allow an end user to start a new process or call a screen to interact with that is outside of the predefined case life cycle sequence.
- For example, while reporting a car accident to an insurance company, the customer mentions having a new phone number. The customer representative uses an optional action to update the customer's contact information.



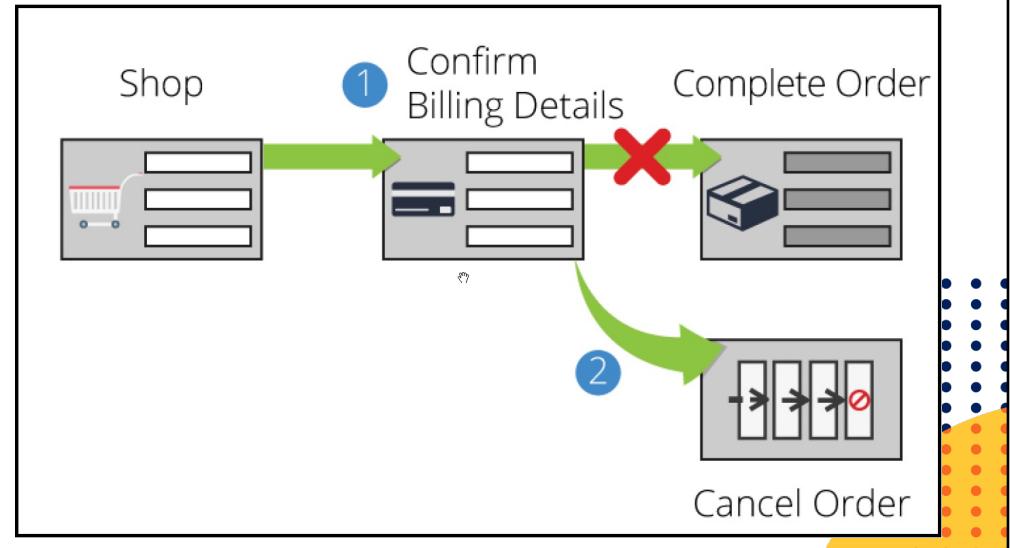
## Case Management – Optional Action (Local Action)

Perform a single task without interrupting the primary path.



## Case Management – Optional Process (Flow)

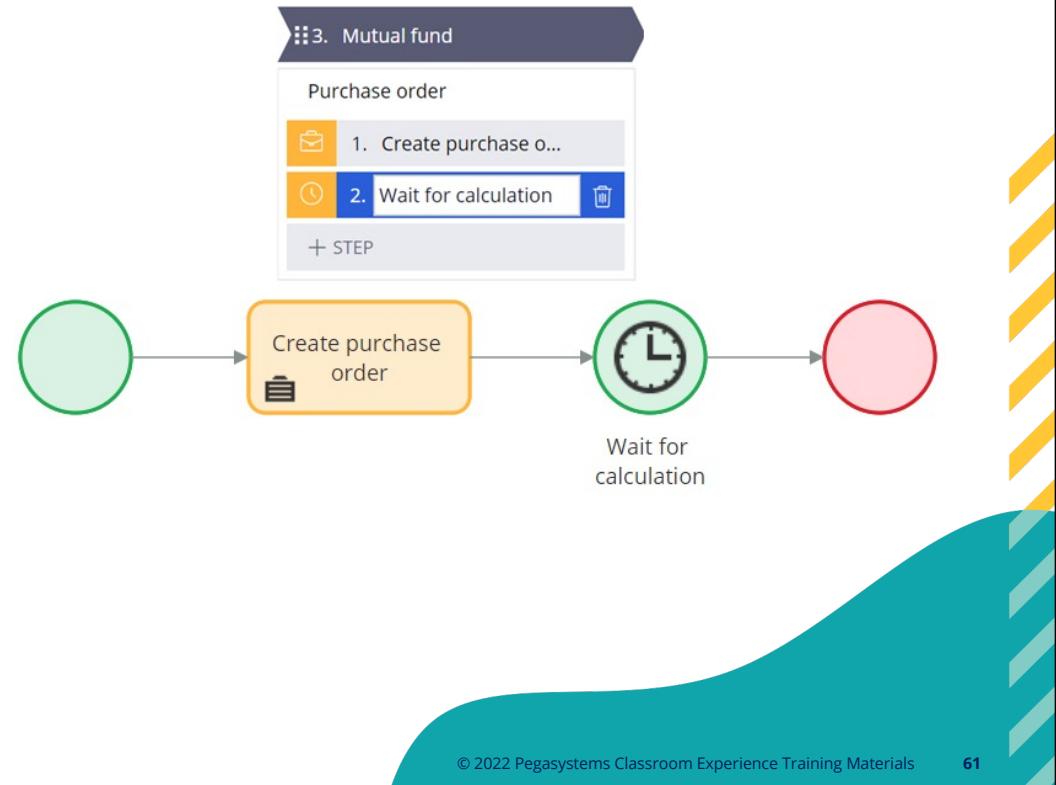
- Use optional process when a user is allowed to determine if a new process needs to be called.
- Used when multiple steps are needed to run a new process within the case.
- After completion, the process may, or may not return to the primary path.



# Wait step

Definition

- Enforce dependencies using the Wait step.
- Allows pausing and resuming case processing when the case meets conditions that defined.
- The ability to pause processes can facilitate the resolution of complex cases that require other processes to end.



# Wait type

Definition and description

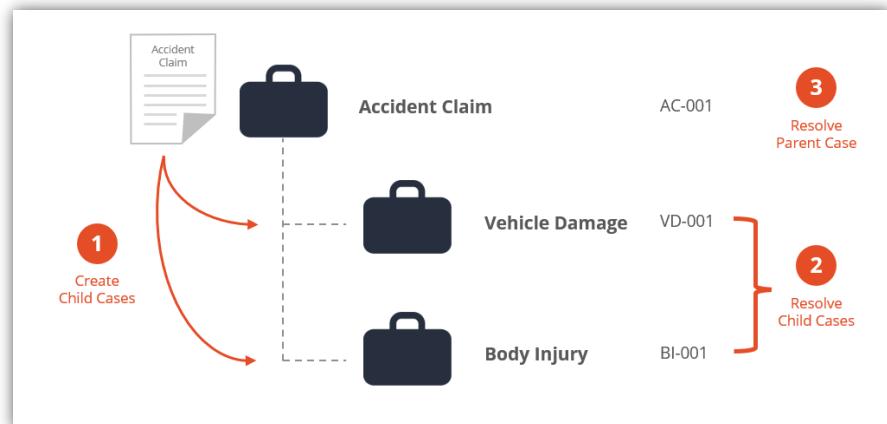
- The Wait step can be configured to pause case processing based on **Wait type**:
  - Case Dependency
    - When a parent case reaches the Wait step, the case pauses until all child cases, or any child case of a given type reach a designated status.
    - The status could be a standard status like Pending-approval or a custom status.
    - **To be resolved:** status is set to a value that starts with the word Resolved.
  - Timer
    - Pauses a case until the Set date/time interval expires or until a Reference date/time is reached



# Child case benefits

## Description

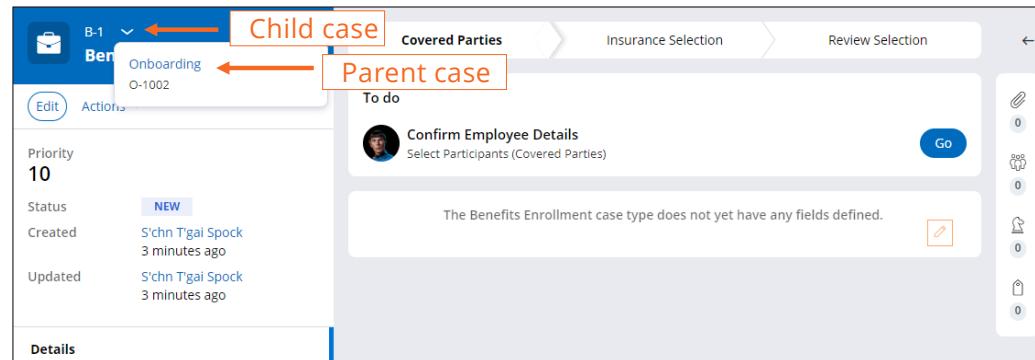
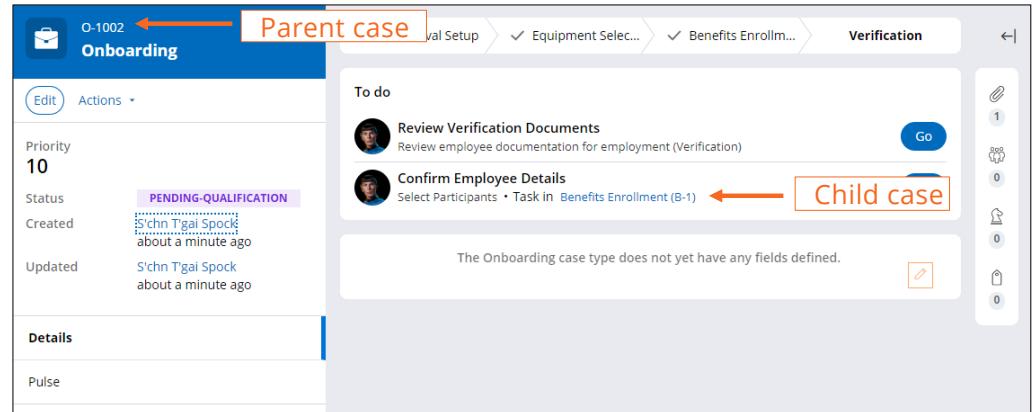
- **Child cases** are **beneficial** in situations for modeling work separately from the **Parent case** when several of the items in the list below are true:
  - Different data model is needed
  - Different life cycle is needed
  - Separate case ID and status for reporting is necessary
  - Separately assigning the case is needed
  - Must complete before the parent completes
- It would likely be appropriate to design a child case versus a subprocess when most of the items in the list exist.



# Child case

## Implementation

- Add a create case step or shape to a process.
- Configure Create case to create a separate case or a child case
- Run the Case Type to the point where the new case will be created
- Click on Actions > Refresh to view the open assignments which should display the child case
- Select the child case assignment which will display the parent case ID (link) above the child case ID



# Child Case Data propagation

## Description

When creating a child case, you can also specify the information to copy from the parent case to the child case through a process known as *propagation*.

Data Propagation  
in App Studio

- Identify the fields in the parent case **to transfer information to a new child case**.
- Option to “*transfer information to new case*”

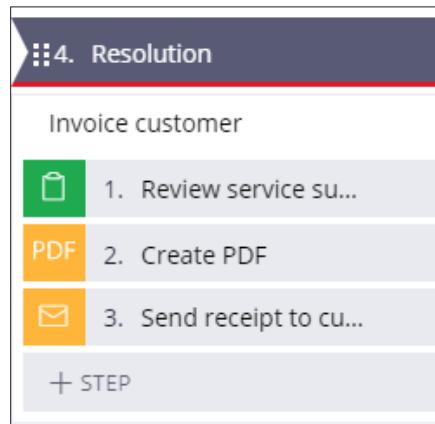
Data Propagation  
in Dev Studio

- Use a **Data Transform** to **copy the data values to fields identified** in a parent case to a child case.
- Option under the **Settings tab select Data Propagation**

# PDF file generation and attachment

## Definition

- During a case life cycle, you can generate PDF files and attach them to the case.
- The Create PDF automation attaches a screenshot of a view to the case in PDF format.
- The view used must exist prior to configuring the automation.
- During execution, the view is rendered, and the screen gets captured and converted to a PDF file.



Section name\*  
ReviewServiceSummary

Description \*

Attachment Category  
pxDocument

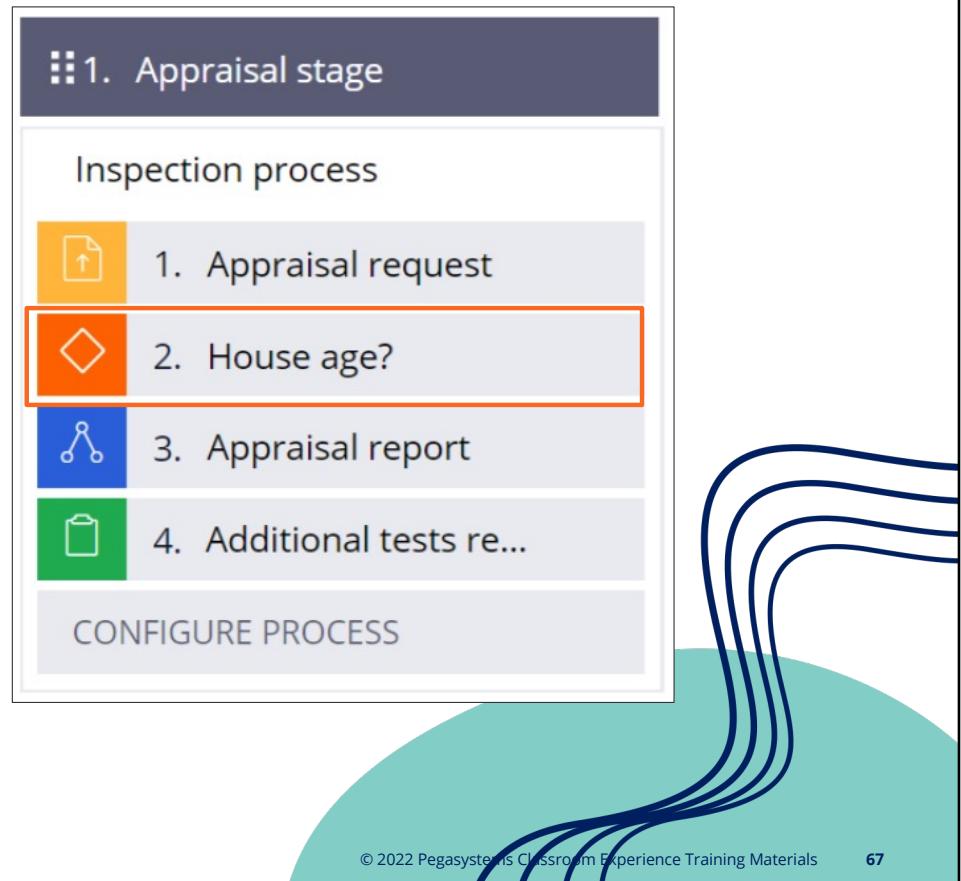
PDF Orientation  
 Landscape  
 Portrait

A screenshot of the "Create PDF" configuration dialog. It includes fields for "Section name\*" (ReviewServiceSummary), "Description \*" (Receipt of Services), "Attachment Category" (pxDocument), and "PDF Orientation" (with "Landscape" selected). There are also "Review service su...", "Send receipt to cu...", and "+ STEP" buttons at the bottom.

# Decision points

Definition

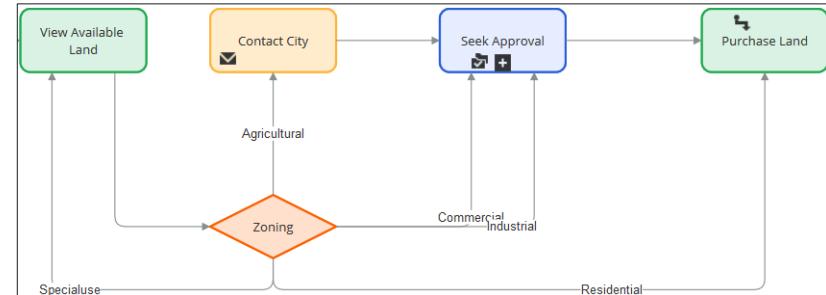
- Evaluates an expression or calls a rule to determine which step is next in the flow progression.
- Decision shapes create alternate paths in a flow because they can have more than one outgoing connector (points).
- Allows creation of different types of processes, or flows, based on the shapes or steps and how a process is integrated within the case life cycle.



# Decision Table

## Definition and description

- Use to derive a value that has one of a few possible outcomes, where each outcome can be detected by a test condition.
- The table consist of two or more rows, each containing test conditions, actions (optional), and a result.
- At run time, the system evaluates the rows starting at the topmost row:
  - Conditions in a row evaluate to false, processing continues with the next row, Actions and Return ignored.
  - All the conditions in a row evaluate to true, Actions and Return are processed.
- Evaluate All Rows
  - not selected, processing ends, returns the value in the Return column as the value of the entire rule.
  - is selected, processing continues for all remaining rows, performing the Actions and Return for any rows which are true.

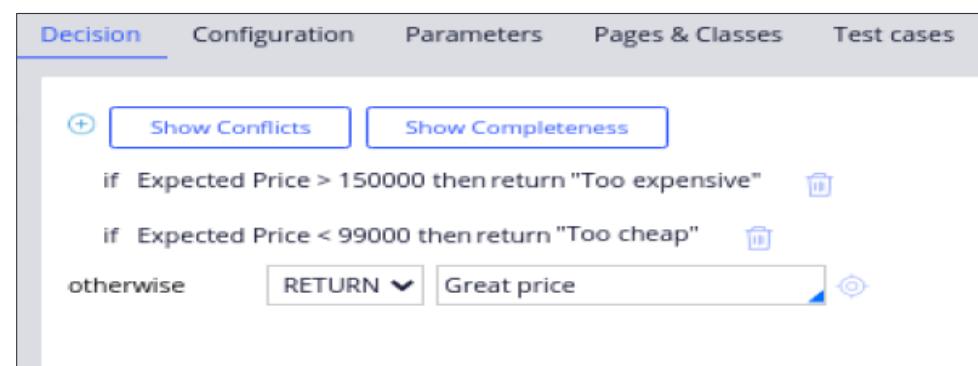
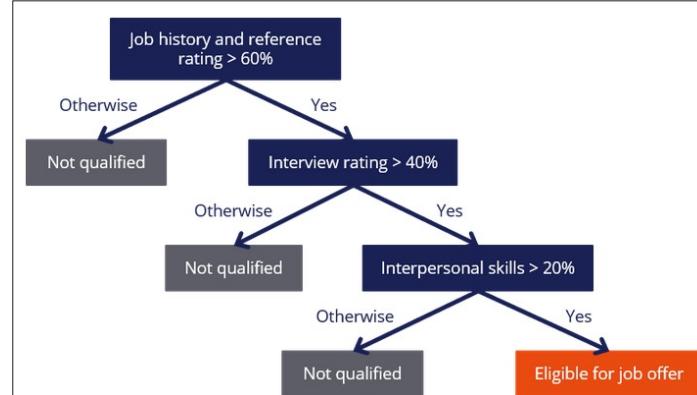


Conditions		Actions
if	"Office Building"	Industrial
else if	"Farm Land"	Agricultural
else if	"Multi-Dwelling"	Commercial
else if	"Single Dwelling"	Residential
otherwise		Special-use

# Decision tree

## Description

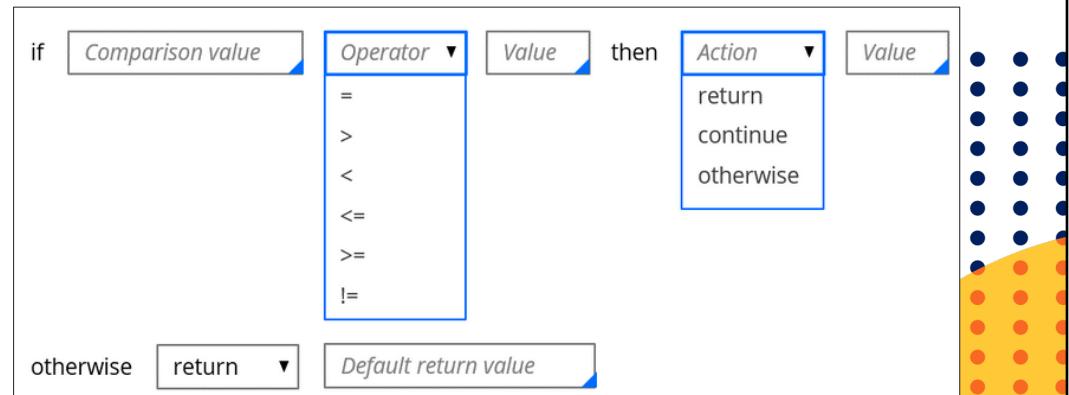
- Used to define comparisons by using a tree structure.
- Contains nested if... then... else conditions to specify a series of tests performed on property values to allow an automated decision.
  - To configure, in the **then** list, select **continue**.
  - Continue** causes the next row of the decision tree to be nested.
  - Each indent level supports comparisons against a single value.



# Tree conditions

## Implementation

- Configure single ***if*** statements with return values.
- Configure nested ***if*** statements with ***continue*** statements.
- **Comparison value** is a property or expression.
- Comparison **operators** are conditional symbols.
  - ( e.g. <,>, <=, >=, =, !=)
- **Value** is a fixed value, property or express
- **Action** can return a result, continue the evaluation, or stop the evaluation.



# Decision rule conflicts

The decision table and decision tree rule forms include the ability to test for **conflicts**.

## Show conflicts

- Identifies potential gaps in the decision rule execution by identifying conditions that may not be tested during execution.
- A warning is displayed on the row, which causes the conflict to specify the condition that did not evaluate.

### Decision Table conflicts

A screenshot of a decision table interface. The table has two columns: 'Conditions' and 'Actions'. The 'Conditions' column contains radio buttons for 'Credit Score >' and 'Outstanding Balance <'. The 'Actions' column lists 'Return', 'Approval Level 1', 'Approval Level 2', 'Approval Level 3', and 'Reject'. A row under 'if' has '900' selected. A row under 'else if' has '1000' selected, with a red warning icon next to it. A row under 'else if' has '500' selected. A row under 'otherwise' is highlighted in grey. A toolbar at the top right includes 'Select values' and 'Show conflicts' buttons, with 'Show conflicts' being highlighted with a red box.

Conditions		Actions
<input type="radio"/>	Credit Score >	Return
<input type="radio"/>	Outstanding Balance <	
if	900	→ Approval Level 1
else if	1000	→ Approval Level 2
else if	500	→ Approval Level 3
otherwise		→ Reject

### Decision Tree conflicts

A screenshot of a decision tree rule form. It shows a conditional structure with three branches. The first branch is 'if .Quantity > 100 then return "Reject" and Take Actions'. The second branch is 'if UnitPrice < 50 then return "Approve"'. The third branch is 'if UnitPrice <= 25.00 then return "Approve"'. The final branch is 'otherwise RETURN "Refer" and Take Actions'. The 'Show Conflicts' button in the toolbar is highlighted with a red box.

+ Show Conflicts Show Completeness

if .Quantity > 100 then return "Reject" and Take Actions

if UnitPrice < 50 then return "Approve"

if UnitPrice <= 25.00 then return "Approve"

otherwise RETURN "Refer" and Take Actions

# Decision rule completeness

The decision table and decision tree rule forms include the ability to test for **completeness**.

## Show Completeness

- Identifies a decision table that has missing conditions or a decision tree that has missing branches.
- Automatically adds rows that cover additional cases, rows can be altered or eliminated.
- You can add returned results as additional rows if the decision rule needs a more detailed evaluation of the values.

The screenshot displays two interface panels for completeness analysis:

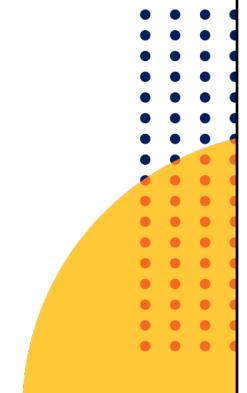
**Decision table completeness:** A grid table with columns for Conditions and Actions. The Conditions column includes rows for 'if', 'else if', and 'otherwise' statements, each with specific conditions like 'CustomerLevel = Bronze' or 'Customer Revenue > 100000'. The Actions column shows corresponding return values like '.03', '.02', '.01', or '0'. The 'Show completeness' button at the top right is highlighted with a red box.

	Conditions	Actions
if	CustomerLevel = Bronze	.03
else if	CustomerLevel = Gold	.02
else if	CustomerLevel = Silver	.01
otherwise		0

**Decision tree completeness:** A hierarchical tree structure where each node represents a condition like 'Customer Level = "Bronze" then continue'. The tree branches into further nodes for different levels (Gold, Silver) and revenue thresholds. The 'Show Completeness' button at the top left is highlighted with a red box.

# Case Management - App Studio

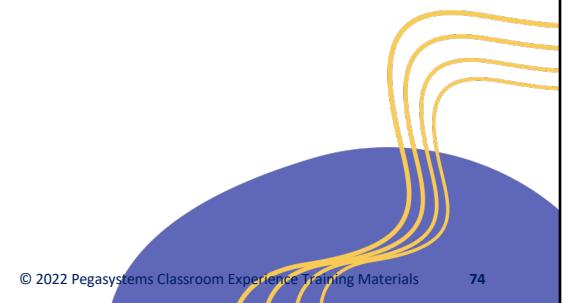
- Focuses on application development:
  - Case Design
  - Data and integrations
  - Channels and interfaces (mobile, email, chat bots, etc.)
  - UI authoring
- Supports real-time UI design as you process cases (helpful for reviews with stakeholders)



# Case Management - Dev Studio

Focuses on advanced functionality:

- System settings
- Complex rules - rule form access
- Security
- Component reuse (across studios)
- Collaborative, branched development
- Versioning and source control



## Case Management - Development Approach

- Low-code / no-code
- Model driven
- Visual tools
- Business users work directly with IT developers

## Case Management 38% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=gev6228cc8eba137>



# Data and Integration

15%

## **Data and Integration 15%**

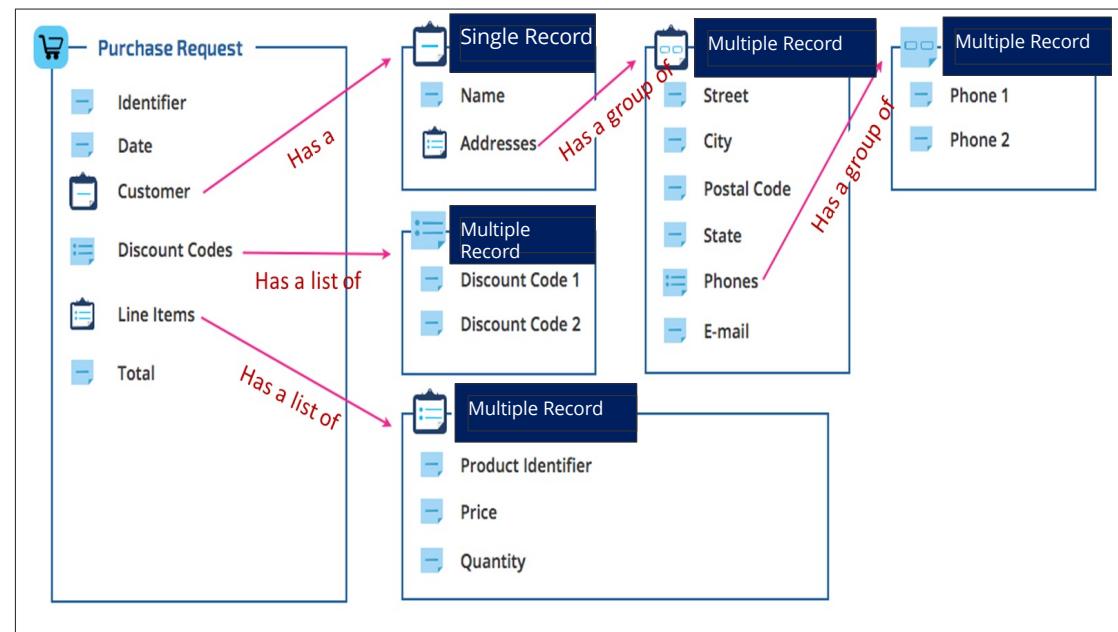
- Create data objects, data relationships, and field types
- Identify and create calculated values
- Identify the role of data records in applications
- Validate data: create and configure data validation rules using business logic
- Capture and present data: fields and views



# Data Modeling

## Use Case

- The data model brings information into your application in a format that makes sense for a business.
- Data modeling allows
  - viewing data and case type relationships.
  - creating new data types.
  - updating data type records.



# Data Records versus Case records

Description

## Data records

- Represents key business entities (e.g. Customers)
- Contain all the fields necessary to describe it
  - Identifier
  - Name
  - Address
  - Date of creation
  - And more

Data object is a template of the data that will be stored as a data record.

## Case records

- Represents business transactions (e.g. Assistance Request)
- Contains all the fields necessary to describe it
  - Customer (who needs assistance) (Identifier, Name, Address, Date of Creation, etc.)
  - Location of vehicle (address, city, state)
  - Type of vehicle (make, model, year, color)

Case type is a template of the business process which captures the data that is stored as a case record.

# Local data storage and External data sources

Description

## Local data storage

- pyGUID – Globally unique ID created by default when created in App Studio
- Stored locally in a database table in Pega. Adding records in AppStudio or DevStudio creates a record in the mapped local database table in Pega. Deleting records removes them from the database table.

## External data sources

- Retrieve records from an external database or web service.
- Configured using data pages.

Data model				
ID*	Name	Country		
ATL	Atlanta	United States		
BER	Berlin	Germany		
HQ	Cambridge	United States		
LON	London	United Kingdom		
TOK	Tokyo	Japan		
VAN	Vancouver	Canada		
<a href="#">+ Add record</a>				

# The integration designer

## Definition

The screenshot shows the Pega App Studio interface for the 'Showcase' application. At the top, there are two sections: 'Data model' (Visualize all the data in your application) and 'Integration map' (Visualize where data is coming from). Below these are two tabs: 'Data objects' and 'Data pages'. The 'Data objects' tab is selected, displaying a list of five data objects: Approver, Interviewer, Job Posting, Required Skills, and Skills. Each data object row includes a 'Referenced By' section showing relationships to other objects like Job Applicant and Interview, and a 'Systems of record' section indicating they are stored in Pega.

Data objects	Referenced By	Systems of record
Approver	Job Applicant, Interview	Pega
Interviewer	Interviewer, List Interviewer, +1 more	Job Posting, Job Applicant, Pega
Job Posting	Job Posting, List Job Posting	Job Applicant, Interview, Pega
Required Skills	List Required Skills, Required Skills, +2 more	Pega
Skills	List Skills, Skills	Pega

- Comprehensive view
- Displays data objects, case types and data types.
- Create data objects
- Access data objects
- Update data objects

# The data model

## Definition

The data model helps with understanding the relationships among case types, data objects, and properties in an application. To model data, you need the following components:

- **Data objects**

- Categories of data that have fields, field mappings, and connections to data sources.

- **Fields**

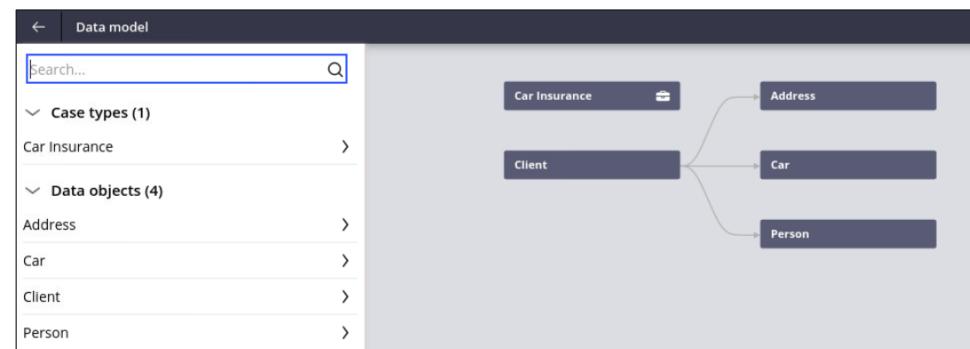
- Properties that store and format the data in your application.

- **Data sources**

- Resources, which can be real or simulated, that host the data.

- **Field mappings**

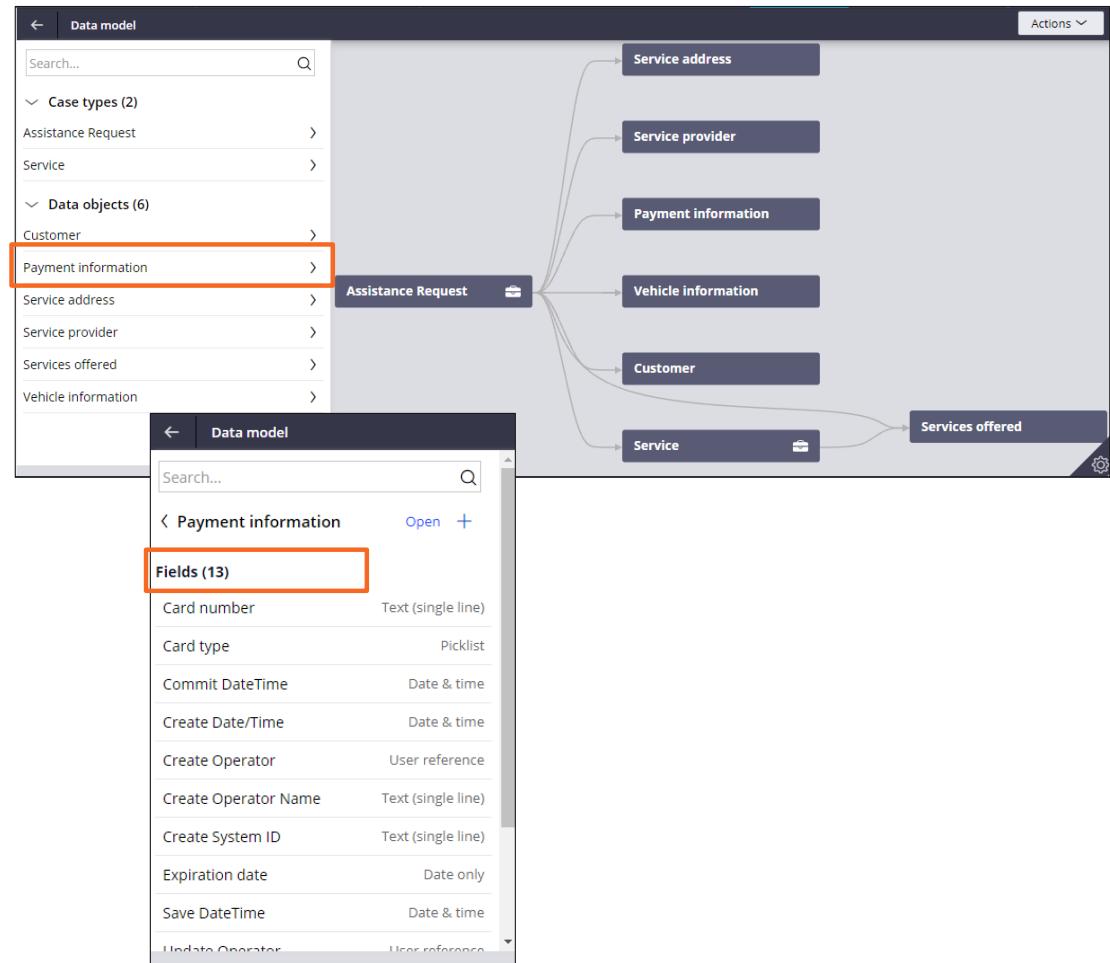
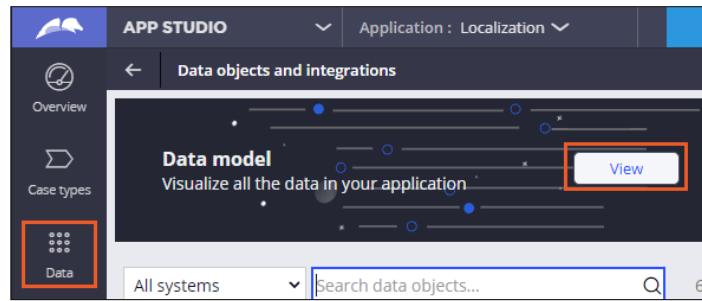
- Logic that links or transforms the fields in a data source to the fields in an application.



# View the data model

Navigation

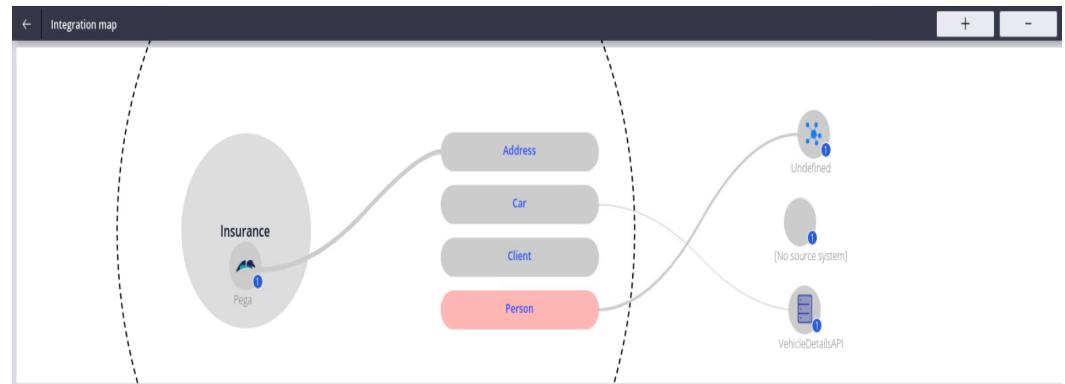
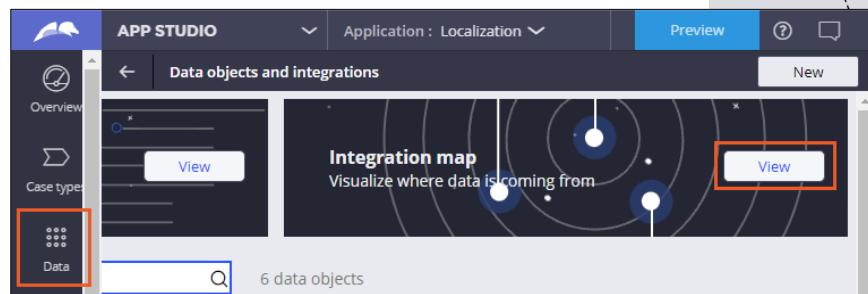
App Studio > Data > View



# View integration map

## Navigation

- AppStudio > Data > View
- Use the interactive integration landscape to understand the relationships between data and data types and systems of record.



# Declare Expression

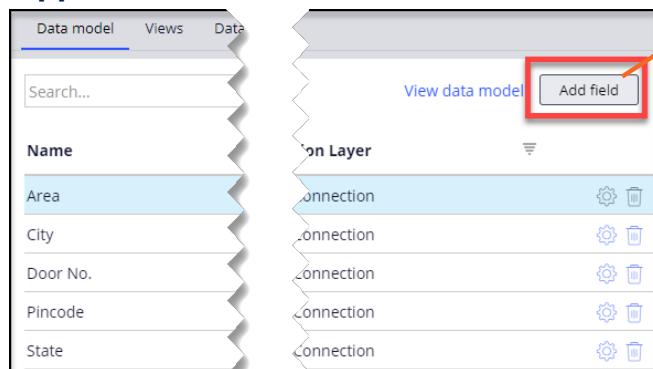
Navigation (1 of 2)

## From App Studio:

Data Model > Add field > **Advanced** section

- Select the check box to indicate it's a calculated field.
- Use Expression or a Decision Table from the calculation drop down. Click Submit.
- Verify the Declare Expression rule in the Dev Studio

## App Studio > Data Model > Add field



Add field to Current Address

Field name **\***  
Test

Type  
Text (single line)

Advanced

ID **\***  
Test

Description

Max length  
256

Expected length

This is a calculated field (read-only)

Calculation  
**Use expression**

Area + " " + .Pincode

Use '!' (dot) for field prompts to enter a simple equation such as:.Amount\*.Quantity

Cancel      Submit & add another      **Submit**

# Declare Expression

## Navigation (2 of 2)

- When a property is created as a calculated field in the App Studio, Pega automatically creates a Declare Expression in the Dev Studio.
- Switch to Dev Studio to view the Declare Expression in the Data class.

The screenshot shows the Pega Dev Studio interface with the following details:

- Top Bar:** Application: ServiceLevelAgreement, Configure, Launch portal, Create, Search, DEVELOPMENT.
- Sidebar:** DEV STUDIO selected. Sections include Recents, Case types, Data types, App, Records, and Favorites. Under Data types, WIND-Mobile\_C-Work is selected. In the Pinned classes section, WIND-Mobile\_C-Data is expanded, showing SysAdmin, CardDetails, CurrentAddress, Data Model, Decision, and Declare Expression. The Declare Expression item is highlighted with a red box.
- Central Area:** Title: Declare Expression: .Test [Available]. CL: WIND-Mobile\_C-Data-CurrentAddress. ID: .Test. RS: ServiceLevelAgreementC [Branch: ServiceLevelCLab]. Sub-sections: Expressions, Pages & Classes, Test cases, Specifications, History. The Expressions tab is selected. The Overview section shows "Build Expressions" with "No conditions defined". A "Set Test" dropdown is set to "Value of" and contains ".Area + \" \" + .Pincode".



# Declarative network display

## Navigation

### From Dev Studio

1. Configure menu > Select Case Management > Business Rules > Declarative Network.
2. Click application to select the app for review.
  - o To view the rule and related properties, click the "Display this top-level declarative network" icon.
  - o Open the declare expression, click the Display declare expression icon.
  - o Open the target property, click the property name.
  - o Open the class to which a rule belongs, click the class name.

Case Management: Business Rules

Declarative Network Analysis

1 application

- WIND-Auto-Work-EvaluateAndSellAVehicle Evaluate and Sell a Vehicle ( 3 Networks )

- Calculate Value : Whenever inputs change
- Calculate Value : Whenever inputs change
- .UserAgreement Calculate Value : Whenever used ⚠️

- WIND-Auto-Work-RVCalculation RV Calculation ( 1 Networks )

- .RoundedResidualValue Calculate Value : Whenever inputs change

## Declarative network analysis

### Description

- To identify the relationships between fields, the Pega Platform establishes and updates a network of calculations for an application.
- When you define a field calculation, the Pega Platform adds that calculation to the calculation network.
- This calculation network allows the Pega Platform to update all relevant fields whenever a value changes.
- Shows the target property and all potential inputs that might affect its final value.
- Unit test a Declare Expression rule to reduce the number of processing errors.

Users add an item to the list or update the quantity of an item already in the list. Pega Platform calculates the **Line total as .UnitPrice \* .Quantity**.

The updated line total triggers a calculation to update the **order total**, using the **Sum of function** to add each line total in the list of items.

You added the following items to your order:

Item	Quantity	Unit price	Line total
Team logo hat	3	USD 7.75	USD 23.25
Team logo magnet	4	USD 4.25	USD 17.00

Order total: USD 40.25

Tax (8%): USD 3.22

Your cost: **USD 43.47**

The updated order total triggers a calculation to update the **tax** applied to the order as the **value of .TaxRate \* .OrderTotal**.

The updated tax amount triggers a calculation to update the **total cost** as the **value of .Tax + .OrderTotal**.

# Validation

## Definition

**Validation** is used to ensure that data is correctly formatted and valid before being input into an application. This allows the system to process the information without errors.

Use validation rules when you cannot predict or control the value users enter in a form.

The data must:

- Be the right type
- Fit the business logic
- Be restricted to possible values

Validation conditions

Validate .Employee.pyPostalCode  Required  Enable conditions

Select a function Validation of [Property Name] using [Edit Validate Name] fails

If Validation of .Employee.pyPostalCode using USZipCode fails

+ Message Please enter a valid ZIP code

Continue validation

Cancel

*Pega also provides property types, controls, and rules to support validation requirements.*

# Validation - Case Data Model

Case Type > Data model tab > Validations

The screenshot shows the 'Edit case type: Job Applicant' interface. The 'Data model' tab is selected. A table lists properties across stages: Collect Resume, Recruiter Review, Interview, Decision, and Offer. In the 'Email' row under 'Recruiter Review', a red arrow points from a tooltip to a validation dialog box. The tooltip says: '→ Stage entry  
Please enter an email address to c...'. The validation dialog is titled 'Stage entry validation' and contains the following fields:

- When conditions are met: Email is empty
- Then display error message as: \* Please enter an email address to continue.
- To validate stage entry for the stages:
  - Collect Resume
  - Recruiter Review
  - Interview
  - Decision

Buttons at the bottom are 'Cancel' and 'Submit'.

# Field types

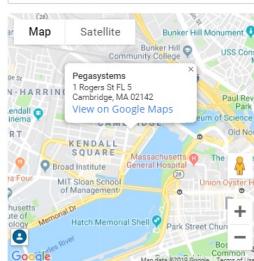
## Description

Field type	Type of data
<b>Text (single line)</b>	Any single line of text.
<b>Text (paragraph)</b>	A large text box that accepts multiple lines of text.
<b>Boolean</b>	Allows users to select a check box to indicate one of two possible responses.
<b>Currency</b>	Currency code and value are stored based on the default currency type.
<b>Date &amp; time</b>	UTC (Coordinated Universal Time) value normalized to Greenwich Mean Time (GMT). Date automatically displays in localized format.
<b>Date only</b>	Calendar date with a localized format.
<b>Decimal</b>	Numbers with a fractional component. Use this field type when fractions are needed.

Field type	Type of data
<b>Email</b>	Valid email format with a @ symbol. An email field is an action-oriented control, meaning the value stored in the field displays as a link.
<b>Integer</b>	Positive and negative whole numbers, including the value zero (0).
<b>Phone</b>	Digits display in a localized format. A phone field is an action-oriented control, meaning the value stored in the field displays as a link.
<b>Picklist</b>	A list of predefined values.
<b>Time only</b>	UTC (Coordinated Universal Time) value normalized to Greenwich Mean Time (GMT).
<b>URL</b>	Web address. A URL field is an action-oriented control, meaning the value stored in the field displays as a link.

# Field types

Description

Field type	Type of data	Example
Attachment	Document or file.	<p>Attach resume</p> <p><input type="text" value="my-resume"/> <input type="button" value="Attach"/></p> <p></p> <p><input type="button" value="Download"/> <input type="button" value="Delete"/></p>
Location	Address input or automatic geolocation.	<p>Office location</p> <p><input type="text" value="1 Rogers St FL 5 Cambridge, MA 02142"/></p> <p></p>
User reference	Enter or select a user ID that exists in the system.	<p>Current user</p> <p><input type="text" value="User@TGB"/> <input type="button" value="X"/></p>

# Field types and data object location

## Implementation

Data relationship field type	Data Source	Use Case
Embedded data	User-supplied data such as a name and address sourced from inside a case type.	A company needs to capture shipping addresses.
Query	A data page or view that is not sourced from inside the case type. The data page defines that the Query data relationship is configured to use.	An application needs to update the current weather.
Case reference	Single or multiple records from a selected case type.	A user selects from a list of service cases from the Service Case type.
Data reference	Single or multiple records from a selected data page.	A user selects from a list of products to order.

# Data Relationships

## Implementation

- .CreditCards is a multiple record data relationship
- It is being used by .Customer which is a single record data relationship
- .Customer
  - .FirstName
  - .LastName
  - .CreditCards()

**Customer**

First Name  
Sandra

Last Name  
Washington

**Credit Cards**

+ Add item × Delete

	Card Number	Expiration Date	Verification Code
1	0000 0000 0000 0000	01/2025	321
2	1111 1111 1111 1111	02/2024	456

# Data and Integration - Referencing a Property

Refer to a property in Pega by prefixing the property name with a “.” (period or dot).

<b>Value mode</b> properties	.OrderDate	• A single value property named <b>OrderDate</b>
	.Phone(Mobile)	• An entry in a value group property, such as the mobile phone number where Mobile is the group subscript
<b>Page mode</b> properties	.DiscountCode(1)	• The first entry in a value list property, such as one of the discount code
	.Customer	• A page that contains customer information
	.Address(Work)	• An entry in a page group property, such as the work address, type
	.LineItem(3)	• The third page of a page list that contains purchase request line items, type

# Standard property names

## Description

- Pega comes with a set of standard property rules.
- The prefix identifies how the standard property can be used.
- System property names start with ***px***, ***py*** or ***pz***.
  - **Px** - properties that users see on a form, but cannot directly enter or change values, i.e. ***pxCreateDateTime***.
  - **Py** -properties that users can explicitly enter or change i.e. ***pyDescription***.
  - **Pz** -properties that are reserved for internal use that end users cannot see, enter, or change i.e. ***pzInsKey***.

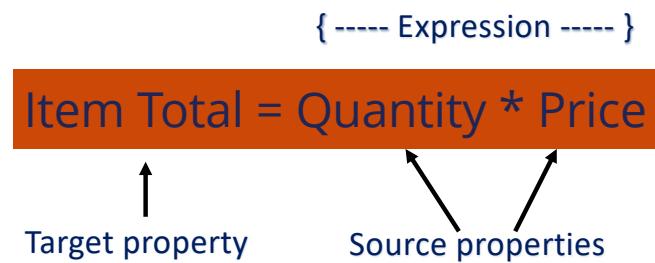
<a href="#">pxUpdateSystemID</a>	pega
<a href="#">pxUrgencyPartyTotal</a>	0
<a href="#">pxUrgencyWork</a>	10
<a href="#">pxUrgencyWorkClass</a>	10
<a href="#">pxUrgencyWorkSLA</a>	
<a href="#">pxUrgencyWorkStageSLA</a>	0
<a href="#">pxUrgencyWorkStepSLA</a>	
<a href="#">pyAgeFromDate</a>	20200708T164935.621 GMT
<a href="#">pyCancelLabel</a>	Cancel
<a href="#">pyConfirmationNote</a>	pyStepRoutedConfirmation
<a href="#">pyCustomerSatisfiedTimestamp</a>	
<a href="#">pyDocumentTitle</a>	
<a href="#">pyElapsedCustomerUnsatisfied</a>	6.0
<a href="#">pyElapsedStatusNew</a>	6.0

## Data and Integration - Identify the User Tasks

- Create a view in which users enter the specified information.
- Before you create a view, ask yourself the following questions:
  - What fields do users need to see?
  - How will users enter values in those fields?
  - Can users modify the field values or only read the values?

## Data and Integration - Declarative Expressions

- **Declare expressions** automatically calculate property values and are comprised of an **expression** and a **target property**.
- The expression determines the calculated value and updates the target property. The expression references one or more **source properties**.



## Data and Integration – Validating Case Data

- You use validation rules when you cannot predict or control the value users enter in a form.
- You can validate by using the correct property type when defining your properties
- Most of the fields you add in the **Data Model** tab in either App Studio or the Case Designer are associated with default control types.

## Data and Integration – Validation “Methods”

You can control validation using a variety of techniques:

- Using the correct property type (i.e. **integer** vs. **text**, etc.)
- Configuring the property as required (\*)
- Using controls to limit choices (dropdown, radio buttons, auto-complete, text input set to max characters, etc.)
- Validate rules – used for conditional validation upon form submission – is typically called from a Flow Action rule or another Validate rule (chaining)
- Edit Validate rules – used for pattern matching like SSN, TIN, phone number patterns and more – most often added to a property rule but can also be called from a Validate rule

## Data and Integration – User Controls

- Configuring a control as Always Required (\*) ensures that users enter a value. If there is no value, users receive a standard error '**The field may not be blank**' when they attempt to submit a form.
- You can use editable settings on controls to restrict the input values to valid formats.
- The settings are specific to the control type.

## Data & Integration 15% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=3a96228fdd7b4e89>



# Pega Express Methodology 12%

## **Pega Express Methodology 12%**

- Describe the four phases of Pega Express Delivery and terminology
- Articulate the benefits and best practices of Pega Express Delivery
- Fill the backlog with stories that are ready to be built; user stories
- Describe the process and benefits of DCO
- Document decisions and actions to confirm the technical architecture

# Pega Express as a Delivery Approach

## Definition

Pega express Pega's approach to achieving business outcomes rapidly.

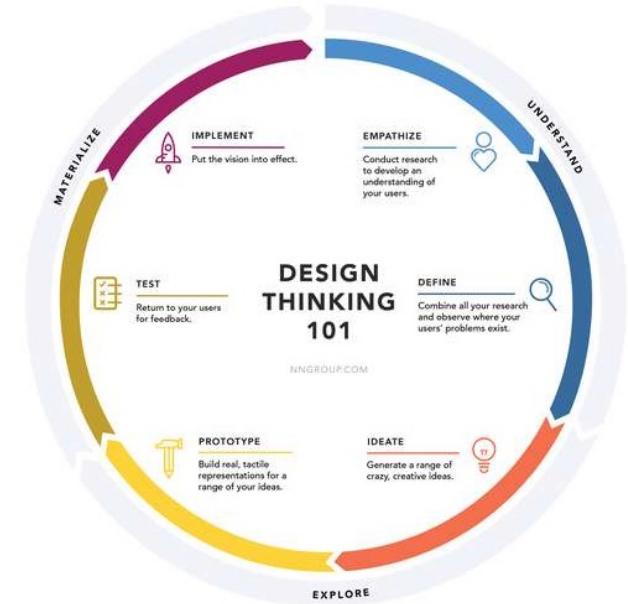
- The approach is a fusion of:
  - Pega best practices for example;
  - Directly Capture Objective (DCO) sessions
  - Concept of the microjourney
  - Strong business and IT collaboration
  - Design thinking
  - Pega's Low code capability and features enabled in the platform.
  - Scrum
- Whether big or small, whether CDH, platform or another Pega product, Pega express is our approach for successful deliveries



# Pega Express Delivery – Design Thinking

## Description

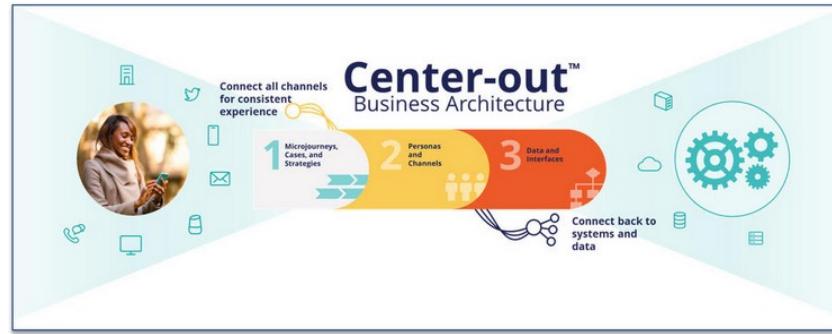
- A Delivery approach that uses design thinking techniques to **break customer journeys** into **smaller**, more manageable pieces called **microjourneys™** that can be designed and deployed in as little as 60-90 days
- Microjourneys are business transactions that are delivered to provide immediate value
- Microjourneys are designed to promote and challenge your project team to deliver meaningful client business outcomes quickly



# Pega Express Delivery – Center Out

## Description

- Pega Express supports the **Center-out approach**
  - Aligns human intelligence and process automation with business logic - beginning at the center of the business
  - Ensures your Pega Platform™ software solution is consistent, seamless, and contextual, providing a great user experience across channels
- A **Center-out business architecture** incorporates the variations within your business
  - variations that extend from your channels to your systems of record
- Pega Platform's layered architecture lets you start small, deploy quick wins, and tackle immediate problems and opportunities
  - You can trust that what you build is scalable as your company grows in scope and size
  - Pega Express reuses microjourneys across the dimensions of customer type, lines of business, or geographies.



# Phases of Pega Express – Four Phases

## Implementation



### Discover

- Use design thinking techniques to clarify the desired outcomes
- Define your Minimum Lovable Product (MLP) and get ready to begin your project



### Prepare

- Design a solution and creates a product backlog using Scrum best practices
- Establish governance and enable your team



### Adopt

- Readies your application for production and prepares the business for transition to "business-as-usual"
- Analyze application performance while planning your next MLP release



### Build

- Iteratively configure and test your application
- Use Scrum project management methods with Pega's low-code platforms and out-of-the-box tools
- Pega Express best practices are built into platforms like App Studio or Next Best Action Designer to define application components and prioritize MLPs quickly

# Pega Express Best Practices

Best practices and tools on which Pega Express is built

- Pega Express™ serves as a best practice delivery approach for your implementation
- The Pega Platform™ incorporates many elements of Pega Express best practices within the tool to deliver business outcomes faster and more consistently

## Business outcomes - value



**Design thinking** is a human-centered approach to solving problems through creativity and collaboration throughout a project, building empathy to satisfy user needs & realize business outcomes to ensure lovable Microjourneys™



**Directly Capture Objectives (DCO)** is the Pega Express discipline used to design solutions with clients directly in Pega to foster ongoing collaboration and feedback between IT and business stakeholders



Pega Express is based on **Agile** and **Scrum** to deliver prioritized business outcomes or **Microjourneys™** in increments known as a **Minimum Lovable Product (MLP)** that provide clients speed to value within 90 days

## Technical outcomes - quality



Pega's **low-code / no-code studios** allow you to create intuitive applications using the 3 pillars: case types, strategies and automations; integrated with live data; and channels, to provide a lovable experience to your customers



Pega's **DevOps Deployment Manager and automated test suite** enables teams to test early, often, and more efficiently than through manual testing alone so that issues can be identified and fixed as early as possible



Pega's **Application Quality dashboard and Predictive Diagnostic Cloud™** automatically monitor the health of your solution and notify you of potential risks so that you can address them earlier, before they become issues

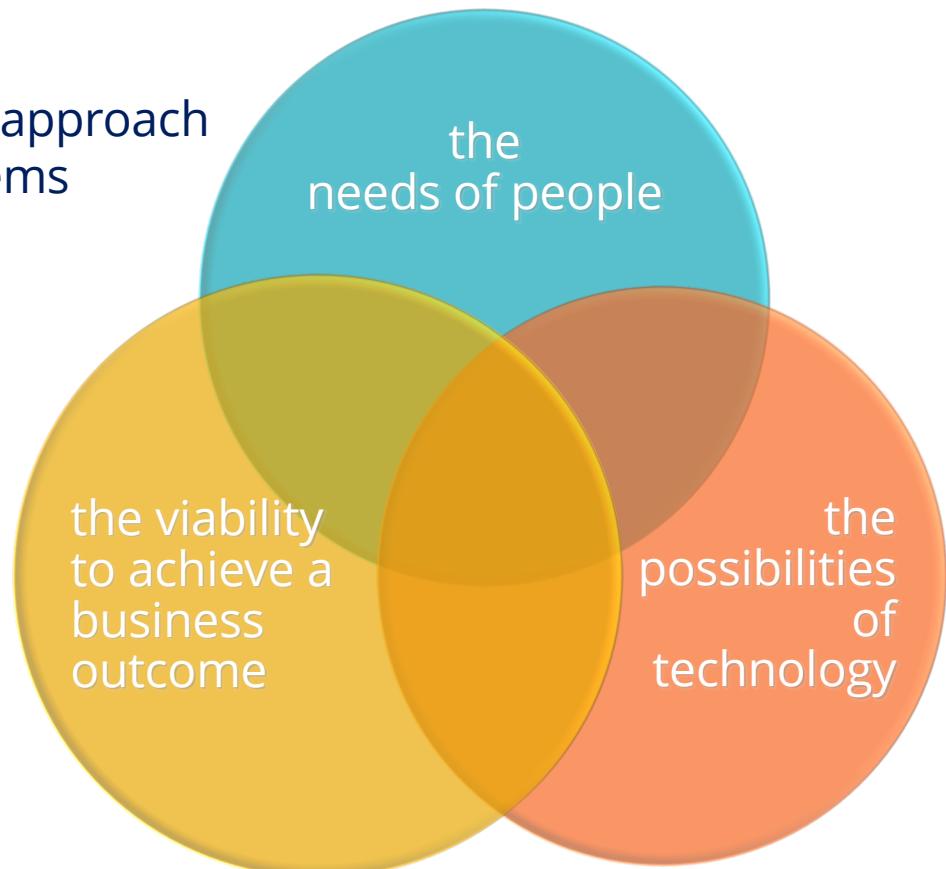
## Use Case

Use **Design Thinking** to bring a group together to successfully generate ideas, push past boundaries, challenge assumptions, and to create alignment on solutions

# Design Thinking

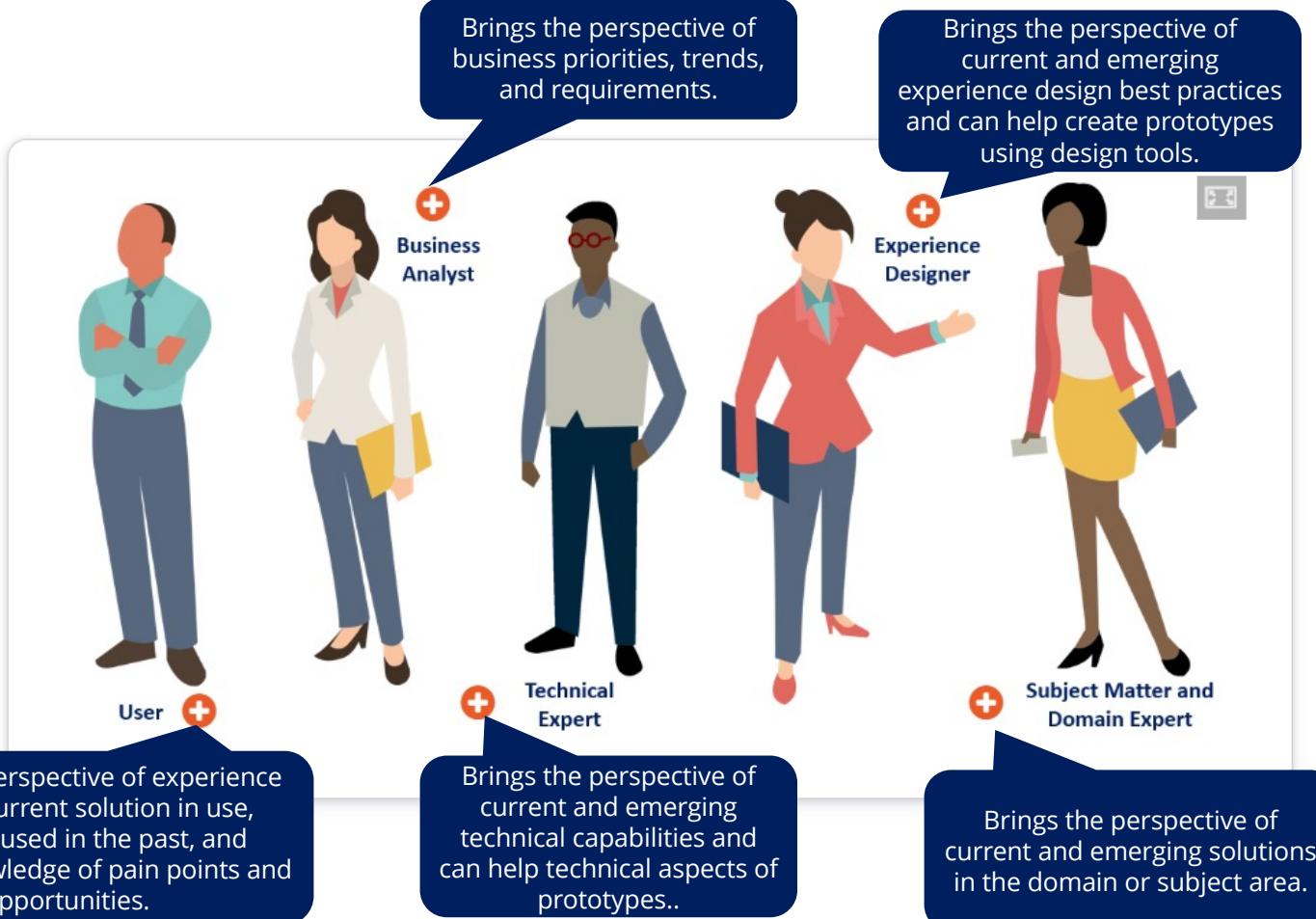
## Definition

- Design thinking is a human-centered approach to identify and creatively solve problems throughout the project lifecycle
- It incorporates:
  - the needs of people
  - the possibilities of technology
  - the viability to achieve a business outcome



# Roles

## Description



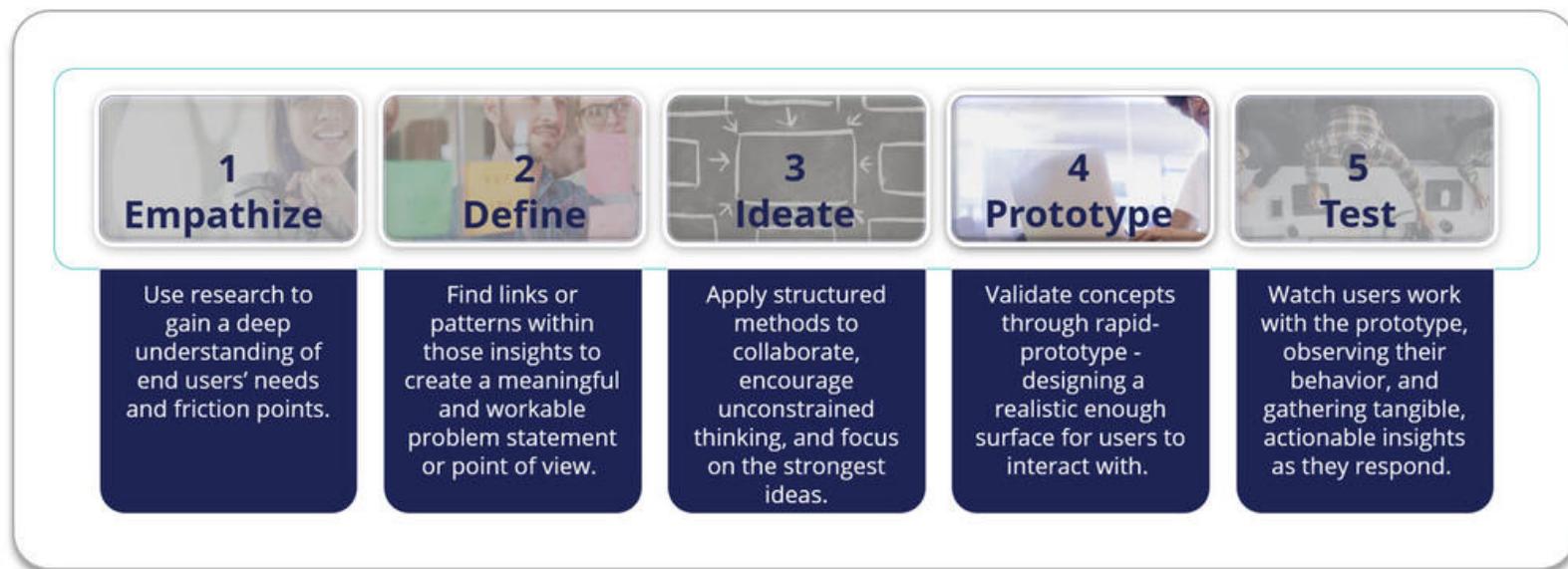
# Roles on a Pega Project

- Business Architect
- System Architect
- System Administrator
- Engagement Leader/Project Manager
- Practice Leader
- Product Owner
- Quality Assurance
- Scrum Master
- SMEs and Specialty Architects

# Design Thinking – Five Steps

## Description

Design thinking brings a group together to successfully generate ideas, push past boundaries, and challenge assumptions



# Design Thinking Tools and Techniques

## Description

- **Design Sprint and Ideation**

- A **Design Sprint** is typically a 5-day workshop that uses Design Thinking techniques to solve a problem using creative methods and innovation.
- It's a way of learning quickly what will and won't work, and it's proven to be very powerful. Amazon, Airbnb, Google make extensive use of Design Sprints
- A **Design Sprint** is a design thinking technique that is proven to help identify the microjourney solution that best achieves the client's business outcomes.

- **Design Sprints** are a great way of bringing both business, IT and users together to design and prototype a solution. A solution that is simple, intuitive and easier to implement.

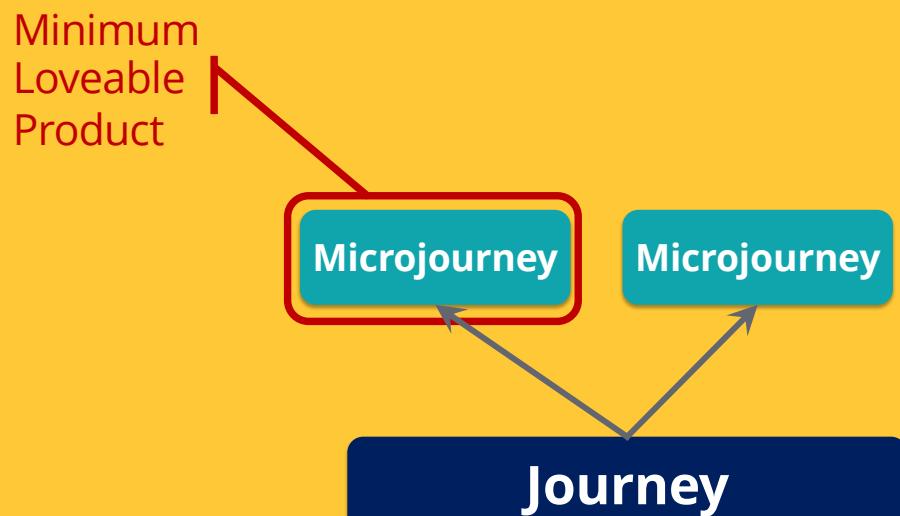
- A **Design Sprint** is a concentrated, collaborative set of design thinking activities that inject a burst of energy into the team, launching the project in the right direction early on.



# Identify Your MLP

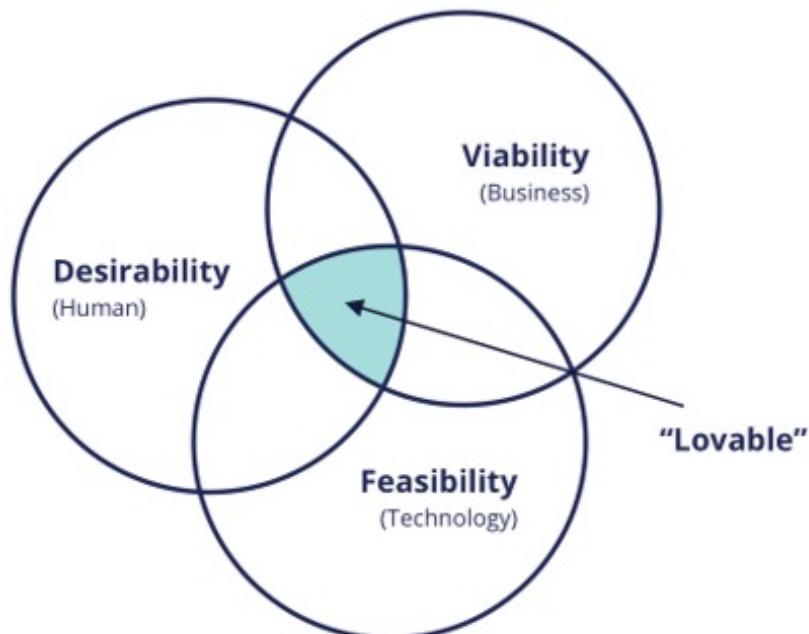
## Definition

- Minimum Loveable Product (MLP)
  - The simplest solution that can be delivered quickly to delight your (or your client's) customers.
- MLPs are composed of one or more Microjourneys supported in the MLP release.



# Minimum Lovable Product

Description



**Design thinking** helps unlock innovative solutions to create a **Lovable** solution by evaluating the following:

## 1. Desirability

Focus is on designing solutions that are desirable for the people that use them

## 2. Viability

Focus is on the business viability of projects for which the sponsors are paying to ensure return on investment

## 3. Feasibility

Focus is on technical feasibility and effort of developing the software

## MLP Priority

Implementation

Higher business value

Medium priority

Lower implementation ease

High priority

Greater implementation ease

No priority

Low priority

Lower business value

# Direct Capture of Objectives

## Definition

- Direct Capture of Objectives (DCO) is a core part of Pega Platform™ and Pega's methodology approach.
- DCO tools are intended to:
  - Facilitate collaboration among project stakeholders
  - Reduce the time between obtaining requirements, design, and implementation
  - Focus on desired business outcomes
  - Speed up and simplify application development

DCO is...



Driven by...

- Collaboration
- Iteration
- Validation

# Application of DCO tools

## Description

DCO-related Tool	Purpose	Benefits
<b>Case Designer</b>	Provides the ability to design the stages and steps of your process or journey. Capture the users and data applicable to each stage, as well as details such as routing and service levels.	Immediately run your process to verify and adjust the case design.
<b>Agile Workbench</b>	Provides the ability to capture and manage features, user stories, defects, and feedback items while working with the application. Integrate Agile Workbench with an agile project management tool such as Pega Agile Studio, or a third-party tool such as Jira, for features such as sprint execution and reporting.	<ul style="list-style-type: none"><li>• Create and edit items while working in the application rather than having to access another system.</li><li>• Capture a screenshot or video to include with the item.</li></ul>
<b>Agile Studio</b>	Provides agile project management functionality, including the ability to create and manage backlogs, plan and execute sprints, collaborate on backlog items, and monitor results with reporting.	You can support and manage Scrum-based development teams.

# Application Profile Estimator

Description

- **The creation of estimates in Pega Platform is automated.** After you provide the required values, the project estimator calculates the expected development duration.
- **Before you begin:**
  - Define the main elements of your application:
    - **Create a case type**, and then define the case life cycle by adding stages, processes, and steps. See [Adding case types to organize work](#).
    - **Create personas** that represent users of your application. See [Adding personas to organize users](#).
    - **Create data objects** that visualize the information that your cases require to reach the resolution stage. See [Adding data objects to organize data](#).
    - **Create features** that represent usable functionalities in your application. See [Creating features](#).



# Real-time, continuous collaboration

## Navigation

A screenshot of the Pega Case Manager interface. At the top, there's a navigation bar with "APP STUDIO", "Case Manager", "Publish", and "Turn editing on". Below that is a search bar and a "Actions" dropdown. The main area shows a case titled "External Training Request (T-15) PENDING-APPROVAL" due "1 day 23 hours from now". The case status is "Request" with a checkmark. A progress bar shows "Review" and "Approved" steps. On the left is a sidebar with icons for "New Case", "Case List", "Case Details", "Assignments", "Attachments", and "Participants". The "Case details" section on the right shows last updated by Dennis Grady (3m ago) and created by Dennis Grady (6m ago). It also lists "Open assignments" (Get Approval via Email), "Attachments (1)" (a PDF file named "External Training Request (T-15) fro..."), and "Participants" (Administrator, Dennis Grady, and another Administrator).

External Training Request (T-15) PENDING-APPROVAL

1 day 23 hours from now

Request ✓ Review Approved

**Approve**  
DUE IN 4 DAYS FROM NOW

Please approve or reject this External Training Request  
Requested by Dennis Grady

**Review**

Name	Start date	Training cost	\$895.00
PegaWorld 2019	6/2/2019	Travel cost estimate	\$2,000.00
Type	End date	Total cost	\$2,895.00
Conference	6/5/2019	Number of days	3
Link			
<a href="https://www1.pega.com/events/pegaworld">https://www1.pega.com/events/pegaworld</a>			

**Business value**

**1. A roadmap for Digital Transformation.**

**2. It's the world's largest Intelligent Automation event.**

**3. Hands-on access you won't get anywhere else.**

**4. Unfiltered insights from top brands.**

**5. Surround yourself with greatness.**

**Notes**

Enter approval or rejection comments

**Case details**

Last updated by Dennis Grady (3m ago)  
Created by Dennis Grady (6m ago)

**Open assignments**

Get Approval via Email (Current)  
Administrator

**Attachments (1)**

External Training Request (T-15) fro... T-15 | Correspondence | De... Jun 01, 2018 02:27 AM

Add Attachment...

**Participants**

A Administrator  
DG Dennis Grady Owner  
A Administrator Manager

Manage

## Pega Express Methodology 12% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=n7n62290be9efdab>



# Reporting 9%

## Reporting 9%

- Creating business reports
- Identify types of reports
- Using columns and filters
- Describe the benefits of using Insights

# Business Metrics

**Business metrics** measure the success or failure of business processes and are based on the data you define for an application. Examples include the number of orders for a certain item, or how many cancellations there are of a certain type of order. Organizations can use these business metrics to make informed decisions about improving business performance.

The following table gives examples of how business metrics can be used in business decisions.

What is the question?	The data indicates	What is the business decision?
What is the average profit margin for all automobile sales last year?	The average margin is below the target percentage.	The sales department decides to train its sales staff on promoting cars and options that have the highest margins.
What is the number of auto loans made in a month as compared to personal loans for the same period?	The number of personal loans is significantly lower than the number of auto loans.	The goal is to have the numbers approximately equal. The marketing department increases marketing resources for personal loans.
What is the number of office desks shipped each week for the past month, and how many are now in inventory?	The number of orders shows an upward trend.	As a result, inventory levels are unacceptably low. The purchasing department decides to restock more desks on a weekly basis.

# Process Metrics

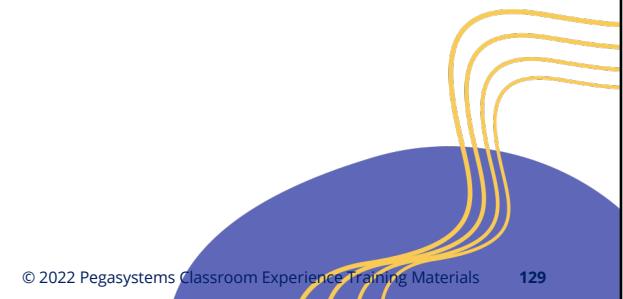
**Process metrics** measure how work is performed and are based on data automatically collected by Pega Platform™. Examples include the time needed to complete an assignment, how often a path is followed in a flow, or the number of Service Level Agreement (SLA) violations. Process metrics enable business analysts and business managers to discover issues that impact process efficiency.

The following table gives examples of how process metrics can be used in process design decisions.

What is the question?	The data indicates	What is the business decision?
Which open loan application cases exceed the standard three-day service level deadline?	Most of the open cases are for loan amounts greater than USD 300,000.	Loan requests that exceed this amount must go through an additional review step, which accounts for the delay. The department manager decides to increase the service level deadline for loans exceeding 300,000 USD from 3 to 4 days.
What is the average duration of assignments by type and action?	This report identifies which user actions take the longest to complete.	Spend time on improving the efficiency of those assignments taking the most amount of time to complete.

## Reporting – Report Definition

- Rule used to query information from external database and data tables
- Automatically generates SQL
- Number of different configuration options available



# Reporting – Report Definitions

Use a report definition to identify the data to retrieve from records from a database.

- Results are organized as a table of columns and rows.
- The rows represent records retrieved from the database.
- The columns contain the data values in each record that you want users to see.

Case ID	Employee	Hire Date	Location
O-101	James Martin	2/21/16	Atlanta
O-104	Anne Walker	2/4/16	Boston
O-746	Julia Phagan	1/12/16	Atlanta
O-983	William Kirk	9/8/16	Atlanta
O-171	Leonard Kelley	9/5/16	Boston
O-623	Kate Picardo	2/25/16	Atlanta
O-421	Robert Wang	2/1/16	Boston

## Reporting – Creating Report Definitions

- Can be created by using the Report Browser and selecting:
  - New Report button (Pega Express)
  - Add Report button (Case Manager Portal)
  - Selecting a report in the Report Browser, modifying it, choosing “Save As” and placing the newly created report in the appropriate category
- Many different report categories available out of the box
  - Monitor Assignments, Analyze Performance, Analyze Quality and Monitor Processes, etc.
  - You can create your own categories for both public and personal reports
- Can be created by creating a Report Definition rule

## Reporting – Report Definition

The Report Definition rules allows you to create two different types of reports

- List Reports
  - Spreadsheet style reports that can combine business and process metrics
- Summary or Chart Reports
  - For aggregating data with counts, totals, averages
  - Often used to show trends over time
  - May include a chart or graph to display summarized data
    - Available charts include – line, bar, column, pie, bubble and gauges to name a few

## Reporting - Filters

Filters allow you to only show records that are relevant to your design requirement.

- A filter compares a data value in the record against a defined condition.
- If the comparison result is true, the report includes the record.
- If the comparison fails the filter tests, the record is not included.

Case ID	Employee	Hire Date	Location
O-101	James Martin	2/21/16	Atlanta
O-104	Anne Walker	O-104	Anne Walker
O-746	Julia Phagan	1/12/16	Atlanta
O-983	William Kirk	9/8/16	Atlanta
O-171	Leonard Kelley	O-171	Leonard Kelley
O-623	Kate Picardo	2/25/16	Atlanta
O-421	Robert Wang	O-421	Robert Wang

## Reporting - Filters

- The comparison can be an explicit value or the value of a property.
- You can also use more complex conditions such as testing values that are greater than a specified threshold, like a date.
- You can implement more complex filtering by combining filters with AND/OR conditions.

## Reporting - Filters

Rather than bringing all the data back to Pega 8, make your report more performant by adding filters.

Report Definition: Get HR Plans List (Available )  
SAE-HRServices-Data-HRPlan • GetHRPlansList | HRServices:01-01-01

Private edit Save as Actions ▾ Close

Query Chart Report Viewer Data Access Parameters Pages & Classes History

EDIT COLUMNS

COLUMN SOURCE	COLUMN NAME	SUMMARIZE	SORT TYPE	SORT ORDER	⋮
.Name	Name	<blank>	▼	<blank>	⋮
.Type	Type	<blank>	▼	<blank>	⋮
.Description	Description	<blank>	▼	<blank>	⋮
.Id	Id	<blank>	▼	<blank>	⋮
.EmployeeCost	EmployeeCost	<blank>	▼	<blank>	⋮

Add column

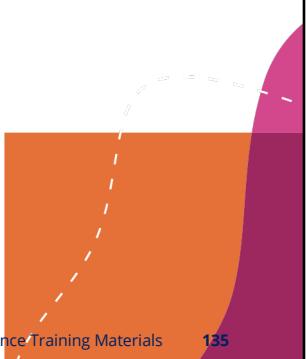
EDIT FILTERS

Filter conditions

A

CONDITION CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE	⋮
A	.Type	Is Equal	Param.Type	Select values

Add filter

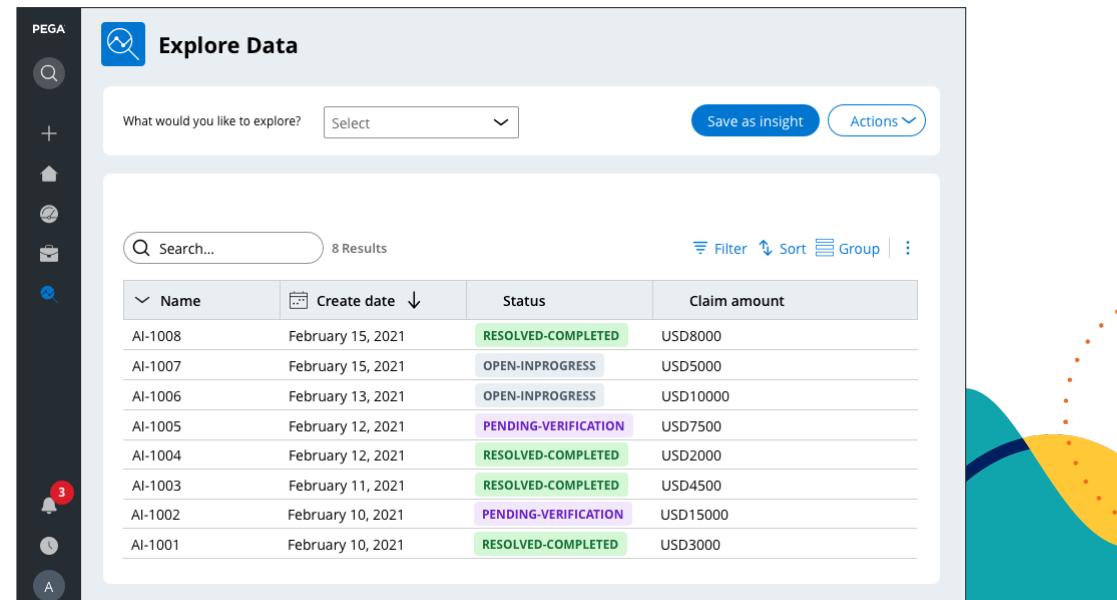


## Data Explorer and Insights

- Explore Data uses a React-based landing page and is the preferred data exploration tool for applications that use the Pega Cosmos design system.
- With the Explore Data landing page, you can access the list of assignments that are routed to members of your team. You can then drill down and analyze the assignment statuses to determine the remaining workload for the current release. You can save queries you make on the Explore Data landing page as **insights**.
- **Insights** are rules that Pega Platform™ uses to transform data queries into tables or visualizations that you can then share between users.
- You can use insights to retrieve specific data and present the data as a list or an interactive chart.

# Insights

- **Insights** are rules that Pega Platform™ uses to transform data queries into tables or visualizations that you can then share between users.
- You can use insights to retrieve specific data and present the data as a list or an interactive chart.
- Create or modify Insights:
  - Save an insight
  - Perform an action on an insight
  - Access application data
  - View data based on business needs
  - Customize columns



The screenshot shows the Pega Explore Data interface. On the left is a dark sidebar with icons for search, plus, home, refresh, and a bell with a red notification badge containing the number '3'. The main area has a header 'Explore Data' with a magnifying glass icon. Below it is a dropdown menu 'What would you like to explore?' with 'Select' and 'Actions' buttons. A search bar 'Search...' shows '8 Results'. To the right are buttons for 'Filter', 'Sort', 'Group', and a three-dot menu. A table lists eight claims with columns: Name, Create date, Status, and Claim amount. The data is as follows:

Name	Create date	Status	Claim amount
AI-1008	February 15, 2021	RESOLVED-COMPLETED	USD8000
AI-1007	February 15, 2021	OPEN-INPROGRESS	USD5000
AI-1006	February 13, 2021	OPEN-INPROGRESS	USD10000
AI-1005	February 12, 2021	PENDING-VERIFICATION	USD7500
AI-1004	February 12, 2021	RESOLVED-COMPLETED	USD2000
AI-1003	February 11, 2021	RESOLVED-COMPLETED	USD4500
AI-1002	February 10, 2021	PENDING-VERIFICATION	USD15000
AI-1001	February 10, 2021	RESOLVED-COMPLETED	USD3000

## Reports 9% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=akm622911b23df5d>



# User Experience 9%

## User Experience 9%

- Customizing user interface element
  - Dashboards
  - Portal Content
- Configure Action Sets

# User Interface Concepts

## Description

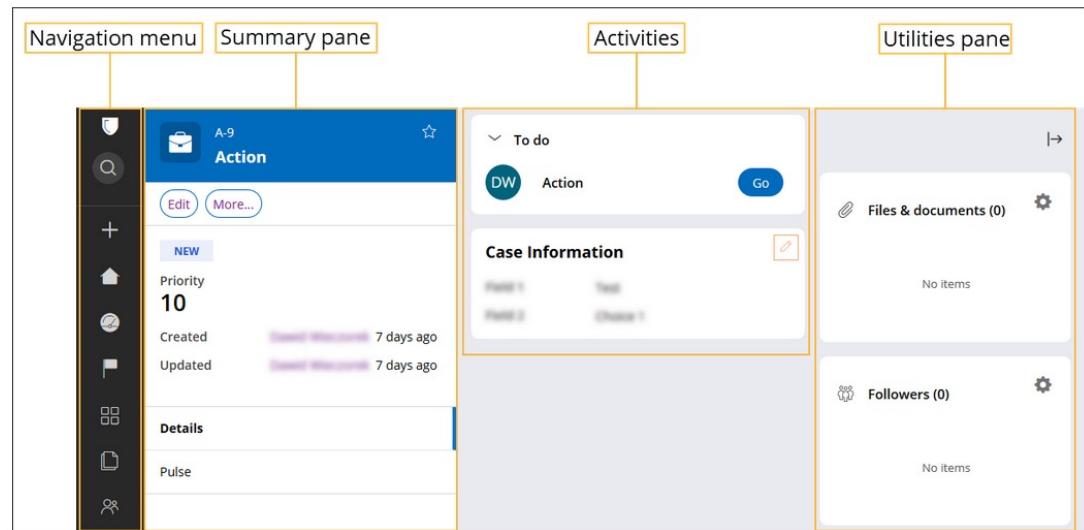
- The application presents a User Interface or View to the user to interact with the application.
  - Entering values into fields
  - Selecting a value from a dropdown
  - Clicking on a control, a link or image
- Careful consideration of design is important.
  - How will the user interact with the UI?
- Presentation consistency
  - What devices will be used by the user?
  - Are dynamic features required
  - Are Responsive UI features required
- Users of Pega applications can interact with the application through a web browser, tablet or mobile application.

The image displays two side-by-side screenshots of a Pega application interface. The left screenshot shows a desktop browser window with a multi-step form titled '1. Customer Details', '2. Collect Payment Info', and '3. Vehicle Details'. It contains several input fields for bank details, account type, and amount. The right screenshot shows a mobile application interface on an iPhone, titled 'C-2', with sections for 'Collect Personal Details' (including 'Selected position' set to 'Marketing Manager') and 'Candidate information' (with fields for First Name and Last Name). Both interfaces feature a bottom navigation bar with buttons for 'Cancel', 'Save', and 'Submit'.

# Pega Cosmos Design System

## Description

- A complete UX toolkit designed for large organizations' needs with case management application use cases
- UI Redesign
  - **Navigation menu** - allows for easy access to the main pages
  - **Summary pane** – for Case data and related objects, which has a customizable interface
  - **Work area** - features activity and life-cycle tasks, such as ad-hoc, suggested, or completed tasks
  - **Utilities pane** – expandable, displays widgets for participants, attachments, and tags



# Cosmos Design System

## Description

- Provides a library of reusable UI components based on 35+ years of business application experience.
- Intuitively presents data and actions to business users.
- Extensible by front-end developers: Publish new components by using the open-source React UI.
- Ensure that your application includes a modern, responsive, and consistent UI by using the out-of-the-box Theme-Cosmos in your application stack

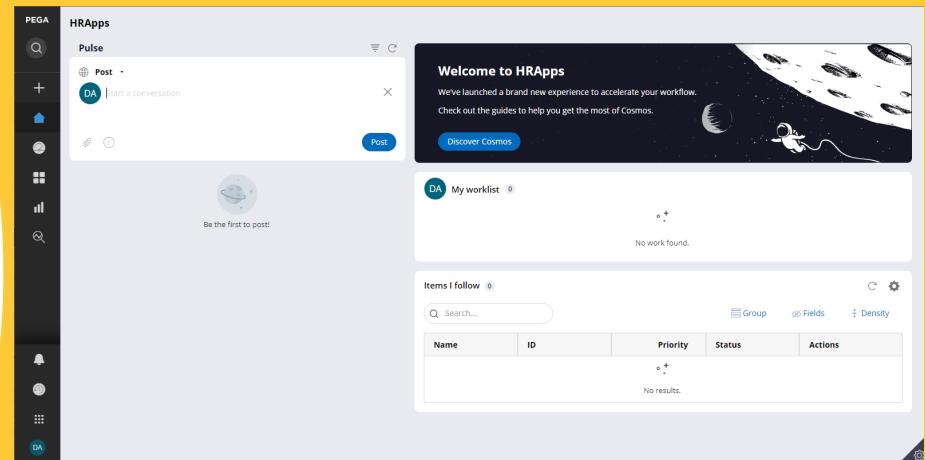
Name	Version
1 Theme-Cosmos	03.01

Name	Version
1 PegaRULES	8

# Portals

## Definition

- Users experience the portal through a browser, regardless of device type.
- Easily configured by choosing from predefined portal templates that define screen layout and required features.
- **User** portal provides a standard user interface for working on cases. It is intended to be used by end users on desktop and mobile devices.



# Dashboard

- Dashboards are customizable by end users and are made up of widgets that consolidate summary information.
- Displays operational information about an application and key performance indicators from different sources.

The screenshot shows a Pega dashboard interface. On the left is a sidebar with a '+ New' button and a list of links: Dashboard, My Work, Pulse, Spaces, Documents, My Teams, Reports, Tags, Following, Recents (with entries for Case C-05 and Case C-04), and a PEGA logo at the bottom.

The main area is titled 'Summary for Your Application'. It contains several widgets:

- Case stages for case type:** A process flow diagram showing three stages connected by arrows, each labeled 'Process' below it.
- Operator:** A summary card showing 5 OPEN cases and 2 OVERDUE cases.
- Cases by status:** A donut chart divided into three segments: Stage 1 (blue), Stage 2 (orange), and Stage 3 (yellow).
- Team members:** A list of roles with corresponding icons: Operator (green circle), Administrator (purple circle), and User (red circle).

Name	Case	Category
-2d Assignment name	C-05	Case name
-1d Assignment name	C-04	Case name
-- Assignment name	C-03	Case name
-- Assignment name	C-02	Case name
-- Assignment name	C-01	Case name

# Pages

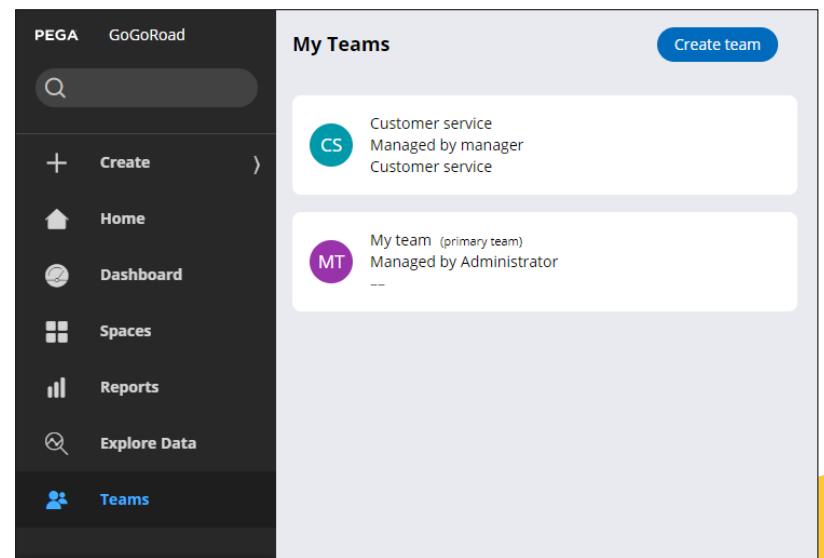
## Definition

- **Page Menu**

- Create, configure, and add pages to menus within an application to display specific information.
- Added to a portal and automatically added to the portal's navigation menu.
- Managing the list of pages, customizes the primary navigation menu, improving navigation and user experience.

- **Page Types**

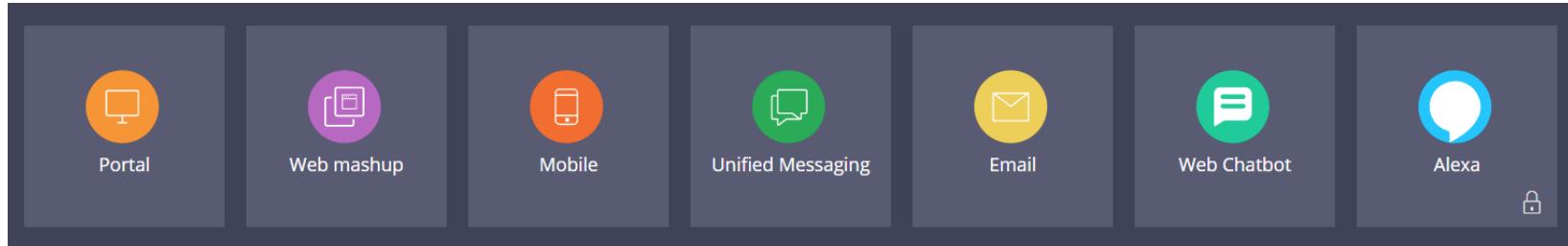
- **Landing Pages** are created to build an application that matches the needs of your users and consists of fields, control and resources presented as images and text.
- **Custom pages** are Pega's out-of-the-box landing pages that are built as part of the application and are not customizable by end users.



# Channel

## Definition

- A messaging service, voice service, web portal or mobile portal.
- Customers interact with organizations through a variety of channels.
- Created from templates that include predefined layouts and navigation for use in an application.
- Provide ways for users to interact with your application by using **Pega Intelligent Virtual Assistant™** and **Pega Email Bot**.
- The **Channels** landing page allows you to create, view, and edit all types of channel interfaces.
- The following Channels can be added:

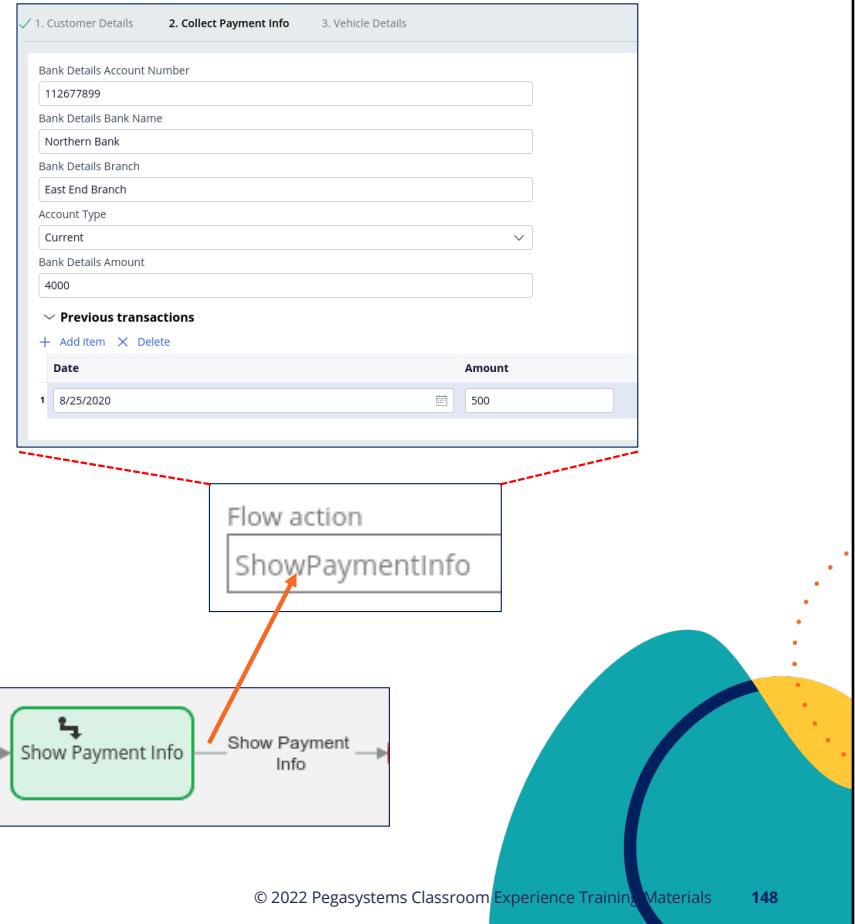


- New components can be downloaded from Pega Marketplace.
- The Lock icon on a channel indicates that the channel component is not yet added to the application.

# View – User Interface

## Definition

- A **View** is associated with a collect information step (assignment shape) in the flow rule of a case
- A **Flow Action** is associated with the Connector leaving the shape in the Main or Sub flow
- A **Flow Action** is associated with the Assignment shape in a Multi-Step Form
- A **View** may be constructed using Case Designer or by configuring the rules directly



# Design Templates

## Navigation

Dev Studio > App Explorer > User Interface > Section > open section > View Editor > Template

The screenshot shows the Pega Dev Studio interface with the following details:

- Left Sidebar:** Shows the application structure under "WIND-Auto-Work". The "CustomerDetails" section is selected.
- Header:** Application: BranchDev, Configure, Launch portal, Create, Search, Save as, Delete, Actions, Check out.
- Section Header:** Section: Customer Details [Available] CL: WIND-Auto-Work-EvaluateAndSellAVehicle ID: CustomerDetails RS: BranchDev [Branch: BranchDevLab]
- Message:** This section is configured using the View editor. To access full design capabilities convert to full section editor.
- Editor Tabs:** Design (selected), Settings, Parameters, Pages & Classes, Specifications, History.
- Design Area:** Displays a form with fields: Name ( lorem ipsum ), Phone number ( (123)-456-7890 ), Date of Birth ( 11/10/2021 ). A placeholder "Address" section is present with the label "Section - CustomerAddress".
- Template Editor:** A modal window titled "Template" shows a 2-column layout with sections A and B. It lists fields: Name (Text Input), Phone number (Phone), Date of Birth (Date time), and CustomerAddress (Section). A "Change" button is available to switch to a full section editor.

# Design Templates

## Description

Each view can contain one design template. You can combine smaller, modular views into one larger view to mimic the use of multiple templates in a single view.

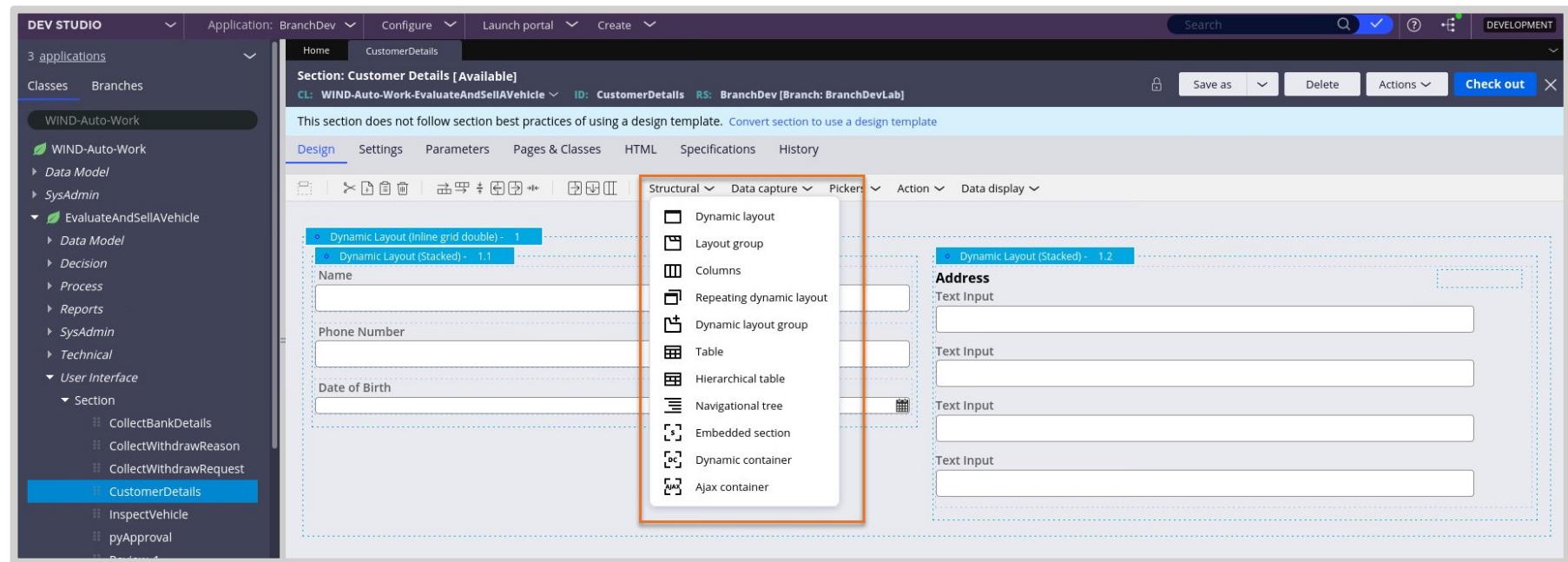
This screenshot shows a Pega view window with two distinct design templates. On the left, under 'Customer information', there are seven input fields: First name, Last Name, Street Address, City, Postal code, State or Province, and Telephone number. On the right, under 'Invoice address', there are four input fields: Address line 1, Address line 2, and City, State/Province, Postal Code. Each template is enclosed in a dashed green border.

This screenshot shows a Pega view window where the two design templates from the previous screenshot have been combined into a single, larger template. The 'Customer information' section is now split into two columns: 'Column A' (First name, Last Name, Street Address, City, Postal code) and 'Column B' (State or Province, Telephone number). The 'Invoice address' section remains on the right. The entire combined section is enclosed in a dashed green border.

# Layouts

## Navigation

**Dev Studio > App Explorer > User Interface > Section > open section > Full Section Editor > Structural**



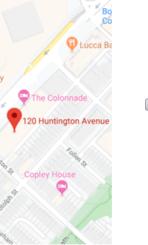
# Repeating Layouts

## Description

There are different types of repeating layouts that support different use cases:

- A **table layout** is useful when you want to present tabular data in a series of columns and rows. Generally, you do not want to use a table layout to present images.

**Booking**

Listing name	Address	Price per night	Maximum number of guests	Description	Map	Book this listing!
Cozy Room	120 Huntington Ave, Boston, MA	\$100.00	2	A unique approach to hospitality. We pride ourselves on offering a Back Bay Boston hotel experience for each of our guests.		<button>Done</button>
Downtown Condo	965-A, Boylston St, Boston, MA	\$200.00	4	Back Bay address offers premier access to the Boston		<button>Done</button>

- A **repeating dynamic layout** is useful when you want to group and present content in a nonlinear, more aesthetic format.

**Booking**

**Cozy Room**  
120 Huntington Ave, Boston, MA  
Price per night \$100.00  
Maximum number of guests 2

A unique approach to hospitality. We pride ourselves on offering a Back Bay Boston hotel experience for each of our guests.



**Downtown Condo**  
965-A, Boylston St, Boston, MA  
Price per night \$200.00  
Maximum number of guests 4

Back Bay address offers premier access to the Boston Symphony, and the Museum of Fine Arts Boston. Transit is nearby.



Done

## Dynamic UI - Use Case

Dynamic UI leads to a more compelling, modern user experience.

- Real-time response to end-user behavior
- Robust functionality available for most user interactions
- Reduced visual clutter on the screen
- Fewer full-page refreshes, resulting in improved UI responsiveness
- When need to capture more details based on customer's income for loan process
- When the age is a factor of deciding an eligibility for an application



# Showing and Hiding Elements

## Description

Use a **Visible When** condition to control the display of additional fields.

- Controlling the fields displayed can simplify the UI.
- Visible When conditions contain logic that toggles the display of a data.

The user selects **Single** as the marital status in a form. No additional input is required.

- When the user selects **Married**, the fields Name of Spouse and Date of Marriage are displayed.

Visibility	Condition (expression)	.MaritalStatus = 'Married'	
------------	------------------------	----------------------------	--

Name	Marital Status
Linda Brown	Single ▾

Name	Marital Status	Date Of Marriage	Name Of Spouse
Sarah Jones	Married ▾	4/30/2010	Michael Jones

# Control Field Display

## Description

- Pega Platform provides the ability to control how fields are displayed. Application Developers create field attributes for dynamic display.
- You can add conditions to UI elements that affect visibility and user interaction.

Setting	Definition	Attributes
Always	The UI element is always displayed, disabled or required	Visibility, Disable, Required
Condition(Expression)	Uses a Boolean expression to determine visibility, visible when the expression returns true	Visibility, Disable, Required
Condition(when rule)	Uses a when rule to determine visibility, visible when the expression returns true	Visibility, Disable, Required
If not blank	Visible if the value of that field is not blank	Visibility
If not zero	Visible if the value of that field is a non-zero number	Visibility
Never	The UI element is never disabled or required	Disable, Required

# Action Sets

## Definition

- Dynamic UI behavior is implemented with an event-action model.
- An action set is comprised of an event, an action, and (optionally) conditions.
  - **Event** - a trigger performed by the user such as click, double-click, hover, focus, and keyboard entry
  - **Action** – a response performed by the system as a result of the user event. For example, when the user clicks a button, a case is created
  - **Conditions** - restrictions such as when rules, which can be applied to an event and action combination

<b>Event</b> = something happens	<b>Action</b> = something changes
Property-based event  Order Total: <input type="text" value="£501.98"/>	Display message "Purchase orders limited to £500"
User action event  Marital Status: <input type="radio"/> Single <input checked="" type="radio"/> Married	Display partner information section

# Action Sets Configured

Description

- 1. Identifier** – Identifies the action set.
  - Action sets are evaluated in numerical order, starting with Action Set 1.
- 2. Applicability** – Determines when the action set is applied.
  - Action sets can be configured for use only when a field is in an editable state, read-only state, or both states.
- 3. Event** – separate slide
- 4. Action** - separate slide
- 5. Conditions** - Any conditions that must be met before the action occurs.
  - Multiple conditions can be combined using either a boolean AND or OR operator.

The screenshot shows the Pega Studio interface for configuring an Action Set. The main panel is titled "Action Set 1".

- Applicability:** Set to "Editable".
- Add an event:** A button labeled "Change" with a trash icon.
- Add an action:** A dropdown menu showing "Set value".
- Property:** A list of BillingAddress fields: Street, City, State, and PostalCode.
- Value:** A list of ShippingAddress fields corresponding to the BillingAddress properties.
- Conditions:** A section labeled "5" containing a "When" condition. The condition is set to "Other property" and specifies ".CopyAddress" equals "true".

# Action Sets - Event

## Description

Events are the trigger for the action set.

- **Mouse events:** events triggered by one of the following actions performed with a mouse or other pointing device
- **Keyboard events:** events triggered by pressing one of the following keys on the keyboard
- **Other events:** events that do not fall into another category
  - When selecting an event, verify that the selected event is appropriate for both the control and the intended device.

Mouse events	Keyboard events	Other events
<a href="#">Click</a>	<a href="#">Enter key</a>	<a href="#">Focus</a>
<a href="#">Double-click</a>	<a href="#">Up key</a>	
<a href="#">Hover</a>	<a href="#">Down key</a>	
<a href="#">Right-click</a>	<a href="#">Left key</a>	
	<a href="#">Right key</a>	
	<a href="#">Esc key</a>	
	<a href="#">Tab key</a>	
	<a href="#">Any key</a>	

# Action Sets - Action

## Description

- The action or actions performed when the specified event occurs.
- An action may require that you provide values for needed parameters.
- If an action set lists multiple actions, actions occur in the order listed, starting with the topmost action in the action set.
- Any implicit action triggered by a listed action is completed before the system moves on to the next action.

All actions ( <a href="#">Common actions</a> )			
Display	Process Work	Launch	List
Apply conditions	Add new work	Flow in modal dialog	Add child
Close	Cancel	Get directions	Add item
Expand/Collapse	Contents	Harness	Delete item
Home	Enable action section	Mobile View	Edit item
Menu	Explore	Landing page	Open local action
Mobile search	Finish assignment	Local action	Open selected item
Post value	Perform action	Open URL in window	Refresh current row
Print	Review	Open mobile app	Refresh list
Refresh	Save	Report definition	Set focus
Refresh row	Show flow location	Scan Barcode/QR Code	Other
Section	Show reopen screen	Wizard	Invoke action
Set focus	Update	Get work	Log off
Set style	View Attachments	Create work	Manage guided tour
Set value	View history	Get next work	Open rule
Show smart info		Open assignment	Open rule by keys
Show smart tip		Open work by handle	Open rule by name
Spell check		Open work item	Run activity
		Re-open work item	Run data transform
			Run script

## User Experience 9% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=cma62291f679e132>



# Application Development 9%

## **Application Development 9%**

- Manage application development: user stories, feedback, bugs
- Use the Estimator to scope a Pega Platform project

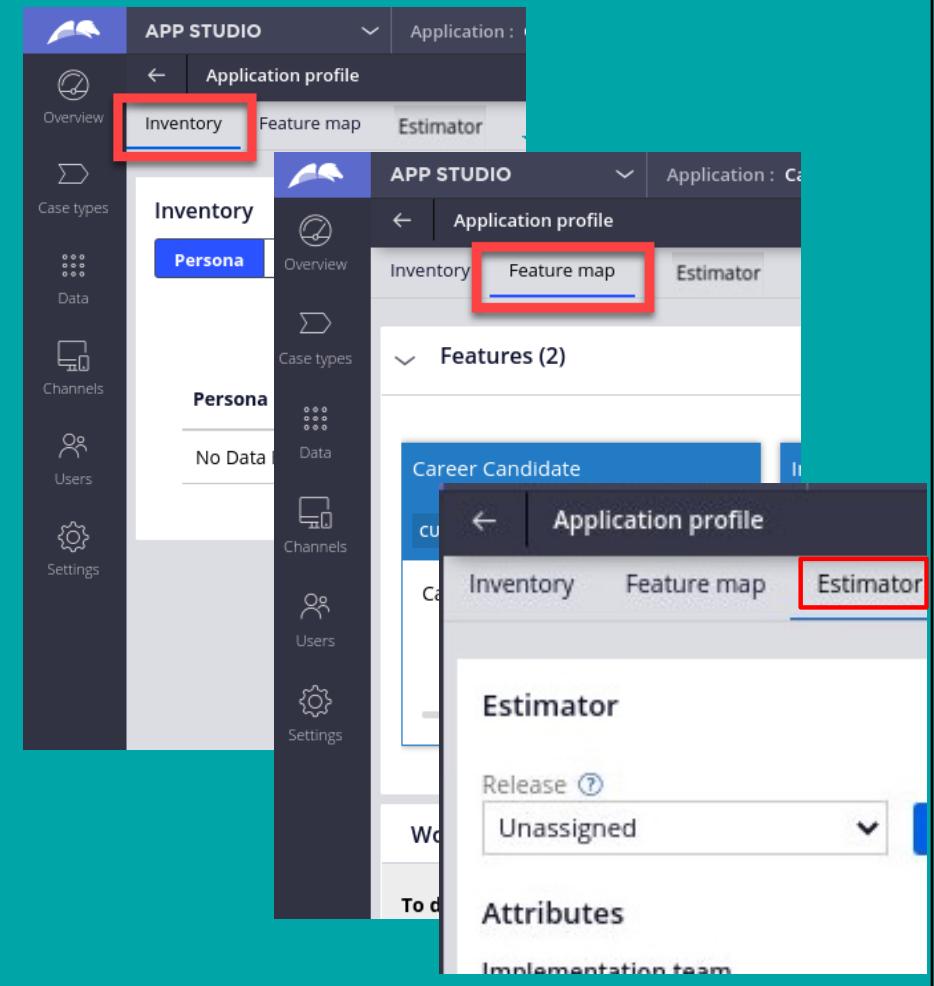
# Application Profile

## Description

The Application Profile landing page tabs:

- **Inventory** - Persona and Data relationships
- **Feature Map** – Feature/subfeature, and work item relationships
- **Estimator** - greater accuracy and efficiency to estimate projects, more intuitive and automated

Improves communication with interested parties and provides a universal understanding of requirements.



# Inventory

## Description

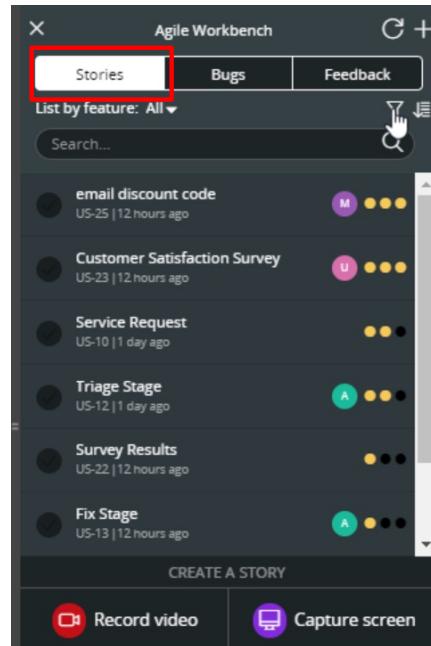
- The Inventory landing page is a tool for implementing the Pega Express methodology in delivering your projects in a goal-oriented and no-code way.
- With the Inventory page, you can view lists of all personas and data objects in your application.
- You can quickly access the information that you need by grouping items in the list by different criteria, such as by persona or case type.

Inventory						
Persona		Data				
▼ Persona	Channel	Case type	Stage	Release	Status	
▼ Channel : Connect on the move App						Total 5
Approver	Connect on the move App	Hire	Decision	MLP 1	TO DO	:
Hiring Manager	Connect on the move App	Hire	Decision	MLP 1	TO DO	:
Hiring Manager	Connect on the move App	Hire	Decision	MLP 2	TO DO	:
Staffing consultant	Connect on the move App	Hire	Interview	MLP 2	TO DO	:
Staffing consultant	Connect on the move App	Hire	Offer	MLP 1	TO DO	:
▼ Channel : Email for job						Total 3
Applicant	Email for job	Hire	Collect resume	MLP 1	TO DO	:
Employees	Email for job	Hire	Collect resume	MLP 1	TO DO	:
Staffing consultant	Email for job	Hire	Offer		TO DO	:

# Add Stories

## Description

Team members may add user stories to the backlog and refine existing items for each application release.



The Pega App Studio interface shows the creation of a new story. The 'Case type: Interview' is selected. On the left, the 'Case life cycle' is defined with steps: 1. Interview (selected), 2. Feedback, 3. Approve Schedule, 4. Conduct Interview. Step 1 is highlighted with a red box and has a blue trash icon. A red arrow points from this step to the 'Feedback' step on the right. The 'Workflow' tab is selected in the top navigation.

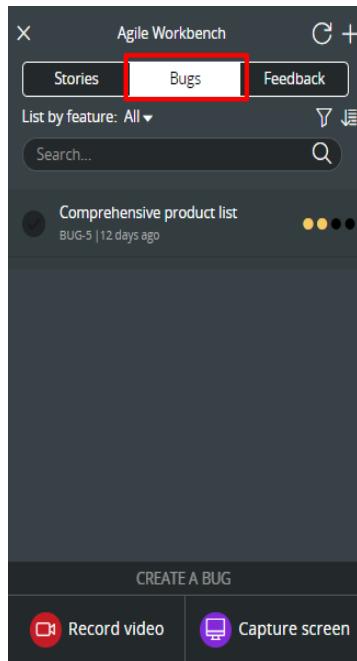
**New Story** details:

- Name: Schedule Interview
- Story ID: US-1
- Description: I, the human resources associate (HRA) need to be able to assign an Interview date/time in the scheduling system.
- Associated feature: Interview
- Owner: Admin@Careers
- Due: 7/1/2021
- Complexity: High
- Priority: Must have
- Attachments: Add attachment
- Acceptance criteria: (empty)
- Additional details: (empty)
- Pulse: Post, Start a conversation

Buttons at the bottom: Cancel, Mark as..., Save.

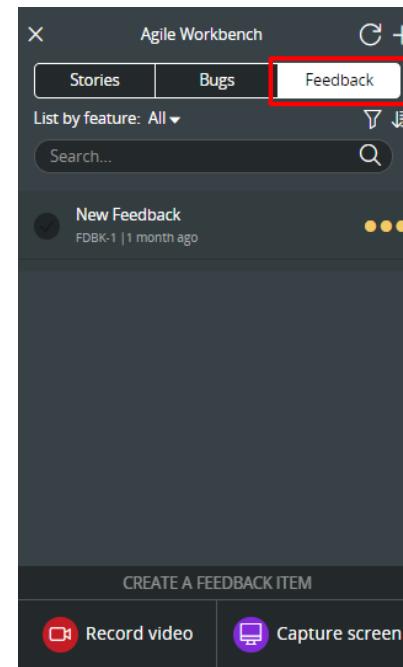
## Record Bugs

As you notice bugs and recognize changes to make, you record those bugs, feedback, and enhancements (user stories) in Agile Workbench.



## Capture Feedback

Capture feedback from a business stakeholder in real-time using Agile Workbench in Pega Platform.



# Estimator

## Description

- A project estimate considers various factors:
  - complexity of the application elements
  - # of teams
  - development environment exists
- The creation of estimates in Pega Platform is automated.
- Estimation results maybe exported to .xlsx file.

The screenshot shows the Pega Platform's Estimator interface. On the left, there are several input fields under sections like 'Attributes' (Release: All, Delivery: Scrum/Agile, Number of teams: 1, Staffing model: Co-production), 'Organization' (Environment: Pega cloud, Organization complexity: Low, Data import effort: Low), and 'Items by complexity' (Complex: 1, High: 1, Medium: 5, Low: 10, OOTB: 1). In the center, it displays 'Estimated size for all releases' with a total of 810 - 1,170 Total hours, a duration of 6 - 8 weeks, and hours of 347 - 502 Pega or Partner hours, 463 - 668 Client hours. On the right, it shows 'Estimated size for all releases' with 18 inventory items, with tables for 'Items by complexity' and 'Items by type'.

Complexity	Count
Complex	1
High	1
Medium	5
Low	10
OOTB	1

Type	Count
Case types	1
Personas	5
Channels	3
Data objects	5
Attachments	2
Features	4

# Application Profile Estimator

Description

- **The creation of estimates in Pega Platform is automated.** After you provide the required values, the project estimator calculates the expected development duration.
- **Before you begin:**
  - Define the main elements of your application:
    - **Create a case type**, and then define the case life cycle by adding stages, processes, and steps. See [Adding case types to organize work](#).
    - **Create personas** that represent users of your application. See [Adding personas to organize users](#).
    - **Create data objects** that visualize the information that your cases require to reach the resolution stage. See [Adding data objects to organize data](#).
    - **Create features** that represent usable functionalities in your application. See [Creating features](#).

# Direct Capture of Objectives

## Definition

- Direct Capture of Objectives (DCO) is a core part of Pega Platform™ and Pega's methodology approach.
- DCO tools are intended to:
  - Facilitate collaboration among project stakeholders
  - Reduce the time between obtaining requirements, design, and implementation
  - Focus on desired business outcomes
  - Speed up and simplify application development

DCO is...



Driven by...

- Collaboration
- Iteration
- Validation

# Application of DCO tools

## Description

DCO-related Tool	Purpose	Benefits
<b>Case Designer</b>	Provides the ability to design the stages and steps of your process or journey. Capture the users and data applicable to each stage, as well as details such as routing and service levels.	Immediately run your process to verify and adjust the case design.
<b>Agile Workbench</b>	Provides the ability to capture and manage features, user stories, defects, and feedback items while working with the application. Integrate Agile Workbench with an agile project management tool such as Pega Agile Studio, or a third-party tool such as Jira, for features such as sprint execution and reporting.	<ul style="list-style-type: none"><li>• Create and edit items while working in the application rather than having to access another system.</li><li>• Capture a screenshot or video to include with the item.</li></ul>
<b>Agile Studio</b>	Provides agile project management functionality, including the ability to create and manage backlogs, plan and execute sprints, collaborate on backlog items, and monitor results with reporting.	You can support and manage Scrum-based development teams.

## Application Development 9% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=qyh62297397ae5f6>



Mobility 6% &  
Security 2%

## Mobility 6%

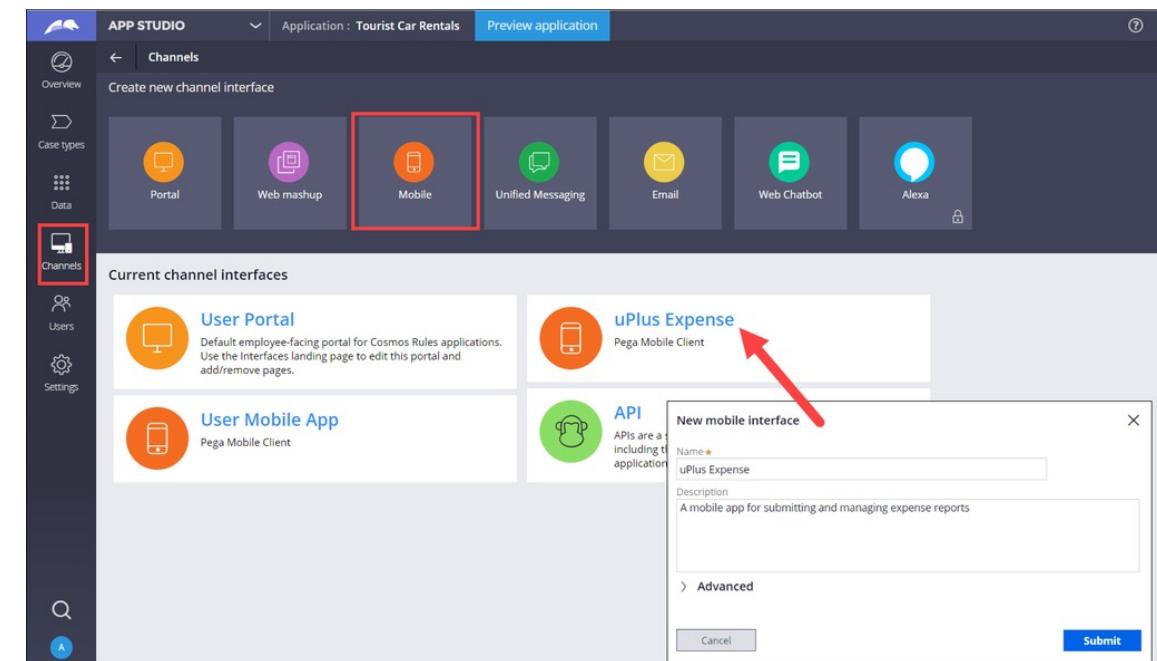
- Configure mobile app channels
- Leverage Pega Mobile Preview

## Security 2%

- Manage user and role assignments

# Mobility – Mobile app channels

- Mobile channels have become more powerful and easy to use providing an additional, convenient access channel to support employees in the field. You can use the existing mobile channel to quickly build a consumer-grade app or modify an app to meet your needs.
- A mobile channel provides developers a way to intuitively configure and customize various aspects of mobile app behavior.



<https://academy.pega.com/topic/mobile-app-channels/v1/in/26826/24056/23811>

# Mobility – Mobile app security

- To make your mobile app more secure, you can use the Security tab to select an authentication model and configure the locking settings for your app. You can also disable authentication requirements for users who want to access the app by selecting a role that determines which users have access to the mobile channel.

The screenshot shows the Pega Platform Configuration interface. On the left, a sidebar lists 'Content', 'Configuration' (which is selected), 'Layout', and 'Manage'. Under 'Configuration', there are tabs for 'General', 'Offline', 'Security' (selected), and 'Custom modules'. In the main content area, the 'Authentication and Security' section is displayed. It includes a list of authentication methods: 'Login not required', 'Pega Platform account' (which is selected), 'Authentication service selected by the end-user', and 'Specific authentication service'. Below this, there is a section titled 'Enable application lock' with a checkbox checked. A dropdown menu next to it says 'Biometrics and device lock'. There are two other checkboxes: one for 'Lock app after 30 minutes of inactivity' and another for 'Lock app after 2 minutes of session duration'. To the right of the configuration interface is a screenshot of a mobile device displaying a lock screen. The screen has a numeric keypad from 1 to 9, with '0' at the bottom. It says 'Touch ID' and 'Unlock your phone to continue'. At the bottom of the screen are navigation icons for Home, My Work, Create, Search, and More. The top of the mobile screen shows 'User Mobile App' and 'Enter PIN'.

<https://academy.pega.com/topic/mobile-app-channels/v1/in/26826/24056/23811>

# Mobility – Mobile app branding

- Pega Platform™, allows you to build custom mobile apps in a low-code environment. You can tailor the app's interfaces to reflect your brand identity. For a professional look and feel and to ensure a consistent user experience across the channels, you can customize the interface to match the branding requirements.
- Icon and launch screen design (App Branding) helps prospective users to find your published mobile app.

The screenshot shows the Pega Platform Configuration interface. On the left, a sidebar lists 'Content', 'Configuration' (which is selected), 'Layout', and 'Manage'. Under 'Configuration', there are sections for 'General', 'Offline', 'Security' (selected), 'Custom modules', and 'Hosting'. In the main area, the 'Authentication and Security' tab is selected. It shows the 'Authentication method:' section with 'Pega Platform account' selected. Below it, the 'Enable application lock' checkbox is checked, and its configuration options are highlighted with a red box. These options include 'Unlock with' set to 'Biometrics and device lock', 'Lock app after 30 minutes of inactivity', and 'Lock app after 2 minutes of session duration'. To the right of the configuration interface is a mobile device displaying a lock screen titled 'User Mobile App'. The screen shows a numeric keypad from 1 to 9, a 'Touch ID' button, and a 'Cancel' button. At the bottom of the device screen are navigation icons for Home, My Work, Create, Search, and More. Icons for Apple, Android, and PEGA are visible at the top right of the device screen.

<https://academy.pega.com/topic/mobile-app-channels/v1/in/26826/24056/23811>

# Mobility – Mobile Theme customization

To ensure a consistent experience throughout the user workflow, define an application-wide theme for your app. In the Layout tab, specify colors for various interface elements and observe the changes in real-time on the mobile app preview.

The screenshot shows the Pega Platform's 'Layout' tab for 'Branding' and 'Theme'. The 'Theme' section includes settings for 'Bottom menu style' (Icons and text) and 'Floating action button position' (Left/right aligned). A checkbox for 'Enable auto hiding bars on scroll' is also present. A red box highlights the 'Mobile specific colors' section, which lists color swatches for various UI components: Mobile header background, Mobile bottom navigation, Primary swipe action, Secondary swipe action, Tertiary swipe action, Primary button, Standard button, Simple button, and Floating action button. To the right, a mobile app preview shows a green header with 'Header' and a floating action button. The main screen displays 'Title' and 'Link Link' buttons, followed by sections for 'Headings' (with sample text), 'Primary Button', 'Standard Button', and 'Simple Button'. The bottom navigation bar includes icons for Home, Expenses, New, Chat, and More.

<https://academy.pega.com/topic/mobile-app-channels/v1/in/26826/24056/23811>

# Mobile Channel

Content Configuration Layout Manage

Description

Content

- Controls the content available to users in the mobile app. Customize app navigation and functionality by:
  - Adding reordering, and removing the default mobile app pages, such as search functionality and notification lists
  - Creating customized app content using mobile list pages
  - Defining swipe actions for list items to provide users with quick access to common tasks

Configuration

- Controls basic app functionality, such as
  - The app name as displayed on the mobile device
  - The role assigned to users of the mobile app
  - Whether the app supports offline processing
  - App security, including user authentication and session management

Layout

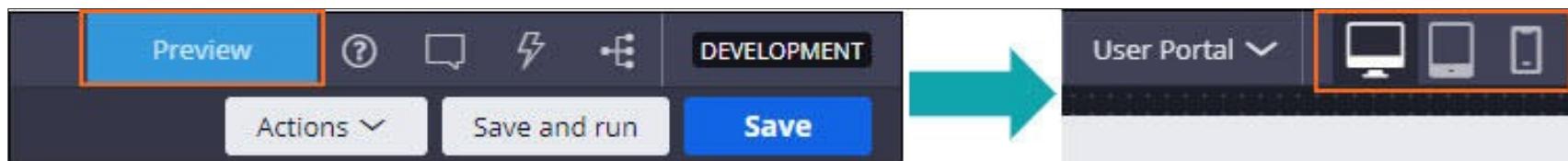
- Controls app branding and styling options, including the launch page and UI element colors. You can change the default icon, using native icons or custom files for the icon to personalize your app.

Manage

- Controls administrative functions such as log access and administrative push notifications.
  - **Note:** Administrative push notifications, which are sent from the console on this tab, differ from the app-based push notifications sent during case processing. For example, you use the administrative push notification console to inform users to upgrade to a newer version of the mobile app.

## Previewing mobile apps

In App Studio, you can preview your application to see how it looks on various devices using the **Preview** feature.



- Before you generate an executable file for your mobile app, verify that the pages and layout correspond with your design, and that the application logic functions in line with your expectations. Use a mobile device with the Pega Mobile Preview app to preview a mobile channel that you configure for an application.
- With Pega Mobile Preview, you do not have to obtain certificates or generate executable files before you access the contents of your mobile app.

## Definitions

Users

An individual who interacts with an application

Roles

Defines how users interact with the application

Channels

Denotes where data associated with a data type is located

Persona

Defines the type of access that you grant to the users of your application

Access Groups

An access group is a group of permissions within an application

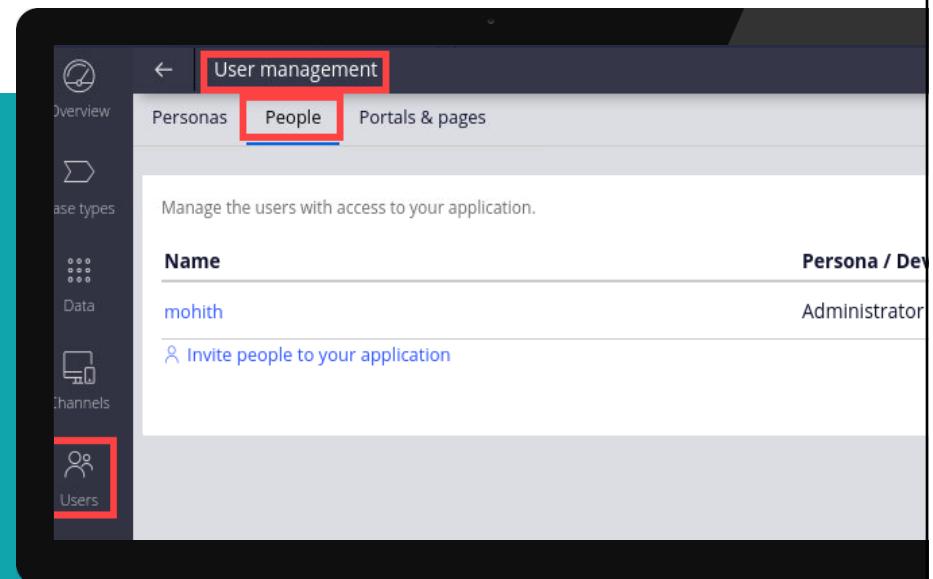
Operator

An operator defines a unique identifier, password, preferences, and personal information for a user

# Viewing users

## Navigation

- To view from App Studio select:  
**Users > User management > People**
- To view users from Dev Studio select:  
**Configure > Org & Security > Organization > Operators**

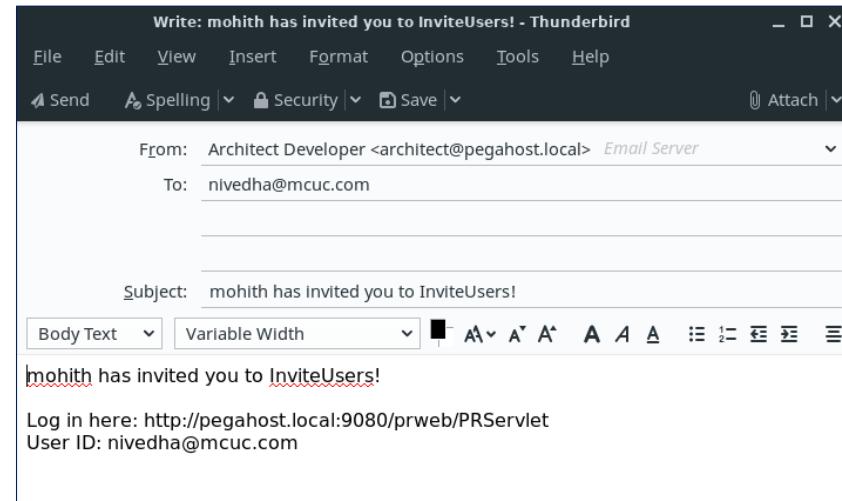




# Inviting Collaborators

## Description

- Enhance your application by inviting collaborators with different skills and roles
  - For example, invite members of a development team to start working on building your application and begin processing your business cases
- When you invite collaborators, you associate them with personas



## Mobility & Security 8% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

<https://www.classmarker.com/online-test/start/?quiz=jpv62297db3757e1>



