Pega Certified System Architect (PCSA) 8.7 Exam Prep

Agenda

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

Pega Certified System Architect Exam

Certification Path:

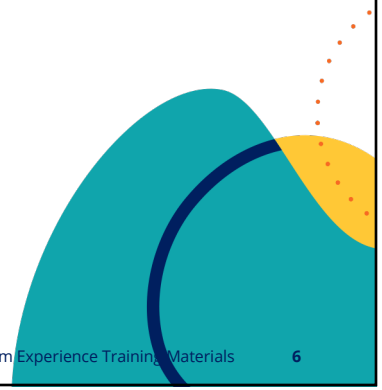
Pega Certified System Architect Exam

- Consists of 60 multiple choice exam questions – 57 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 – 65 % or 37 questions answered correctly

Pega Certified System 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

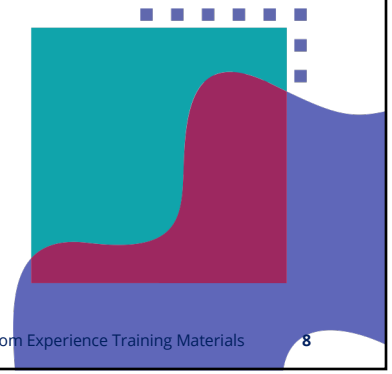


Pega Certified System Architect Exam

- **The following Exam Code is used for registration:**
  - **Exam Code:** PEGAPCSA87V1
- The exam is currently in the following languages:
  - English, French, German, and Japanese

Pega Certified System 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

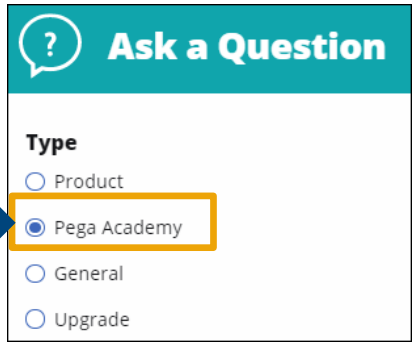


Study Materials

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 PCSA exam topics and their distribution % here:

- https://academy.pega.com/exam/pega-certified-system-architect-3

Exam Advice

- Target your review using the slides in this deck
- You can also use the System Architect Essentials Student Guide
  - **https://v953w.app.goo.gl/ux2j**
  - 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**

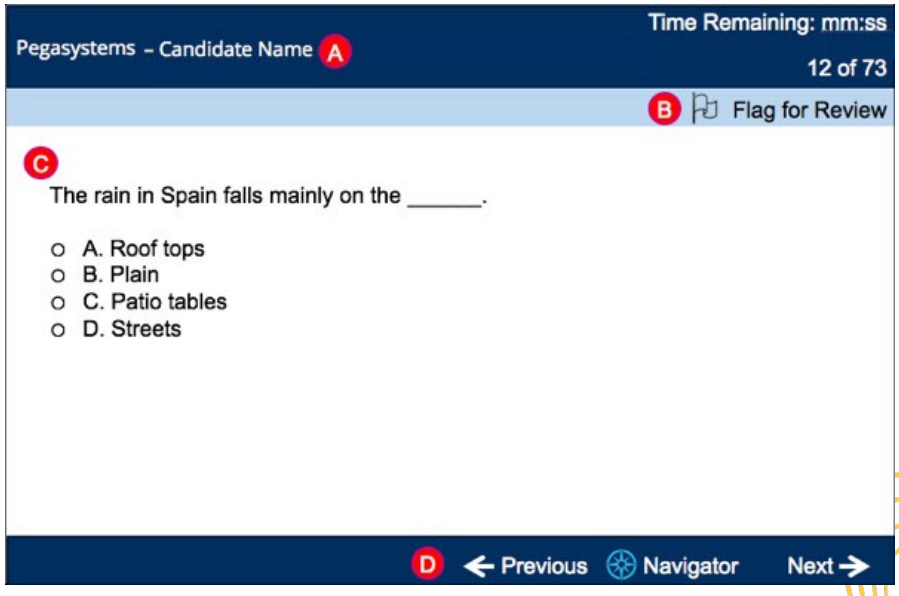
Exam Advice

- 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.

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

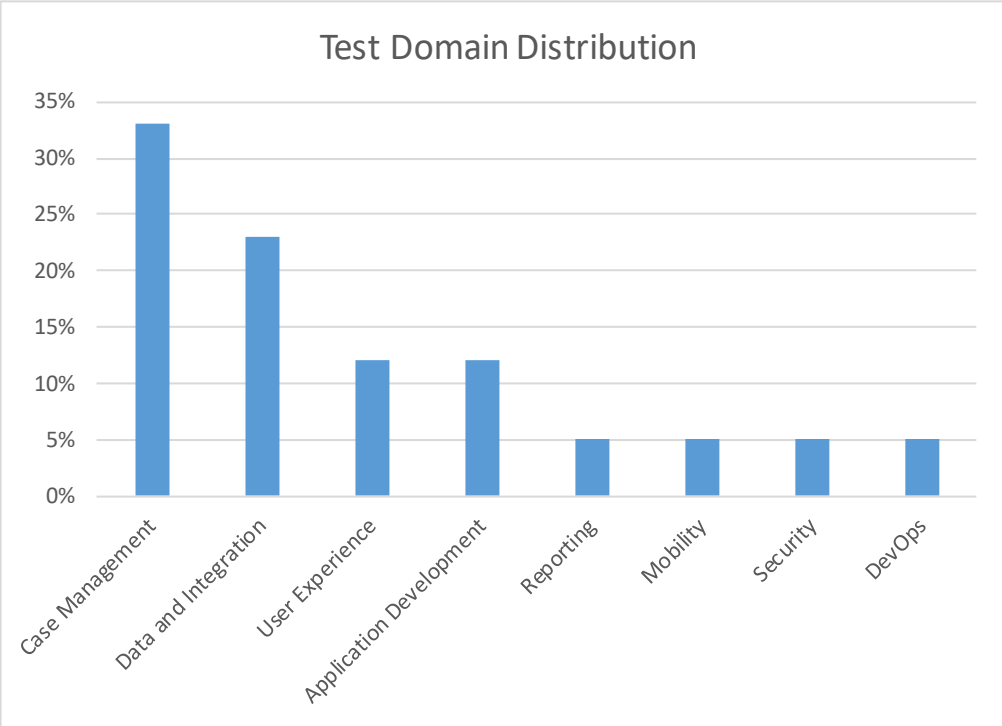
Submitting the Exam

- 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.

PCSA 8.7 Test Domains

Requires 65% to pass

Test Domains	% of Exam	# Questions
Case Management	33%	20
Data and Integration	23%	14
User Experience	12%	7
Application Development	12%	7
Reporting	5%	3
Mobility	5%	3
Security	5%	3
DevOps	5%	3
Total	**100%**	**60**



Exam Blueprint here:
https://academy.pega.com/exam/pega-certified-system-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 behaviour 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 60 question practice exam:

https://www.classmarker.com/online-test/start/?quiz=jr96228f1124022d

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

Self Study Mission Test: https://academy.pega.com/mission/system-architect/v4/test/in/29826

Case Management
33%

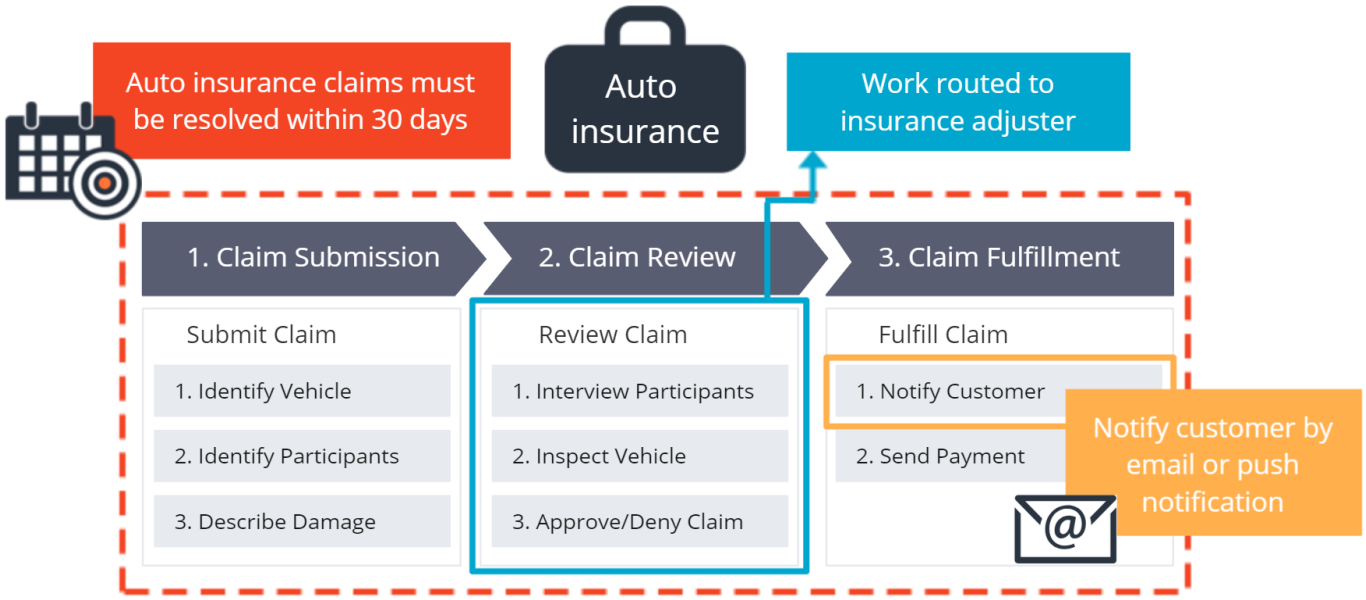
Exam Blueprint

Case Management 33%

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

Case Management - Model-Driven Design

- Case types
- Cases
- Service Levels

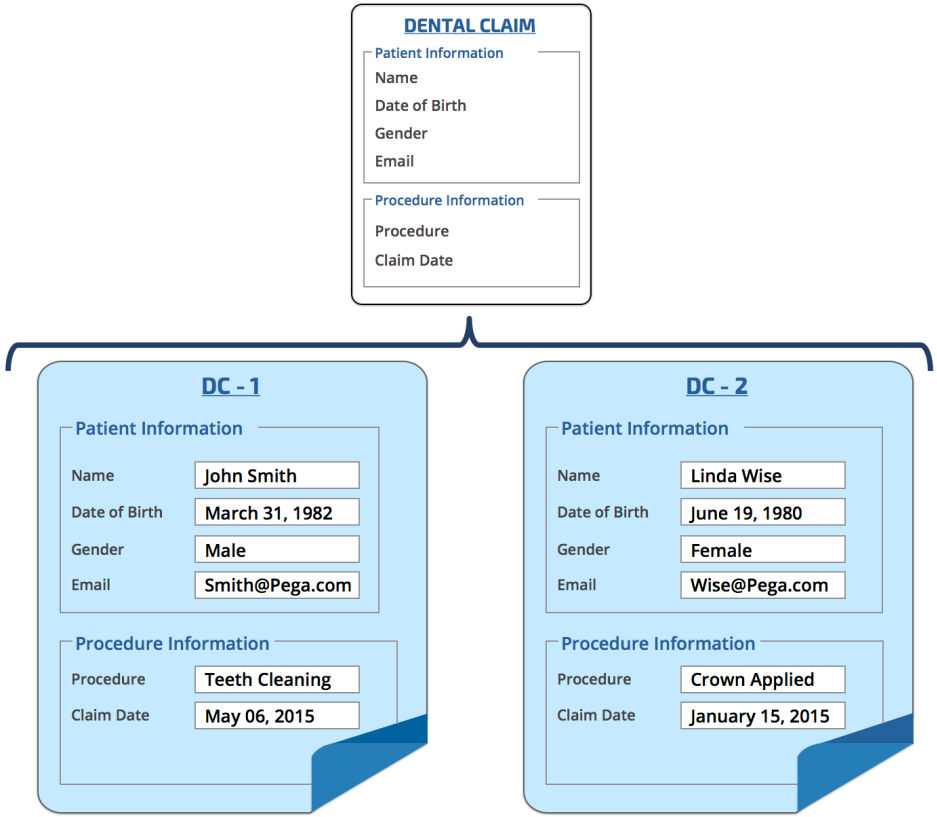


Case Management – Case Types and Cases

- 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 – have a unique identifier, status, urgency and follow at least one process
- Cases are represented as a composite, an object, and they allow us to associate a number of 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.



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

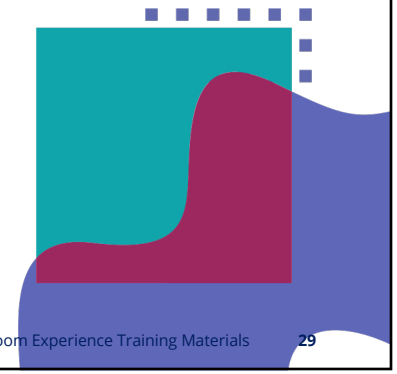
Case Management– Stages, Processes, and Steps

- 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

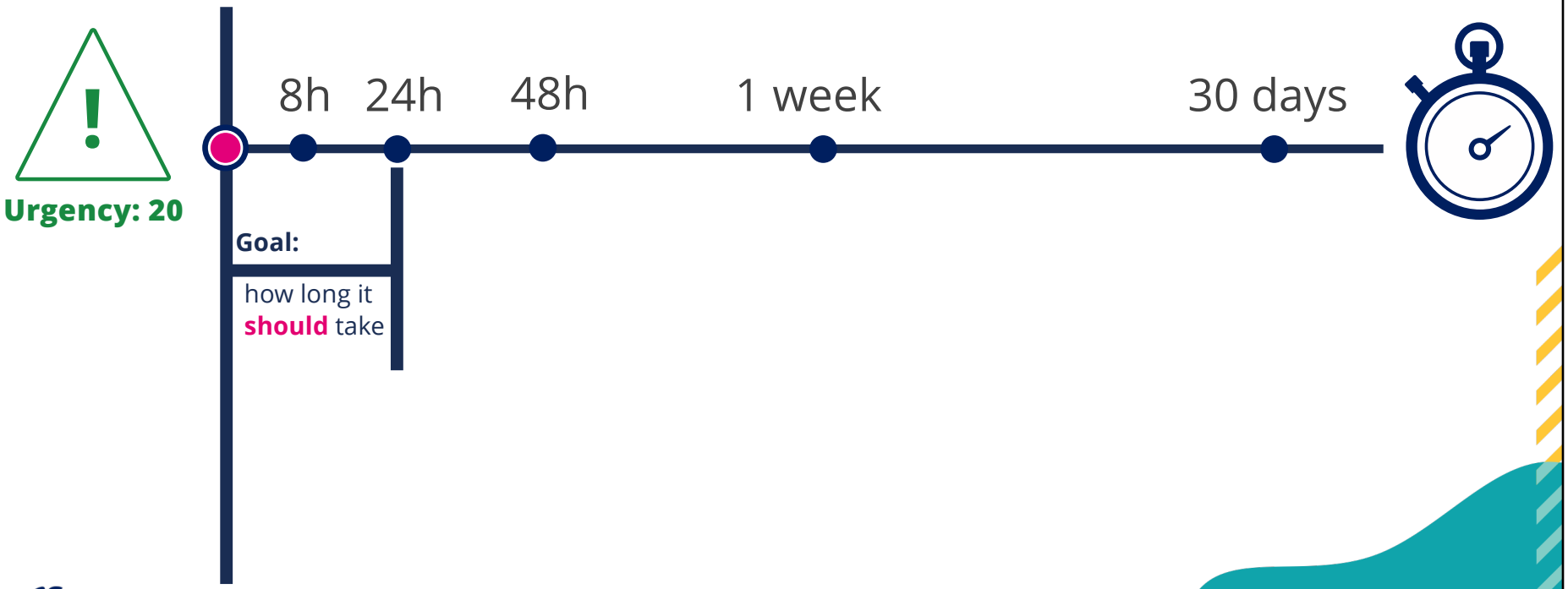
- Service Level Agreements or SLAs are used to ensure work is completed within a certain time frame.
- We define service levels using 3 milestones:
  - Goal – how long it ***should*** take – measured from assignment start
  - Deadline – how long it ***may*** take – measured from assignment start
  - 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

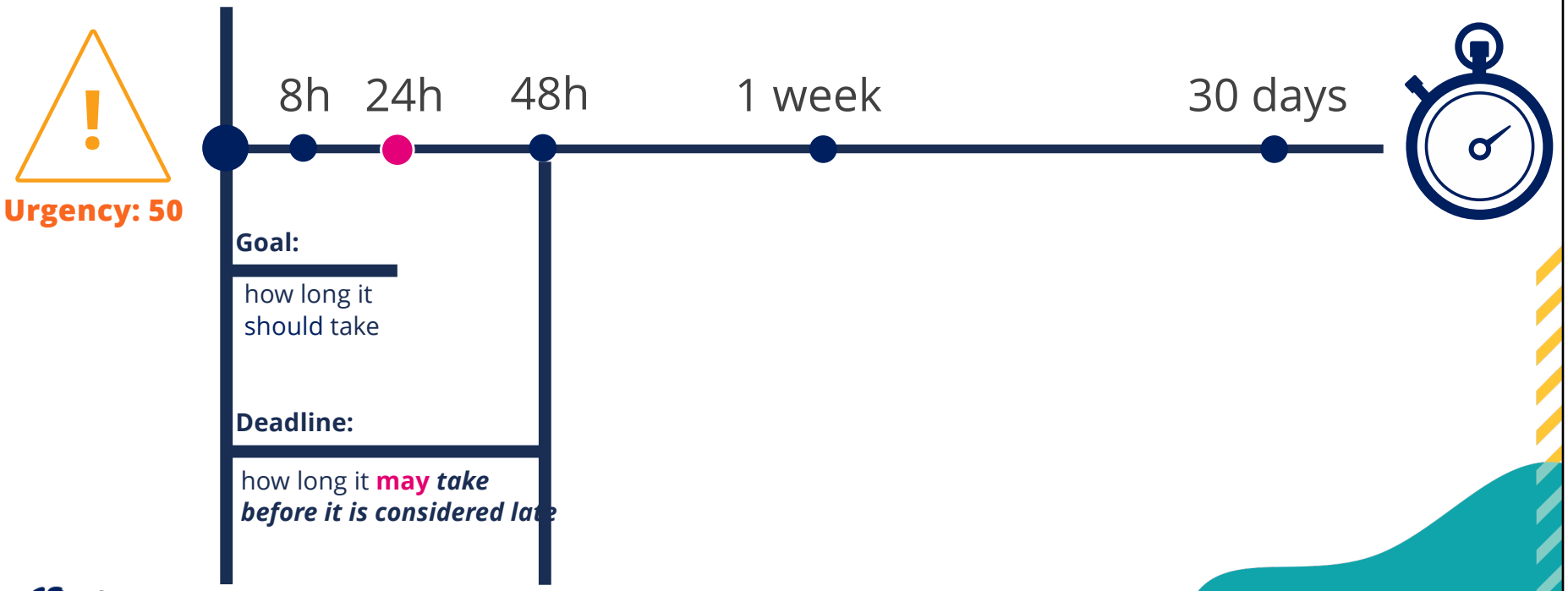
Case Management – Service Levels

Service Level Agreements



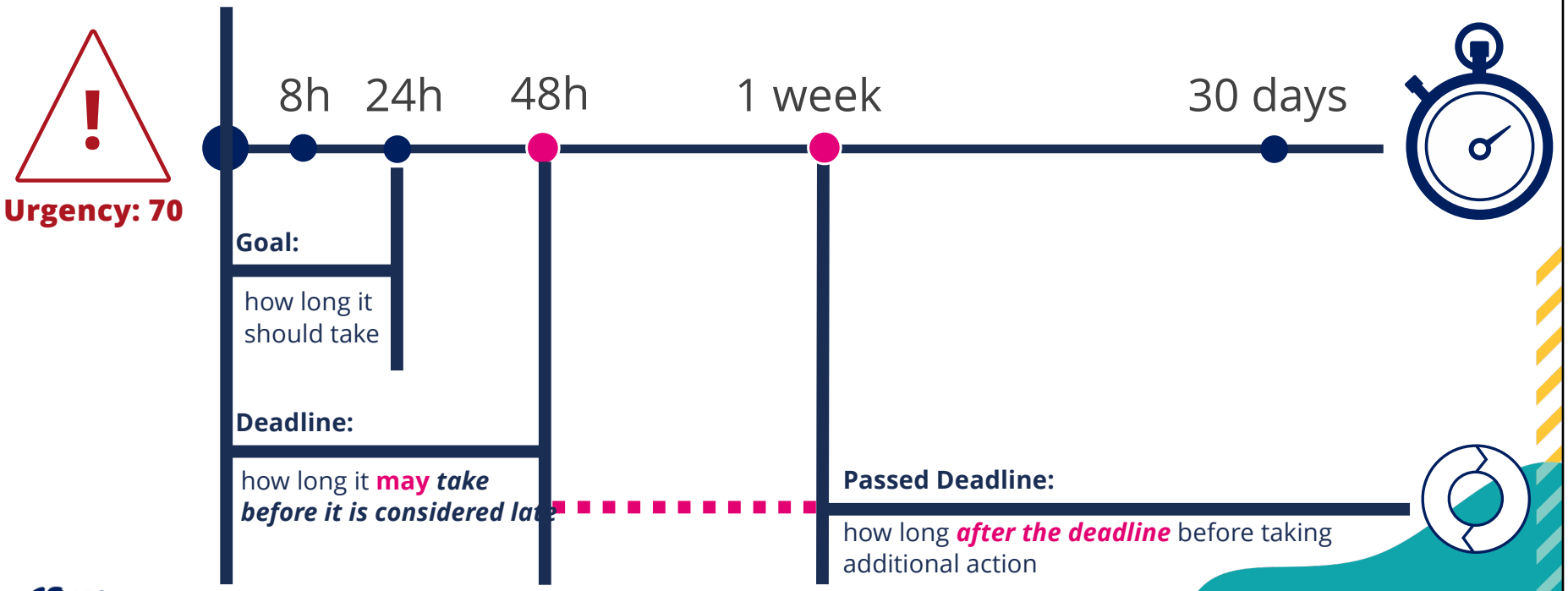
Case Management – Service Levels

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

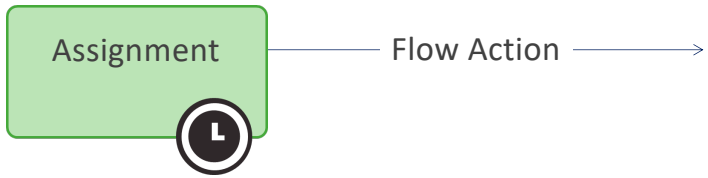
Worklist

- **Work Queue** – routing queue where assignment are stored; multiple operators can access them

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

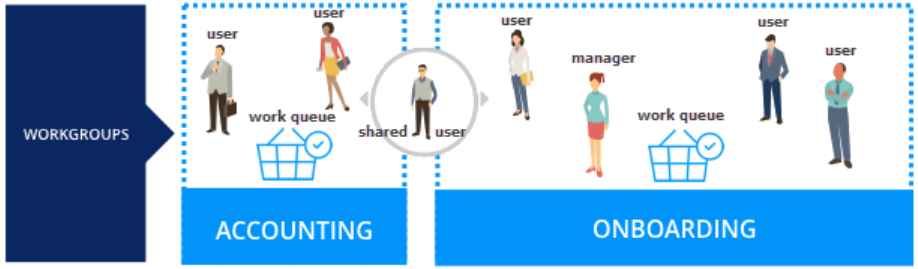
- Case routing to a **Work Queue**
  - A case is routed to a queue
  - System creates an assignment object in memory
  - The assignment appears in the work queue list
  - It stays there until it gets assigned to an operator
  - The assignment moves to the operator's worklist
  - When the operator completes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

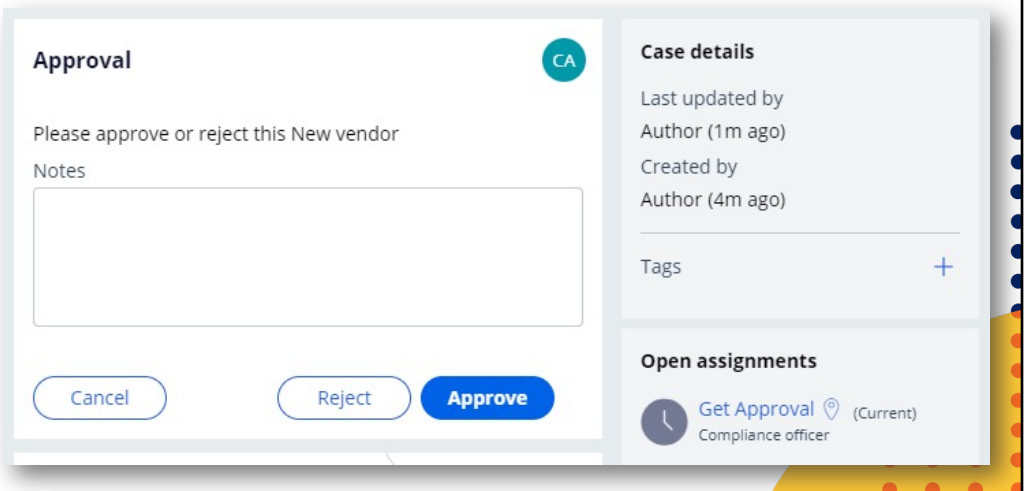
Some Standard Routing Activities available out of the box in Pega 8:

- 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

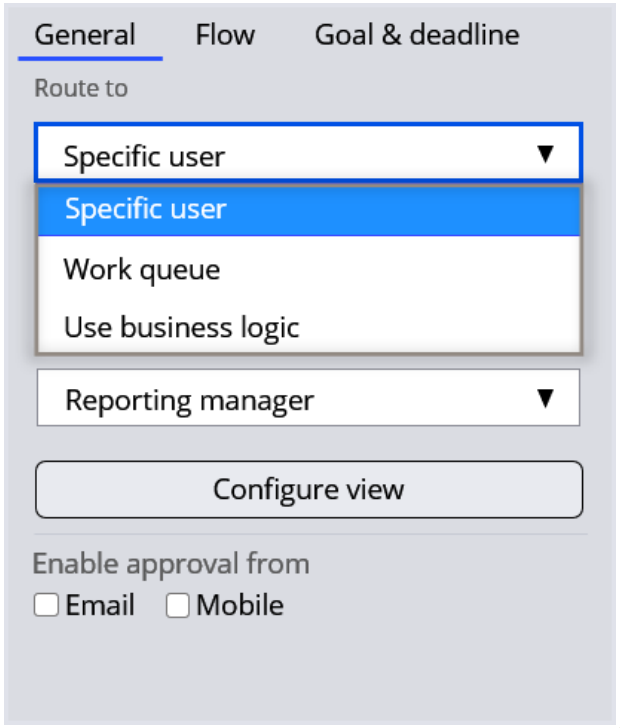
- Single level
  - Receive approval from a single user by indicating a specific user, work queue or business logic.
- Cascading
  - Configure reporting structure or authority matrix
  - Receive approval from people on different levels of your organizational chart or in different parts or departments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

	Conditions		Actions
	TotalPRCost >=		Approver ID
when	0	→	"CSM@Pco.com"
when	25000	→	"VP@Pco.com"
when	75000	→	"VPFinance@Pco.
when	100000	→	"SVP@Pco.com"
when	500000	→	"CFO@Pco.com"
when	500001	→	"CEO@Pco.com"
otherwise		→	""

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.

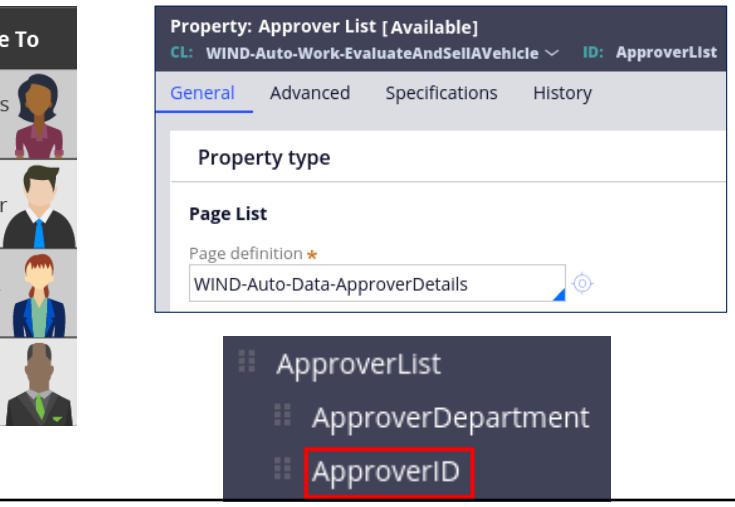


Cascading Approval Models

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



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

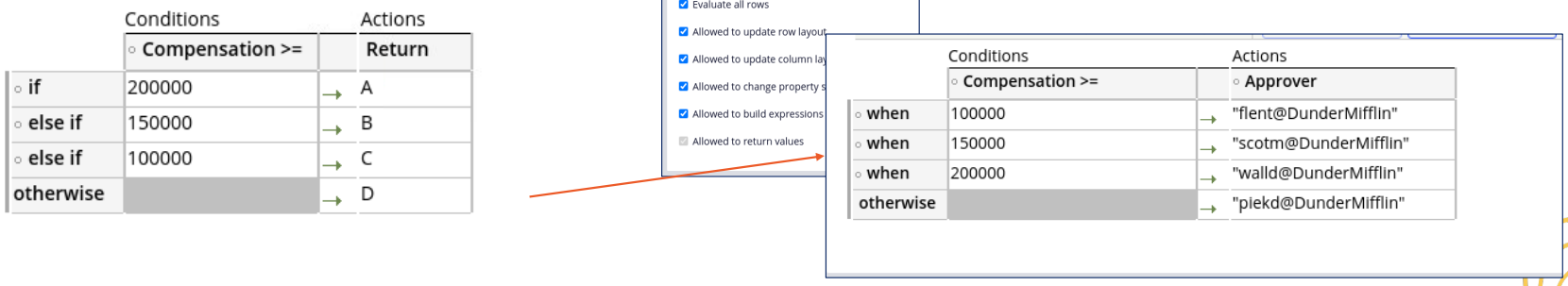
If Amount Is	Route To
<$500 USD	Manager
>$500 USD	Director
>$2000 USD	VP

Authority Matrix - Configuring Decision Table

Implementation

Set Evaluate all rows option and configure the logic

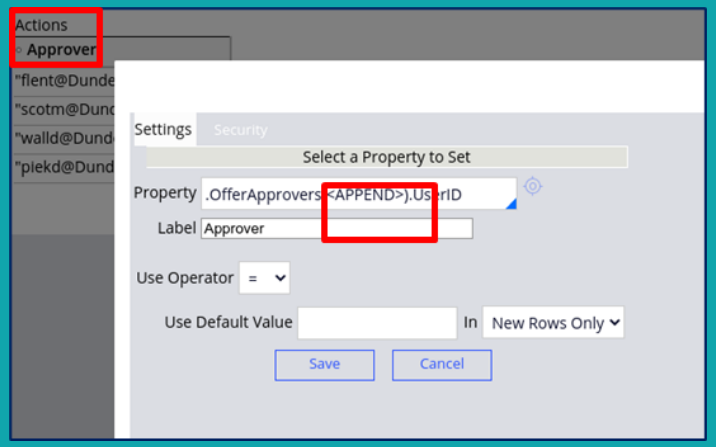
- The first page of the authority matrix page list will be populated with the Action value of the first true row from the decision table each subsequent true row will add another page with the given approver.
- The ***otherwise*** row will only add an approver if none of the other rows evaluate false.



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

	Conditions		Actions
	Compensation >=		Approver
when	100000	→	"flent@DunderMifflin"
when	150000	→	"scotm@DunderMifflin"
when	200000	→	"walld@DunderMifflin"
otherwise		→	"piekd@DunderMifflin"

Authority Matrix - Decision Table and Symbolic Keywords



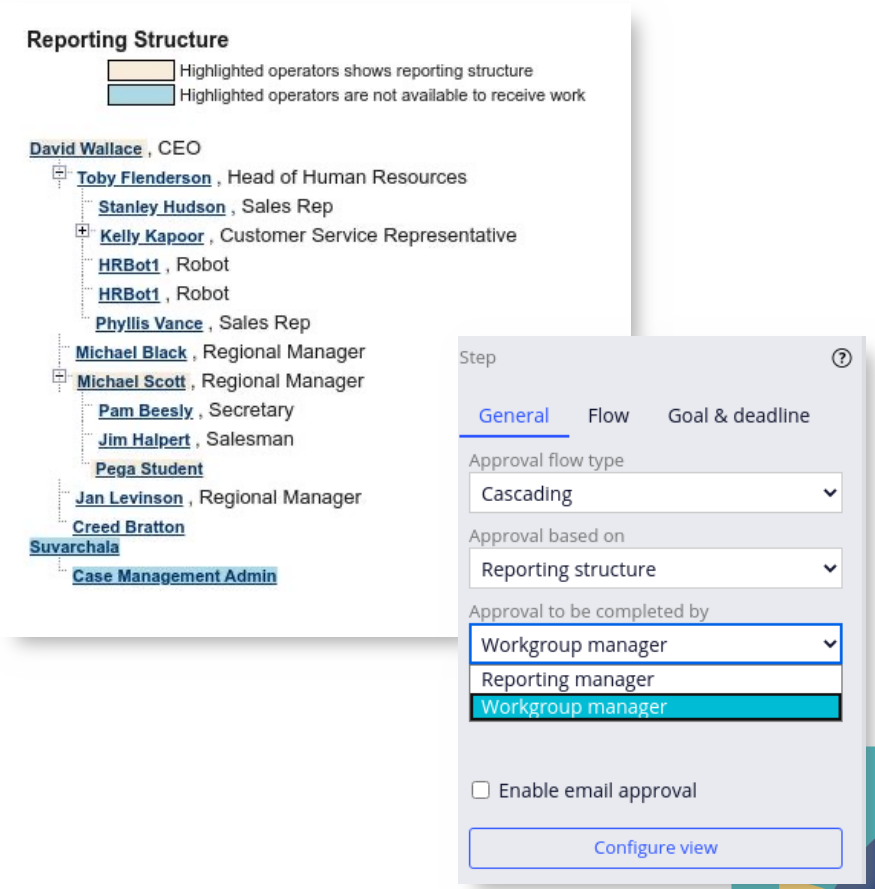
- The ***Actions*** column needs to be configured to append a new page to the list and set appropriate property for each true row.
- Pega provides a set of **symbolic indexes** to access items in a page list without using an explicit index number

<APPEND>	To add an element to the end (highest index value) a Value List or Page List property
<CURRENT>	In that context, the <CURRENT> index identifies the index value for the current iteration
<INSERT>	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
<LAST>	To set or retrieve an element value from the end (highest index value) a Data relationship property
<PREPEND>	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

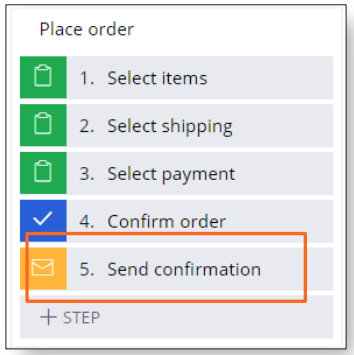
- Type of cascading approval that allows receiving approval from people on different levels of an organizational chart.
- Determines which operator is the starting point in the reporting structure.
  - Reporting manager
    - Reporting manager of current user
  - Workgroup manager
    - Manager of the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

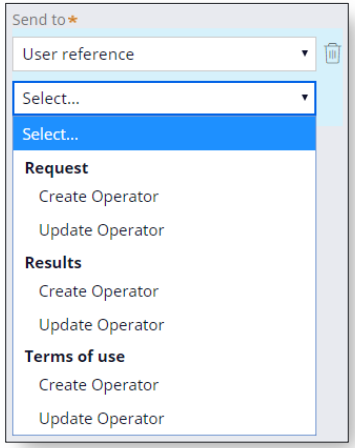


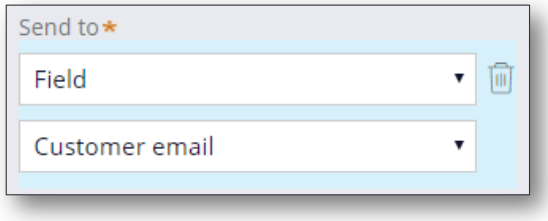
Correspondence - Identify who

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 that are involved in a case.



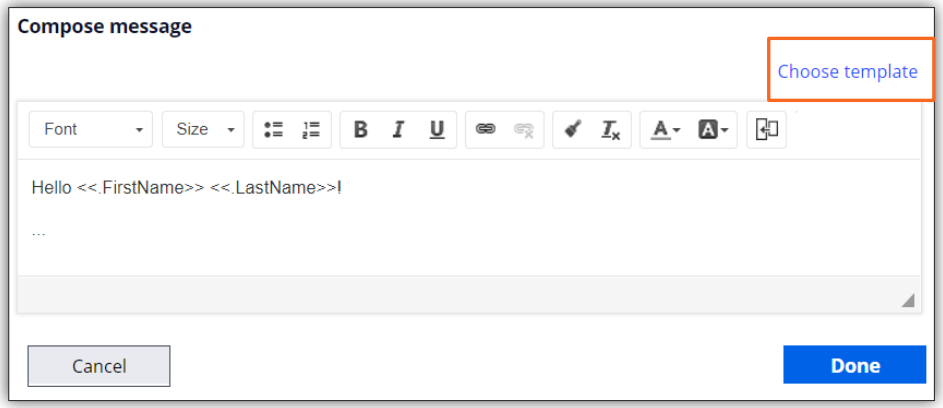




Correspondence - Identify how

Definition

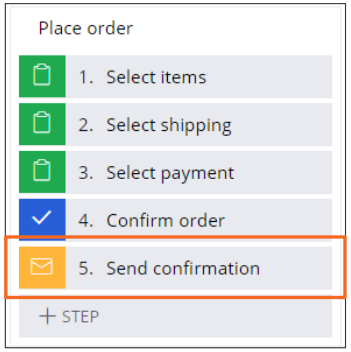
- Determine ***How will the communication be composed?***
- Pega Platform provides a rich text editor to create formatted email correspondence.
- The editor allows reuse of case data in the email.
- Pega Platform provides out-of-the-box templates that use case processing data, such as the case ID and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

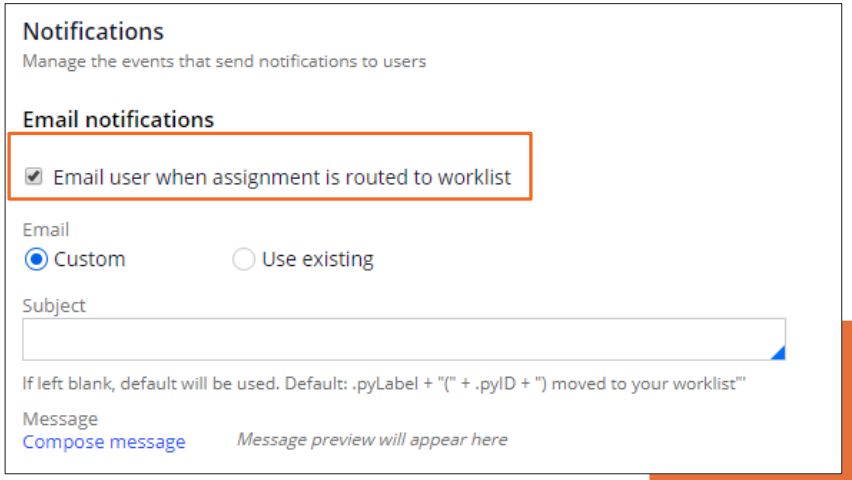


Correspondence - Identify when

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.





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?

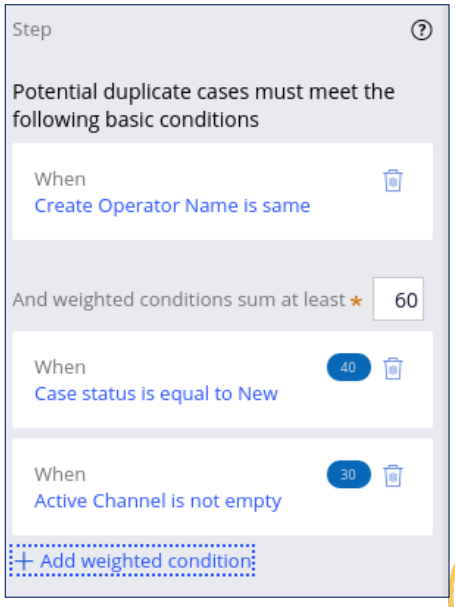
Definition

- 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

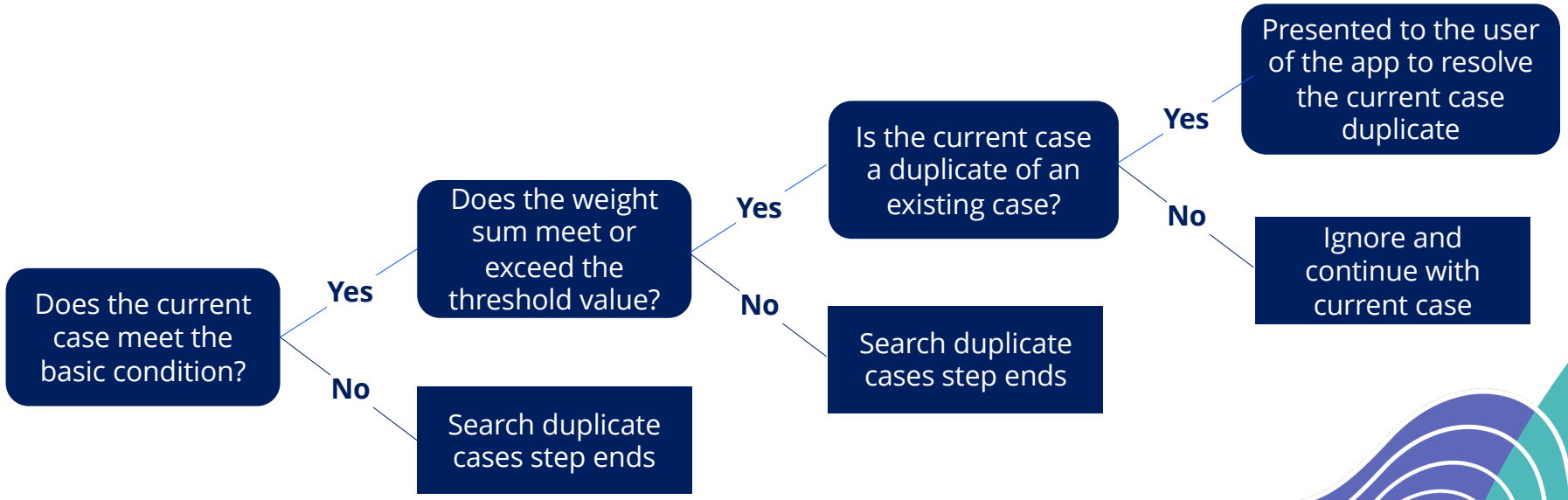
- Configure basic and weighted conditions.
- Basic conditions specify criteria that duplicates must meet.
  - Useful for reducing the number of cases to be matched and improving performance
- The sum of the weighted conditions must be equal to, or greater than a threshold.



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.



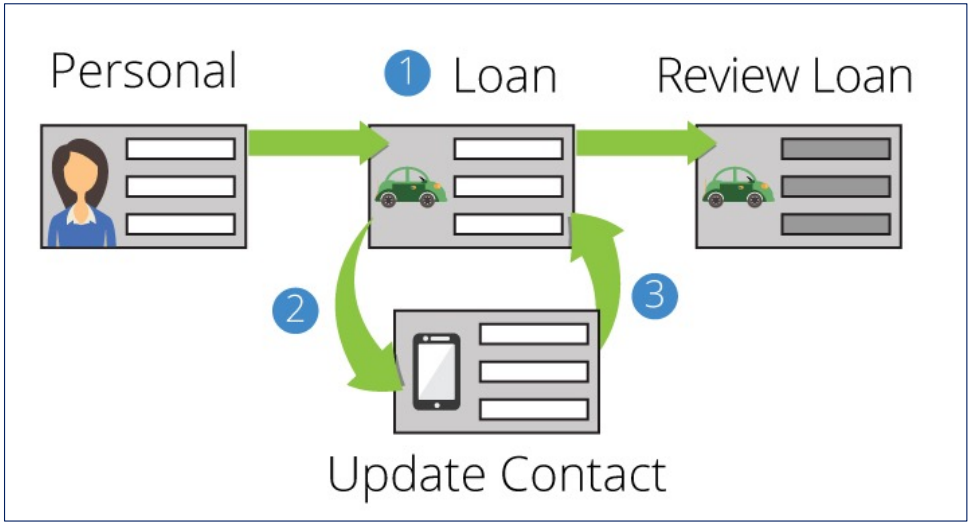
Optional Action - Use Case

- 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.

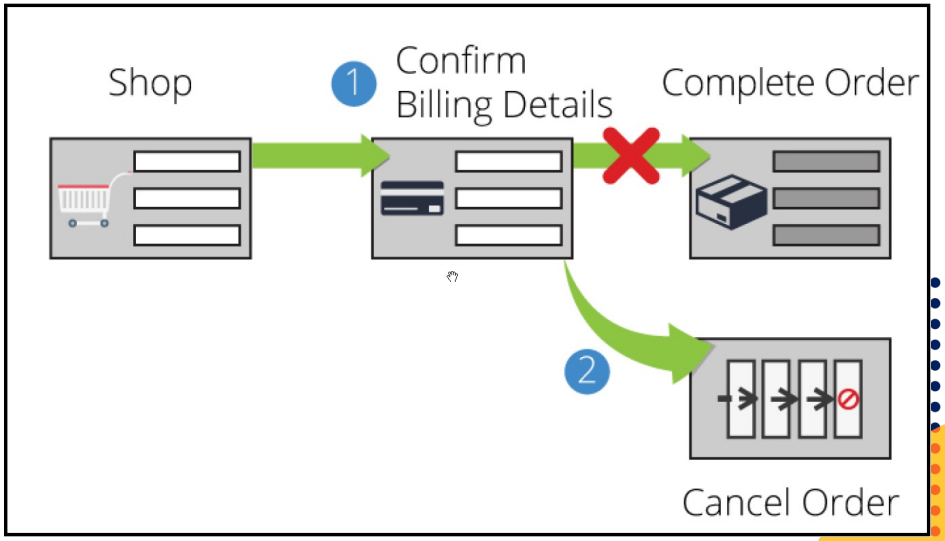
Case Management — Optional Action (Local Action)

Perform a single task without interrupting the primary path.



Case Management — Optional Process (Flow)

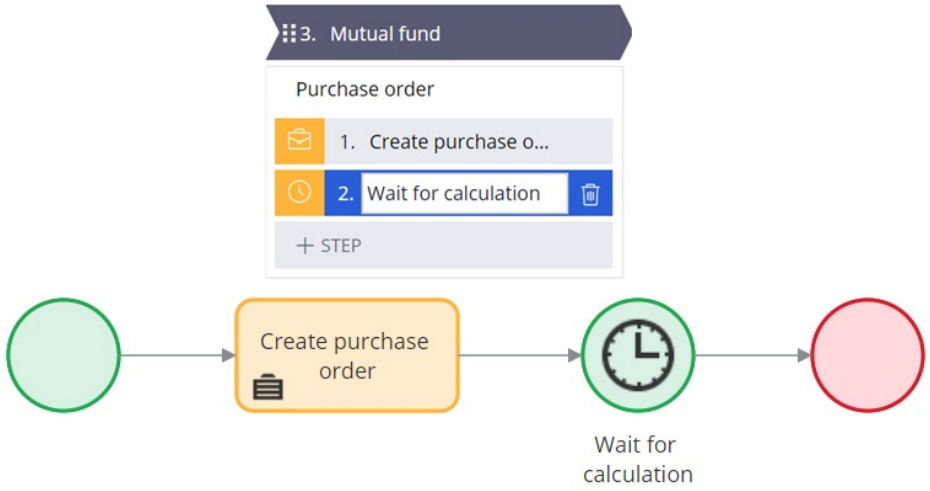
- Use optional process when a user is allowed to determine if a new process needs to be called.
- Used when multiple steps are needed to run a new process within the case.
- After completion, the process may, or may not return to the primary path.



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

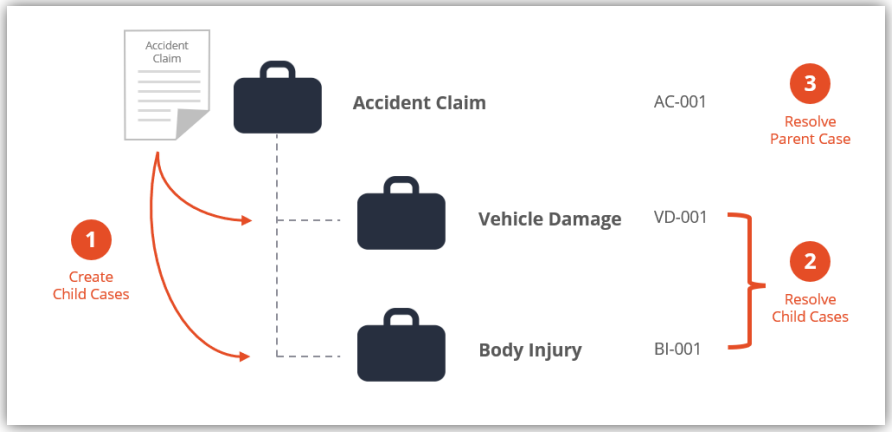
Definition and description

- 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

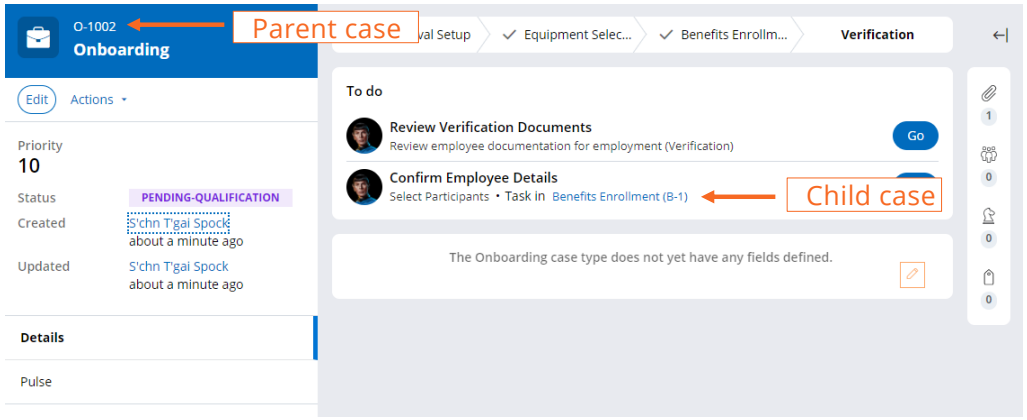
- **Child cases** are **beneficial** in situations for modeling work separately from the **Parent case when** several of the items in the list below are true:
  - Different data model is needed
  - Different life cycle is needed
  - Separate case ID and status for reporting is necessary
  - Separately 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.

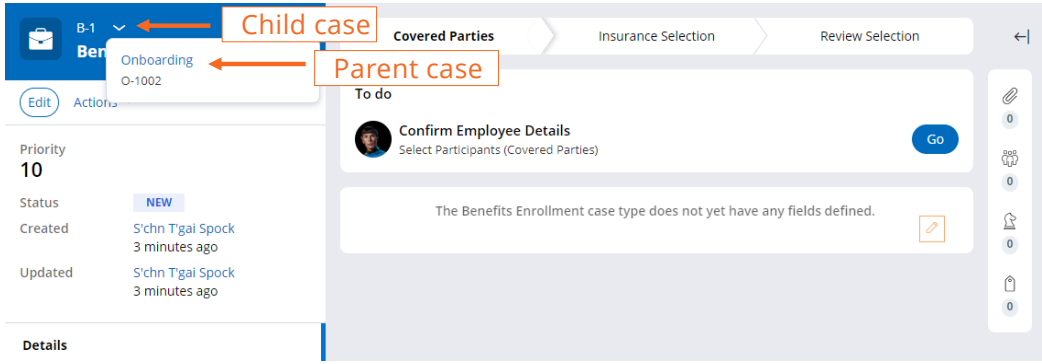


Child case

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***"

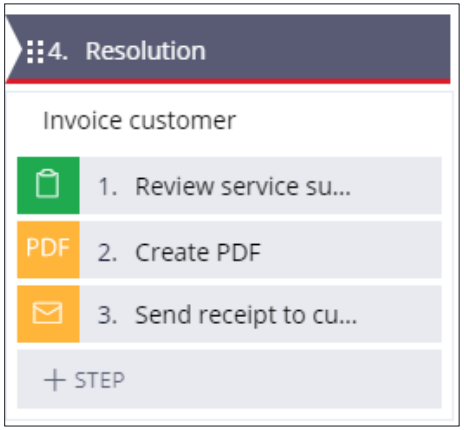
Data Propagation in Dev Studio

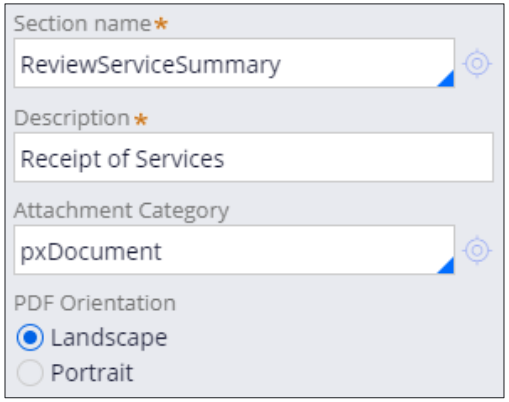
- Use a **Data Transform** to **copy the data values to fields identified** in a parent case to a child case.
- Option under the ***Settings tab select Data Propagation***

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.

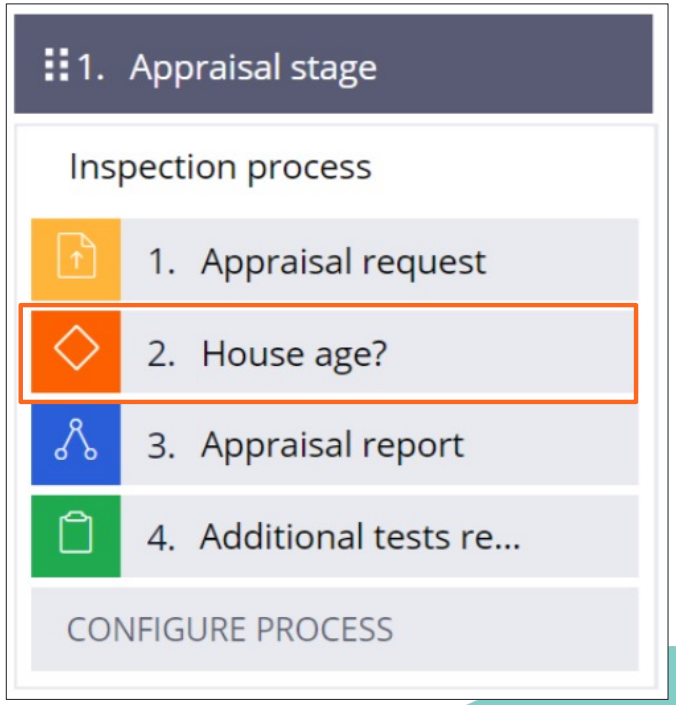




Decision points

Definition

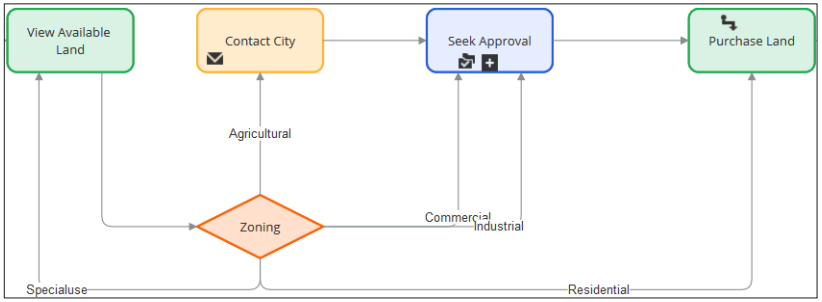
- Evaluates an expression or calls a rule to determine which step is next in the flow progression.
- Decision shapes create alternate paths in a flow because they can have more than one outgoing connector (points).
- Allows creation of different types of processes, or flows, based on the shapes or steps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

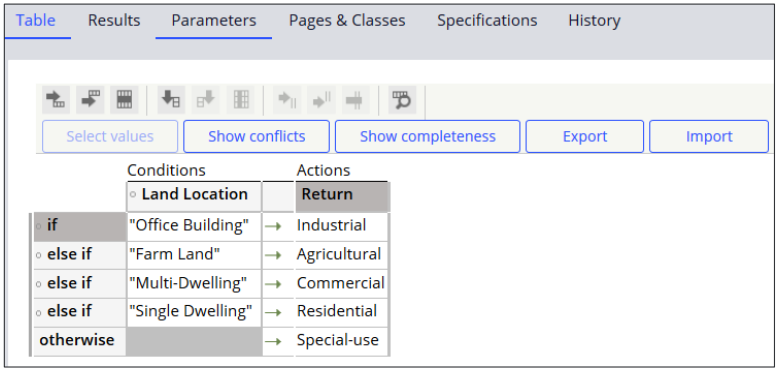


Decision Table

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.

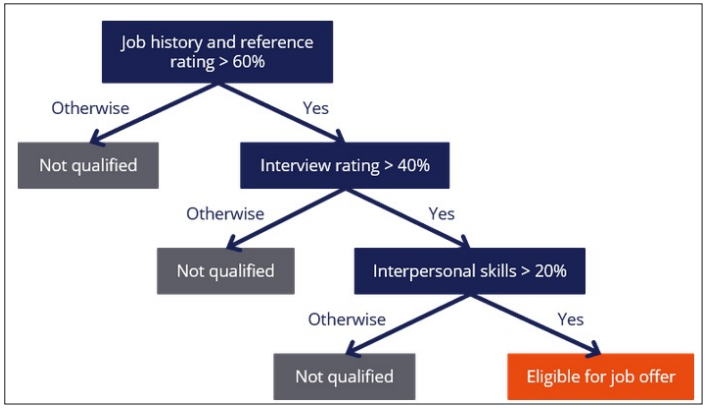


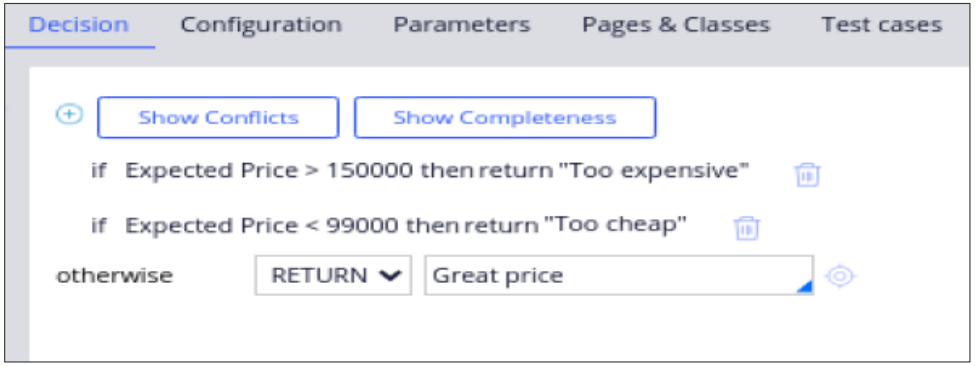


Decision tree

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.

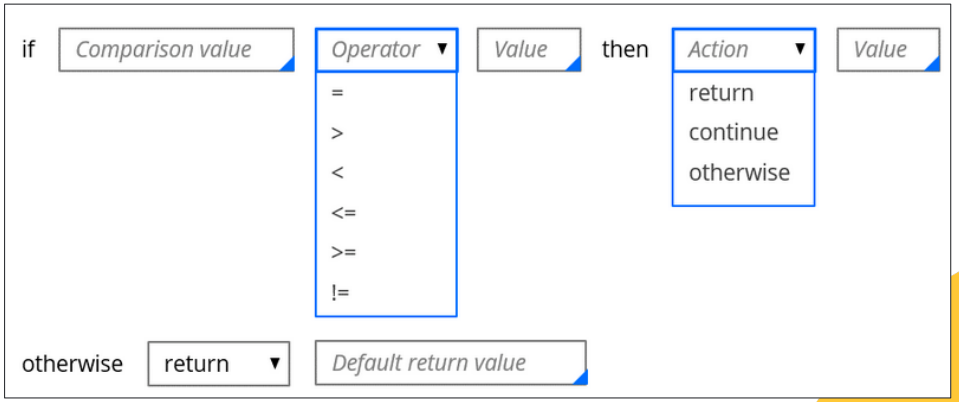




Tree conditions

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.



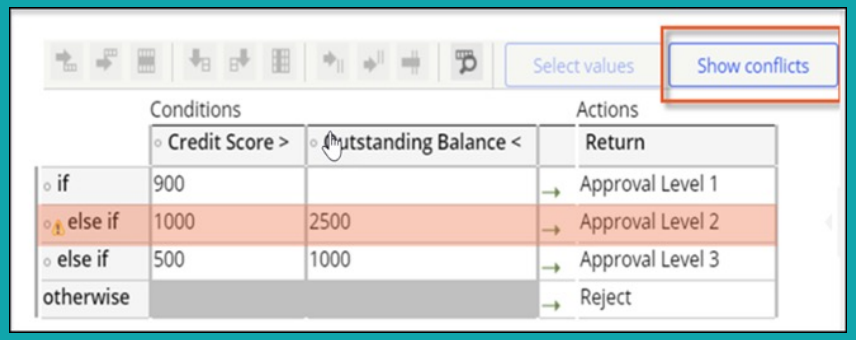
Decision rule conflicts

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

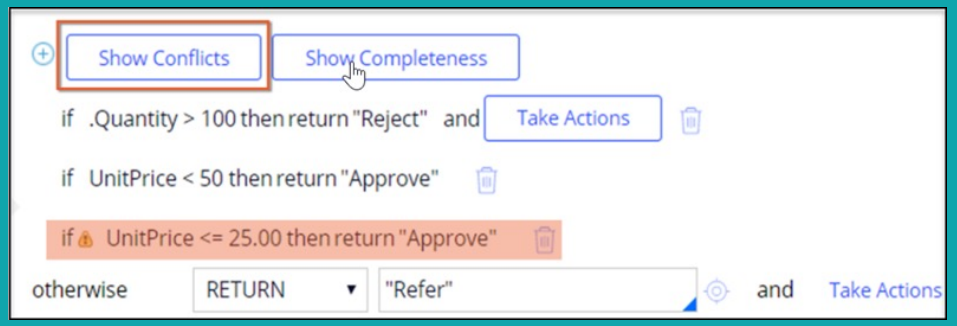
Show conflicts

- 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



Decision Tree conflicts

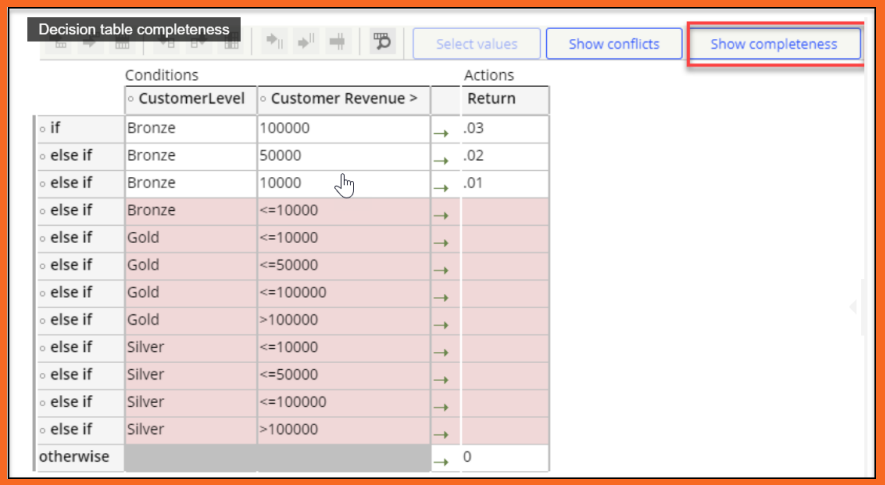


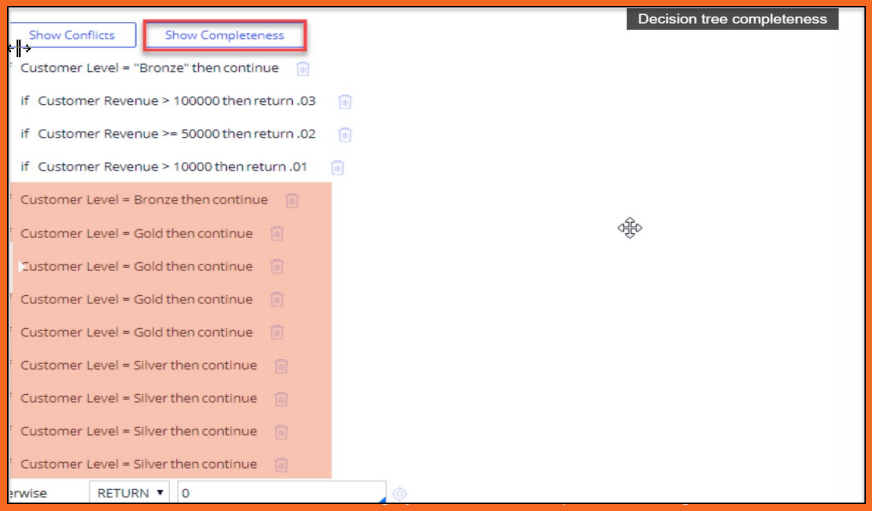
Decision rule completeness

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.





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 33% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=9t3622673aea80b3

Data and Integration
23%

Exam Blueprint

Data and Integration 23%

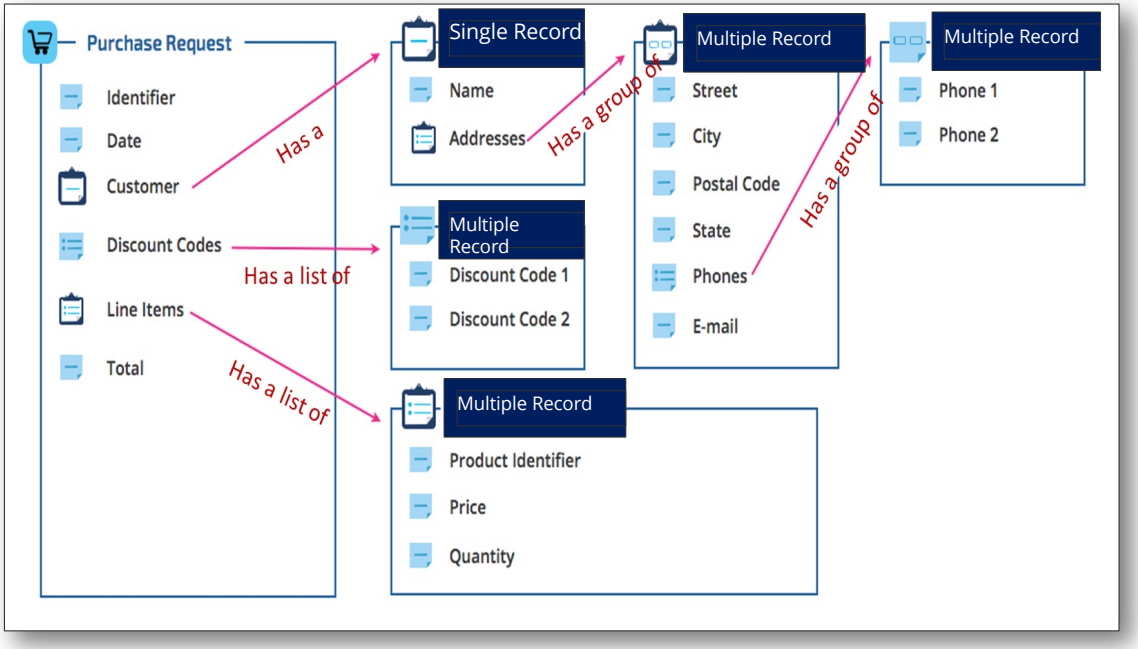
- Configure data types, create data objects, data relationships, and field types
- Identify and create calculated values
- Validate data; create and configure data validation rules by using business logic
- Manipulate application data, set default property values, configure data transforms
- Access sourced data in a case; refresh strategies; populate user interface controls
- Save data to a system of record
- Simulate and add external data sources
- Capture and present data; fields and views
- View data in memory; clipboard tool, pyWorkPage
- Configure field values
- Create and set application settings

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.

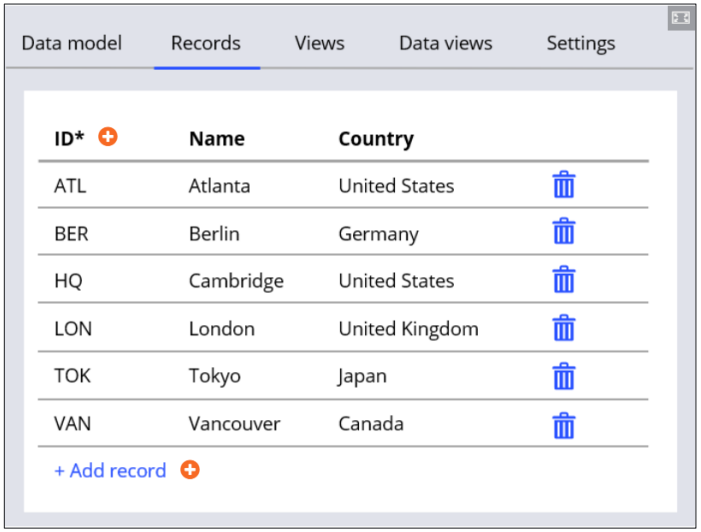
- **Case records**
  - Represents business transactions (e.g. Assistance Request)
  - Contains all the fields necessary to describe it
    - Customer (who needs assistance) (Identifier, Name, Address, Date of Creation, etc.)
    - Location of vehicle (address, city, state)
    - Type of vehicle (make,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

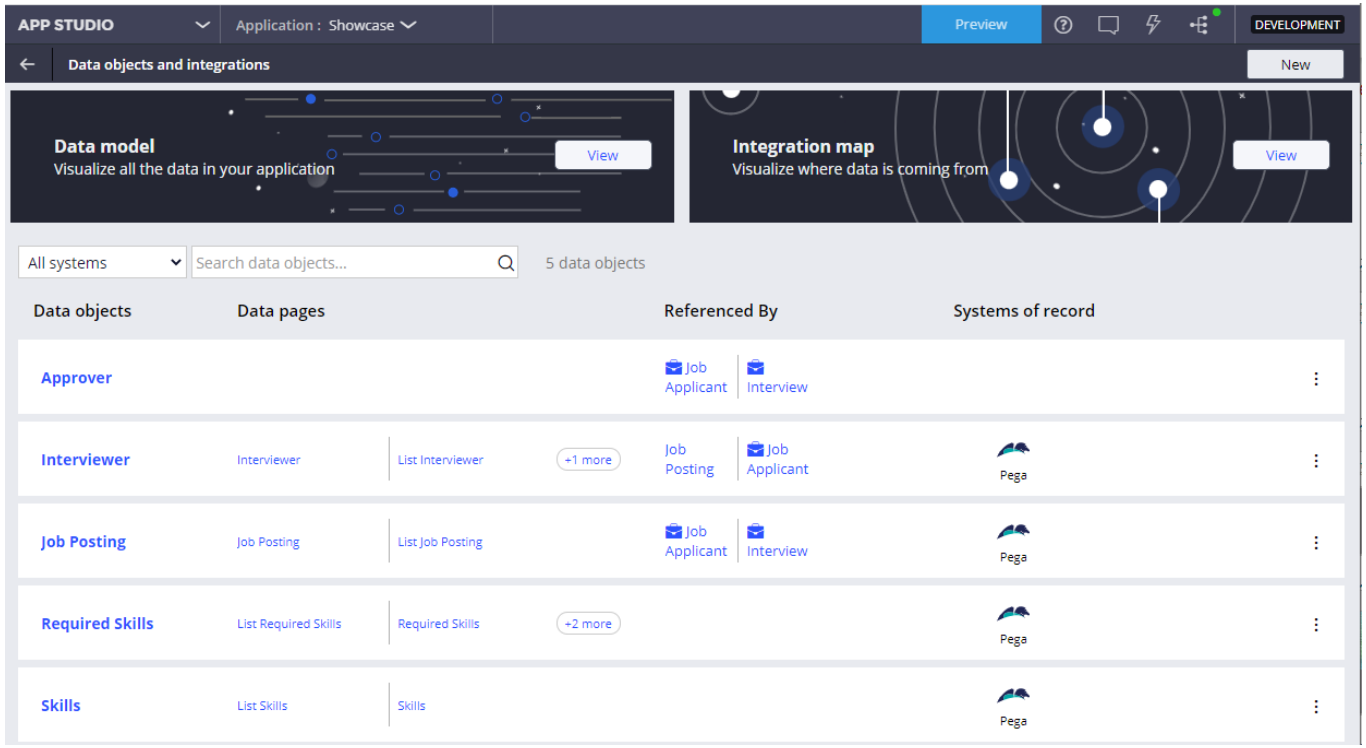
Description

- **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.



The integration designer

Definition



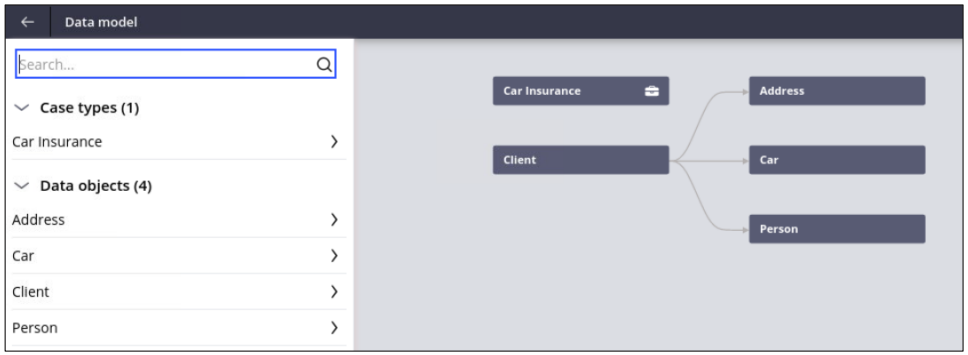
- Comprehensive view
- Displays data objects, case types and data types.
- Create data objects
- Access data objects
- Update data objects

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:

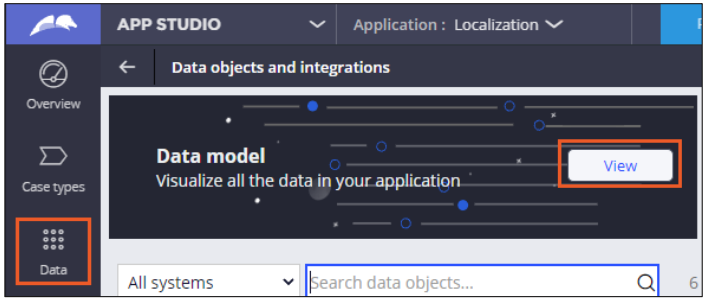
- **Data objects**
  - Categories of data that have fields, field mappings, and connections to data sources.
- **Fields**
  - Properties that store and format the data in your application.
- **Data sources**
  - Resources, which can be real or simulated, that host the data.
- **Field mappings**
  - Logic that links or transforms the fields in a data source to the fields in an application.

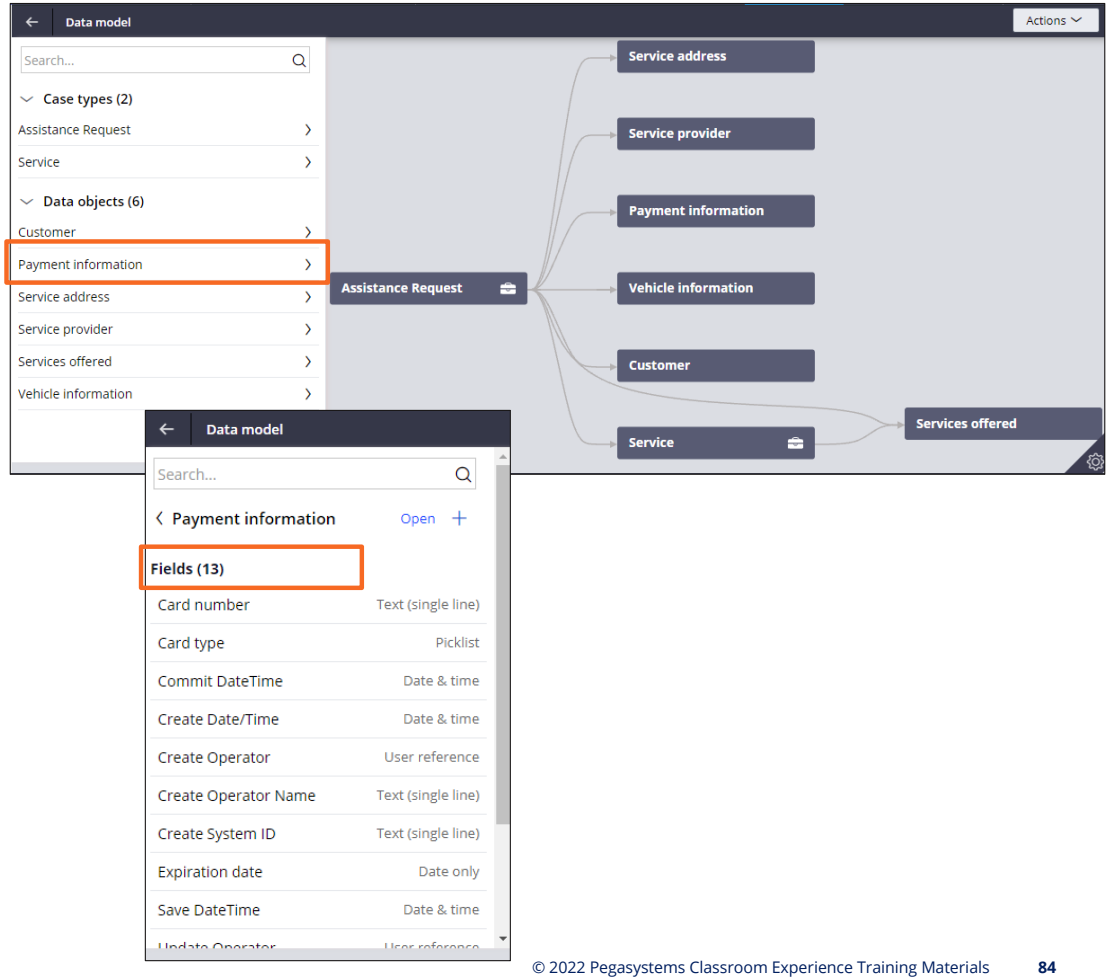


View the data model

Navigation

App Studio > Data > View

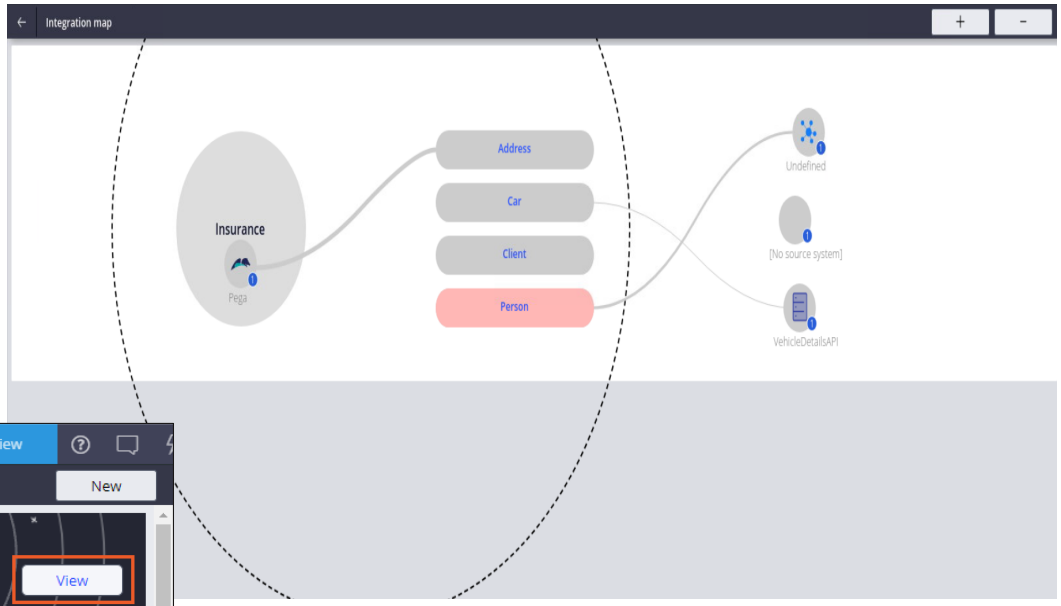


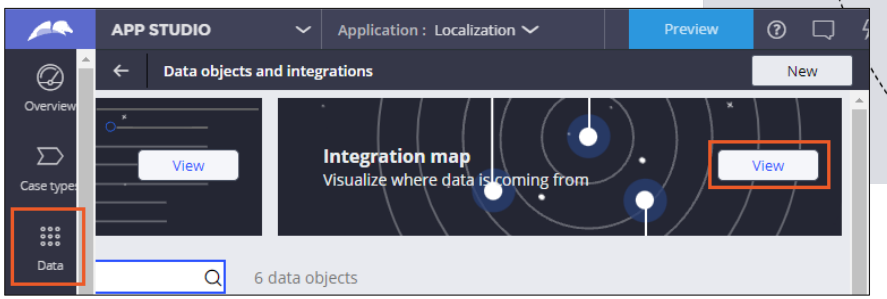


View integration map

Navigation

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





Declare Expression

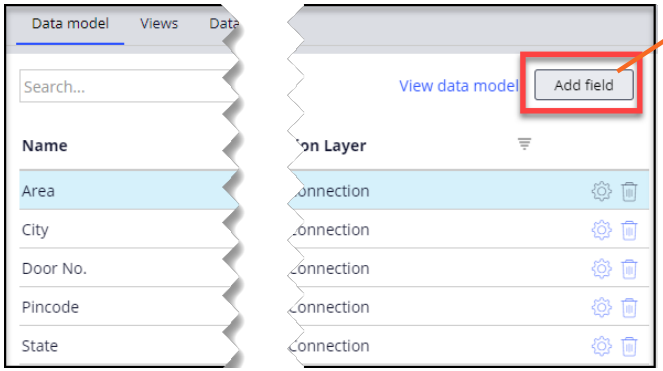
Navigation (1 of 2)

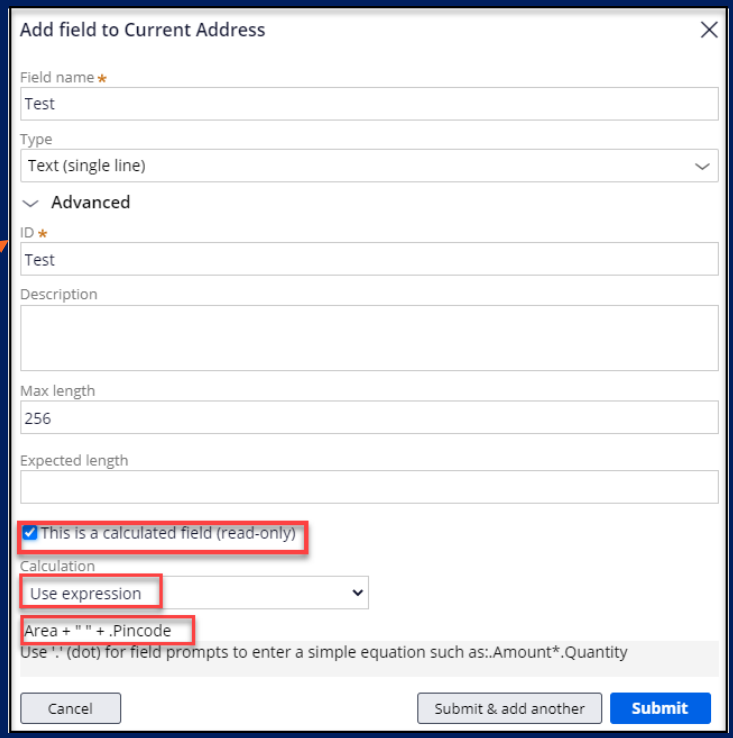
From App Studio:

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

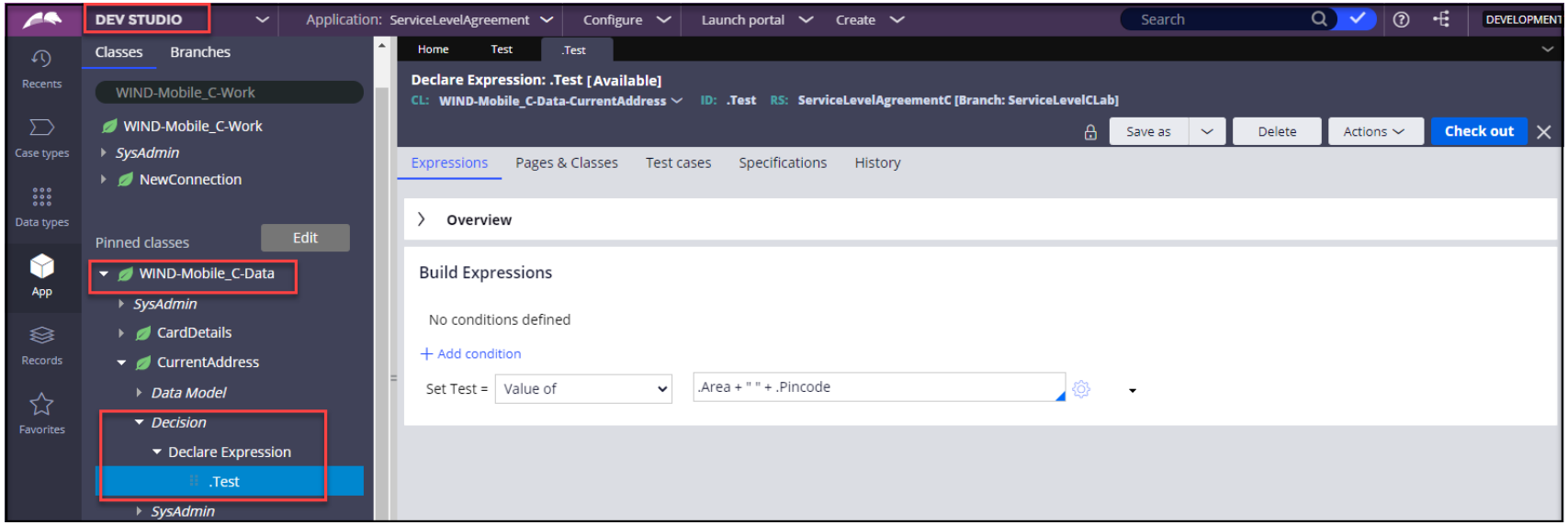




Declare Expression

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.

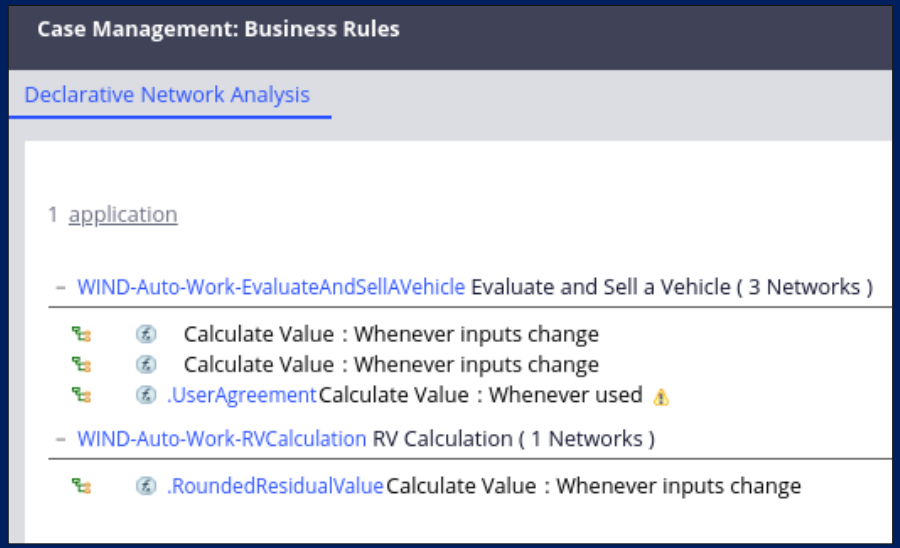


Declarative network display

Navigation

From Dev Studio

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



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

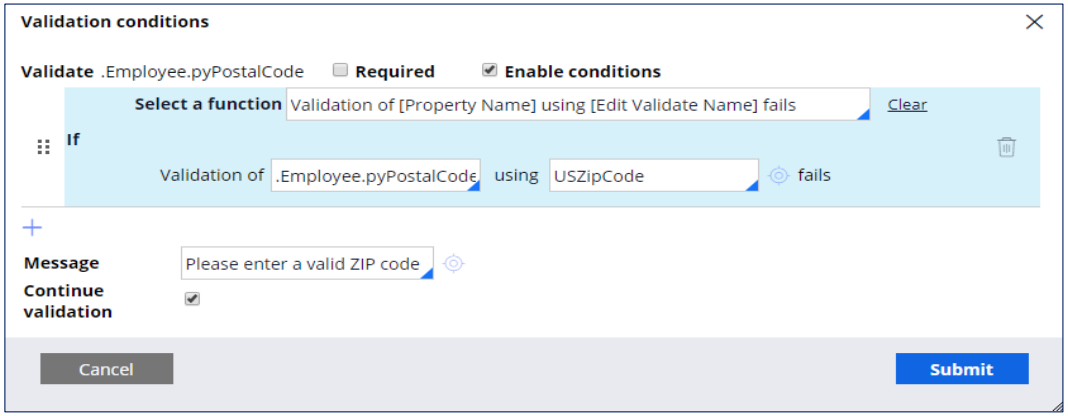
Definition

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



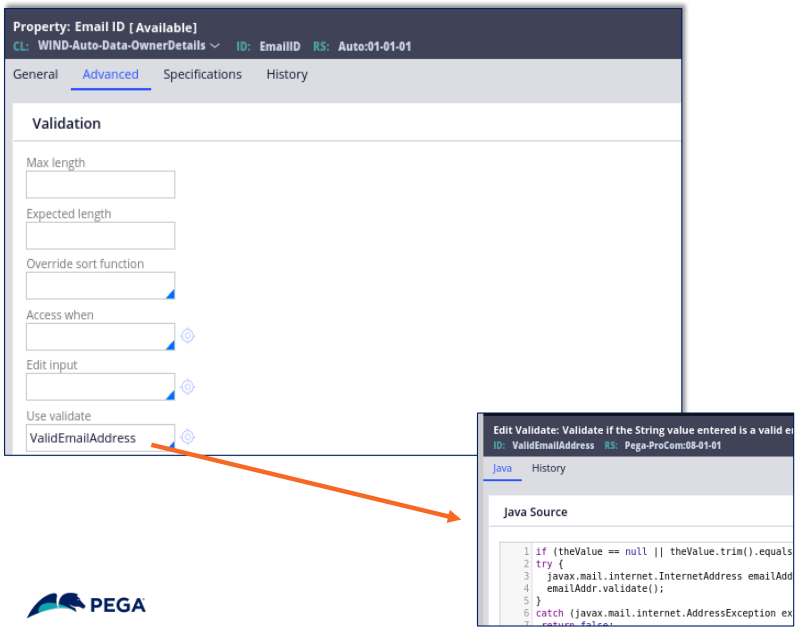
Pega also provides property types, controls, and rules to support validation requirements.

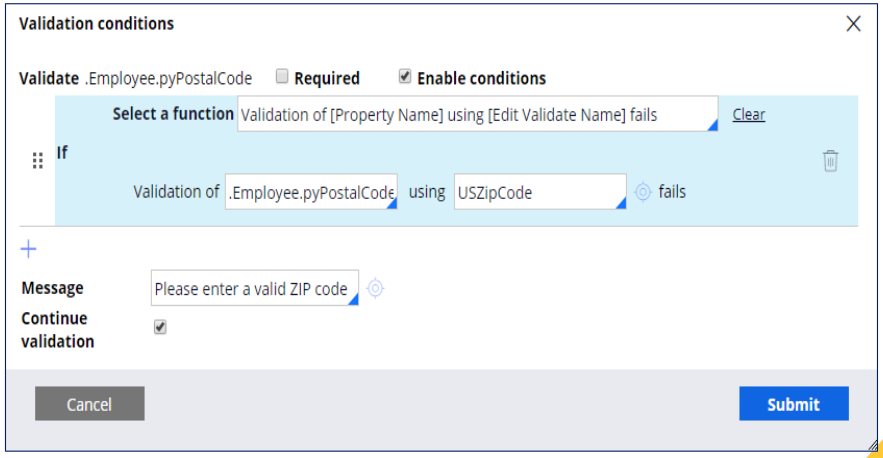
Validation Rule Types

Definition

There are two types of validation rules: **validate** and **edit validate**.

- **Validate** rules compare a property against a condition when the user submits a form.
- **Edit validate** rules test single value, value list and value group properties for patterns. This rule requires knowledge of Java.

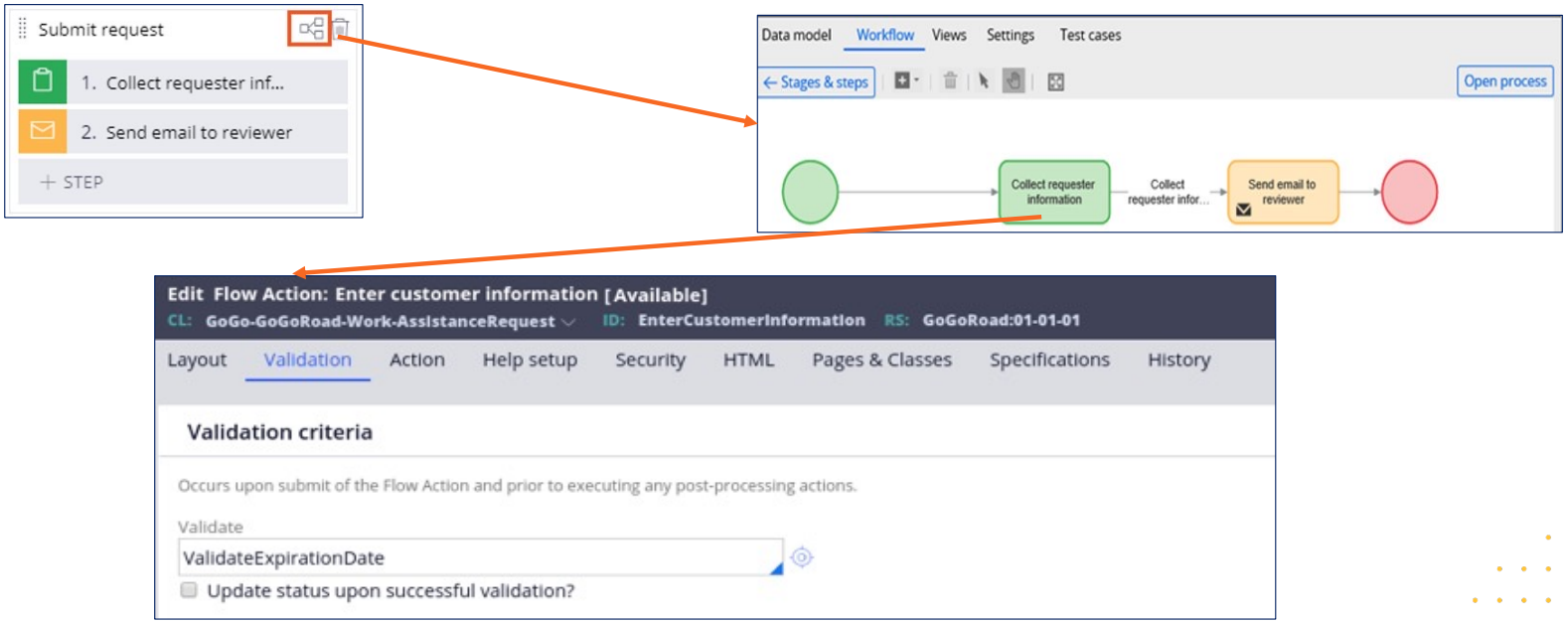




Flow Rule Validation

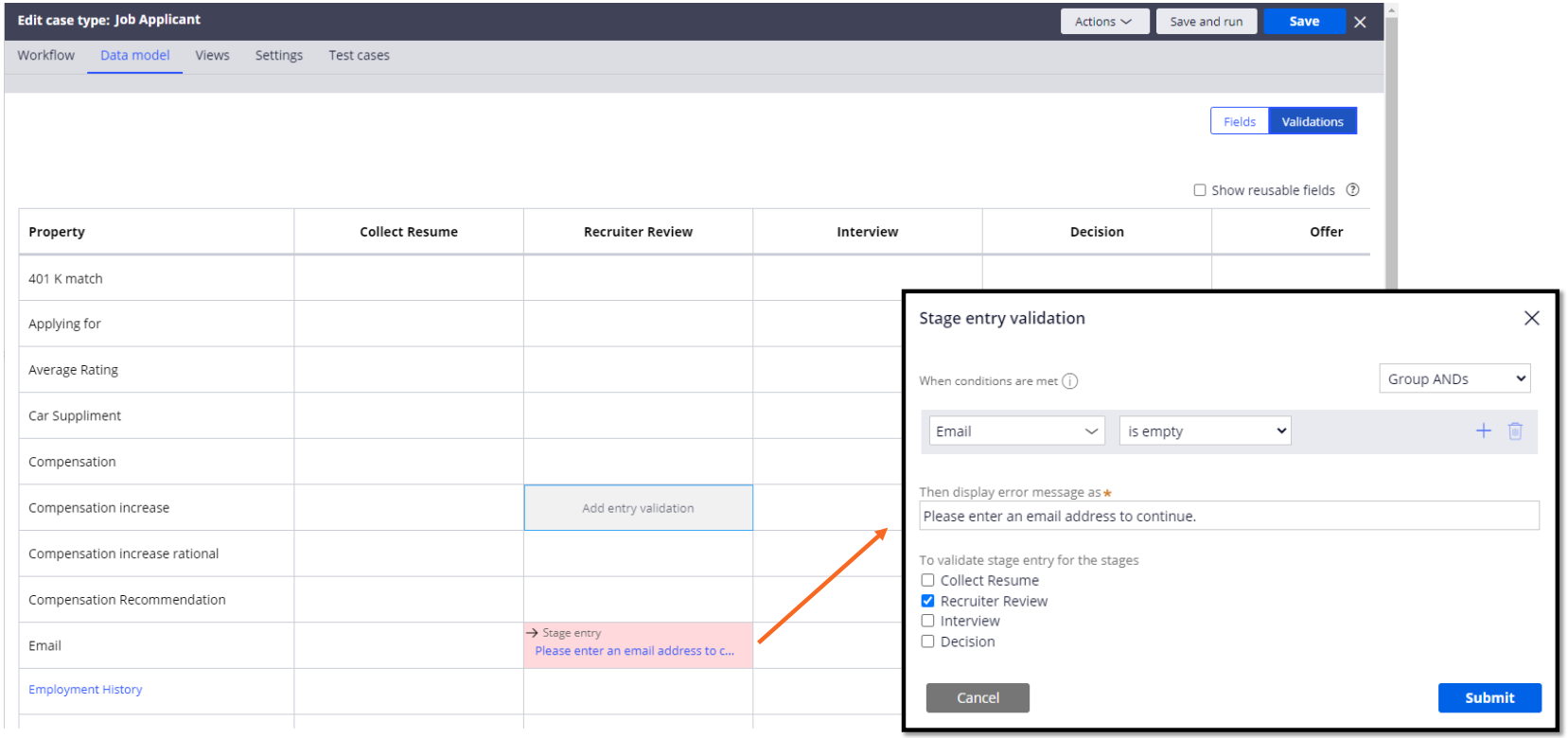
Implementation

- Validate a user form/flow action by opening the Flow rule.
- In the Flow rule right click on the connector and open the flow action you like validated.
- In the Validation tab specify the validation rule you want to use.



Validation – Case Data Model

Case Type > Data model tab > Validations

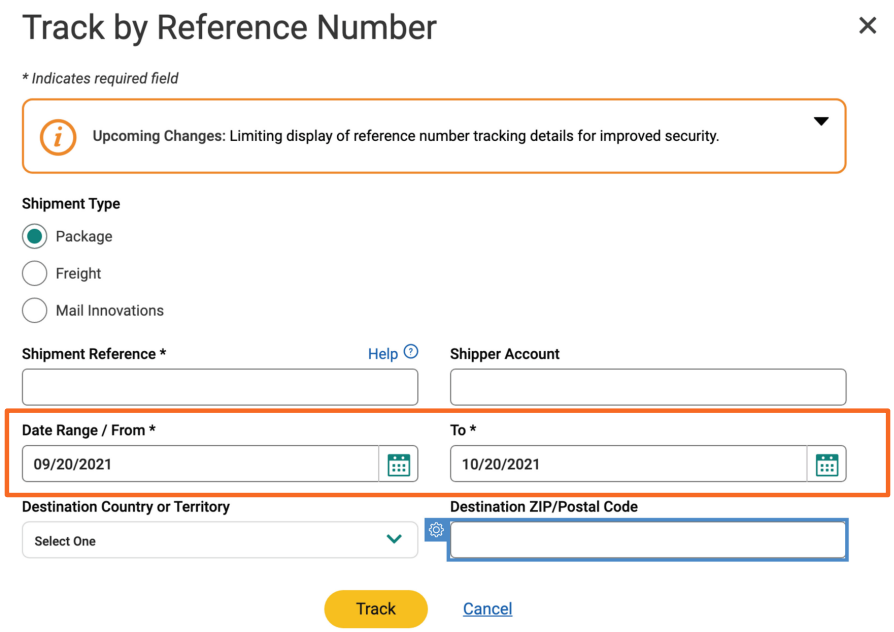


Data Transforms

Use Case

Data Transforms are used to:

- Initialize values in a case type
- Manipulate data between assignments
- Populate property values in other artifacts
- Convert data from one type to another

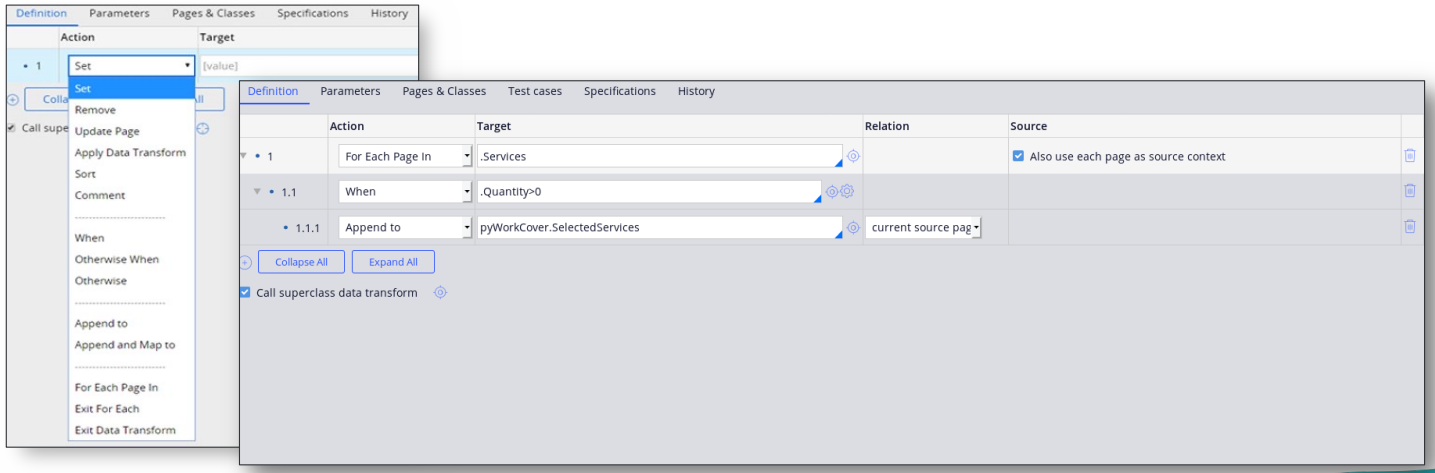


During case processing default values can be initialized. For example, today's date can be prepopulated for the starting date range. The To: date could be populated with today's date + 30 days.

Data Transform Actions

Description

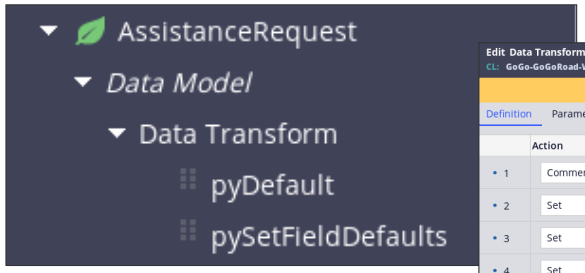
- Configuring a data transform requires specifying an **action** on a property or page and then entering the values for the transform.
- **Actions** are the individual operations such as **Set** a property, **Update Page**, **Append to**, **Remove**, **When** Or a **Page** on the clipboard.
- The **Action** is specified in each row on the **Definition** tab of a data transform.

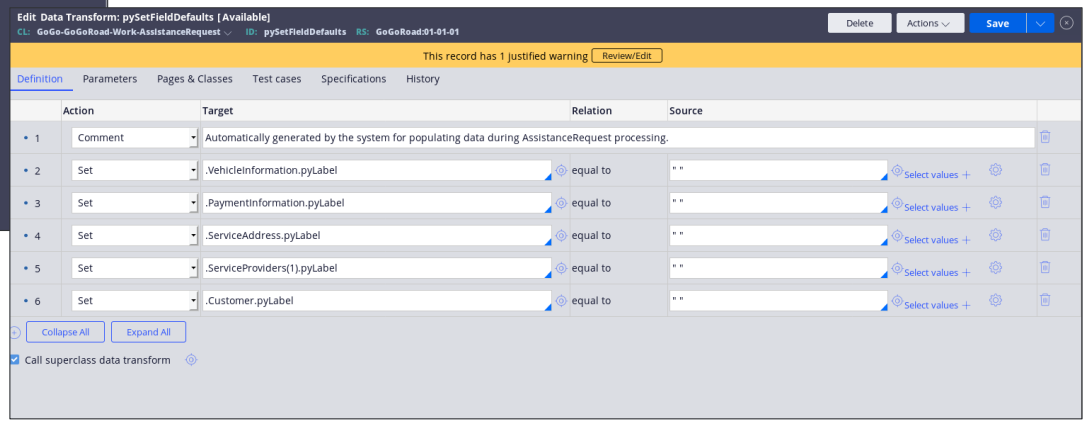


pyDefault and pySetFieldDefaults

Description

- **pyDefault** data transform is invoked when a case is created.
  - It is used to set default values for cases.
- **pySetFieldDefaults** data transform is used to initialize default field values.
  - The case designer wizard will configure it.
- ***pyDefault*** uses the **Apply Data Transform** action to call ***pySetFieldDefaults***.

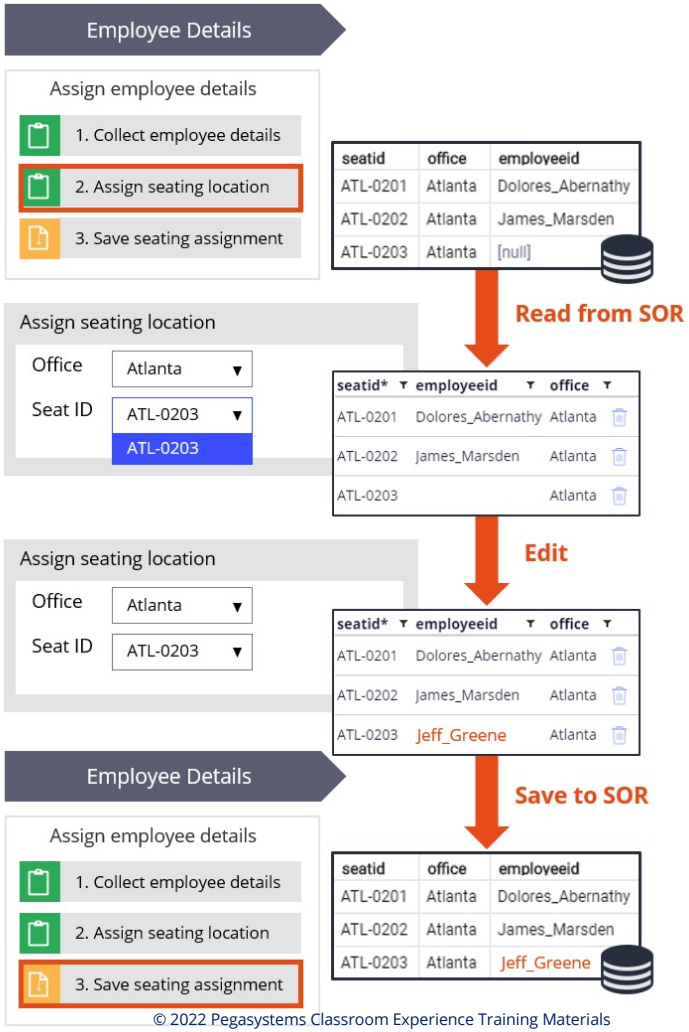




Data page

Definition

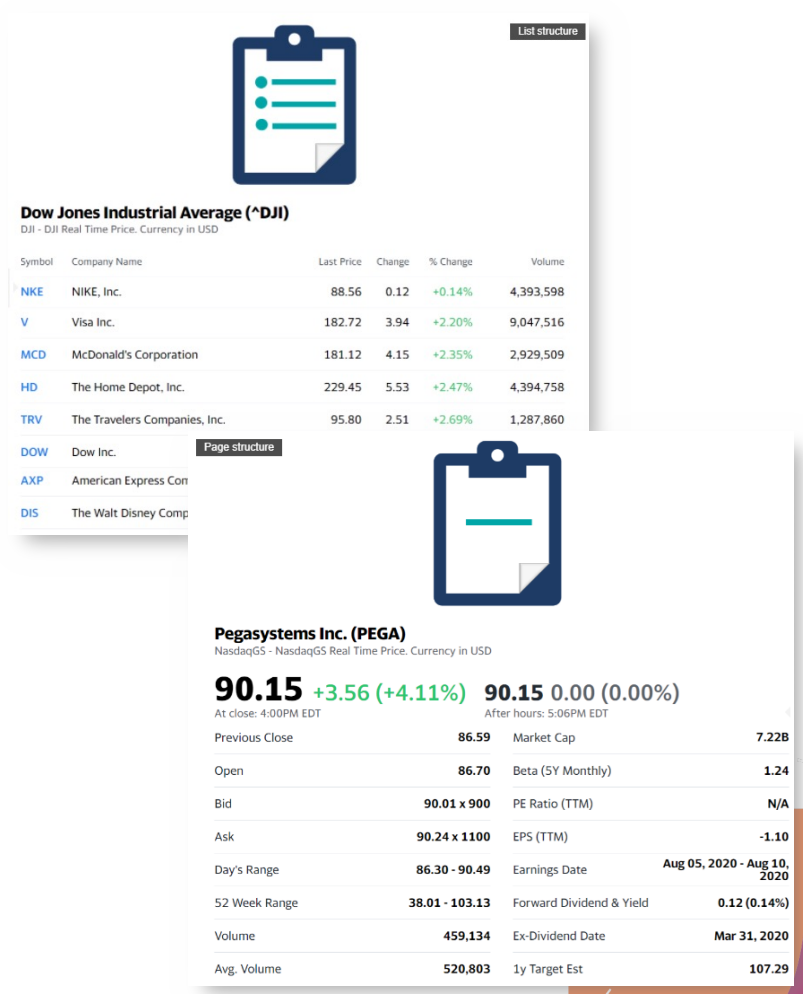
- A mechanism to access data without needing to know the physical location of the data source.
- Defines the data associated with a data object and contains the data object's connection configuration to a system of record.
- Separate the application from the integration with the system of record. Because of this separation, the application can easily adapt to integration changes and access data from a range of sources on-demand.
- Connect a data page of one object type to a data source of another, incompatible type, enabling data virtualization.
- Also referred to as data view.



Data page structure

Description

- The structure of the page determines whether the data page can contain ***one*** item or ***many*** items.
- When using a ***list structure***, the data page embeds the list items in an ordered array named ***pxResults***.
  - To access a specific record within the list, use the syntax DataPageName.pxResults(n), where n is the index of the ordered array that corresponds to the record.
- ***Page structure*** represents a single page or class instance.



Data page scope

Description

- The scope determines the visibility of the page contents within the application.
- A data page scope can be node, thread, or requestor.

Thread

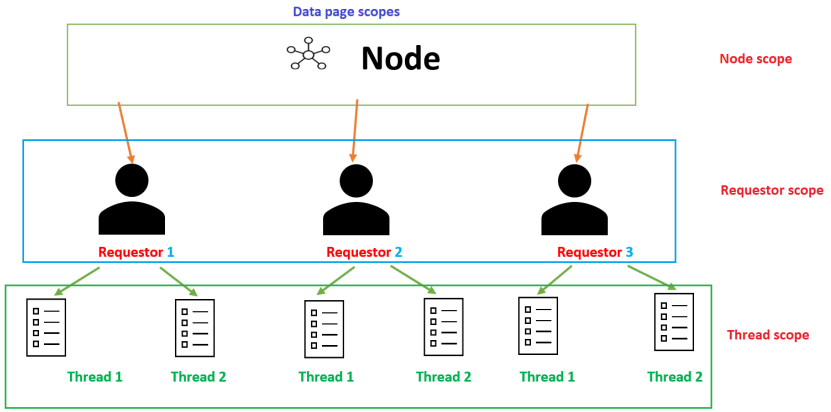
- The page is created in a single requestor thread and can be accessed as often as needed by processing in that thread.

Requestor

- The requestor can access the page(s) loaded across all threads.

Node

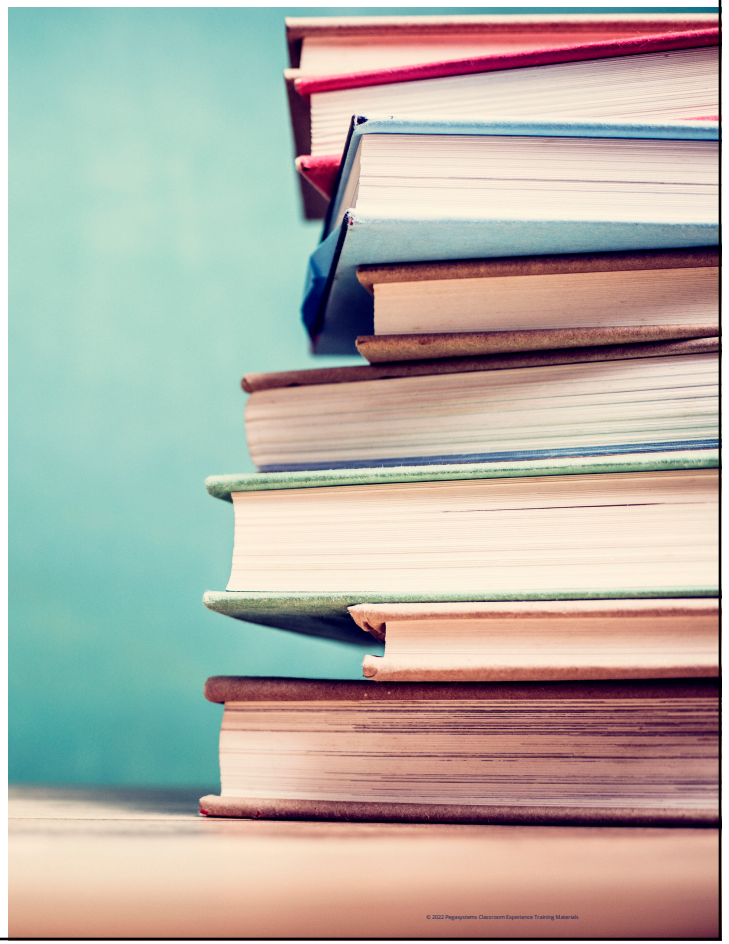
- Any requestor executing on the current node can access the pages.



Data page related terms

Definition

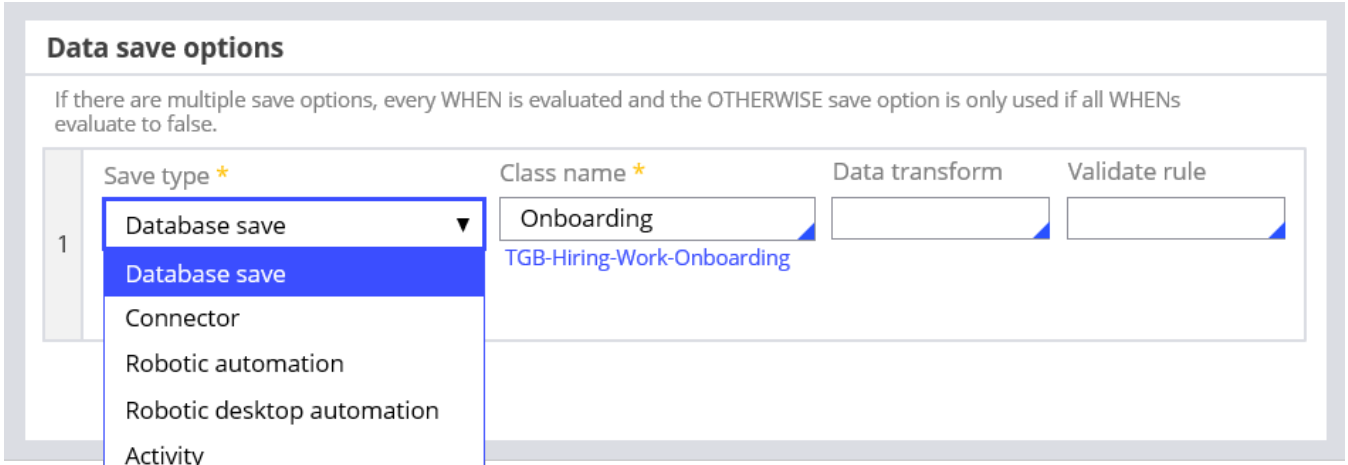
- **Data Record** – values for the fields associated with a data type stored outside the case.
- **Data Type** – define an object and contain all the fields to describe its structure.
- **System Of Record (SOR**) – denotes where data associated with a data type is located: no SOR, Pega or External.
- **Local data storage** - storage of data for a data type, without having to create or maintain database tables.
- **Data reference** - a field configured to display on a form that at run-time has a list of data records or data type instances.



Data save options

Definition

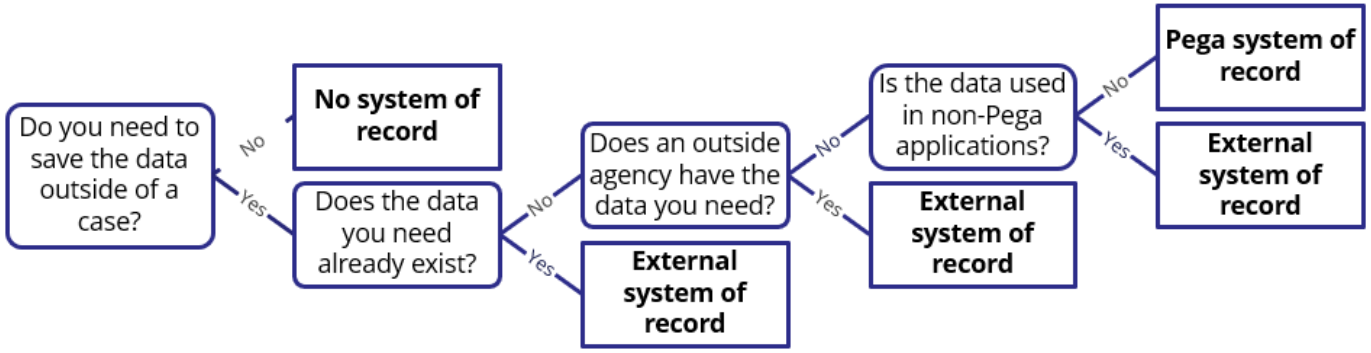
- The **data save options** for a savable data page details how saves are performed. You specify the data save plan in the **data save options** section of the savable data page.
- You can specify multiple save options, where each save option is associated with a when rule that determines when that save option is used.
- The data save options include: Database save, Activity, Connector, Robotic automation, and Robotic desktop automation.



Sourcing

Description

- Data objects can be sourced locally from a Pega Platform system of record, from an external system of record, or not associated with any system of record.
- Determine how to source a data object by considering the questions in the image.



Does the data need to be saved outside of the case?

- ***No, data object with no SOR***

Does the data already exist?

- ***Yes, data object with existing external SOR database***

Does an outside agency have the data needed?

- ***Yes, data object with external SOR connection***

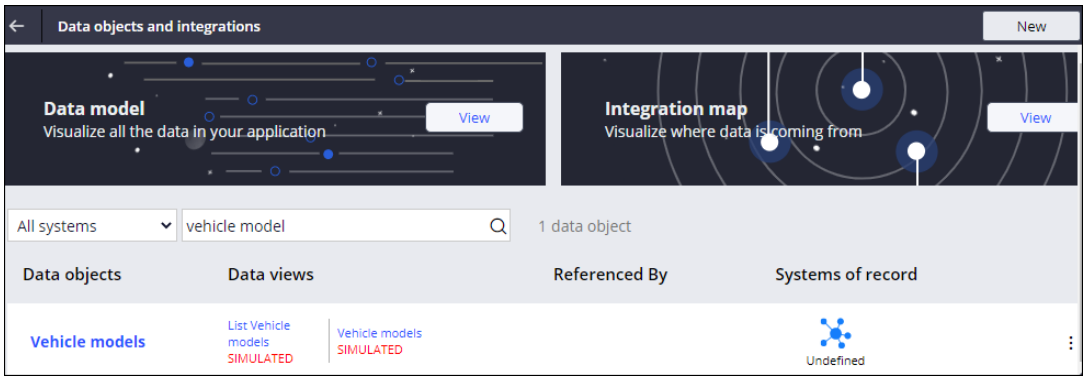
Is the data used in non-Pega applications?

- ***No, data object with local SOR***
- ***Yes, data object with external SOR in a third-party application***

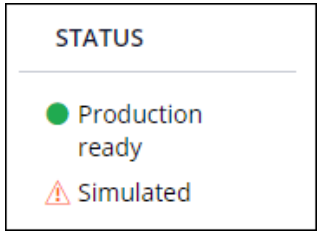
Simulated External Data Source – App Studio

Description

- In **App Studio**, when you create a data view, you identify the source of the data.
- The same data views are created regardless of whether you simulate the source data or connect to a system of record (actual data).



- Use Dev Studio to get an overview of simulated data pages in your application.
- In **Dev Studio**, click **Configure > Data Model > View external data entities** to open the landing page and get an overview of simulated data pages in your application.
- Source systems marked with a green dot are production-ready. Source systems marked with an orange triangle are simulated.



Simulated External Data Source – Dev Studio

Description

- Use Dev Studio to identify the source data directly on the data page.
- You select the **Simulate data source** option on the data page to simulate the data source.
- Then, enter or select the name of the system to use for simulated data.
- You can simulate a data page by using a data transform, activity, report definition, or lookup. Selecting simulation disables any data source configured.

Field types

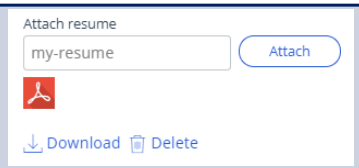
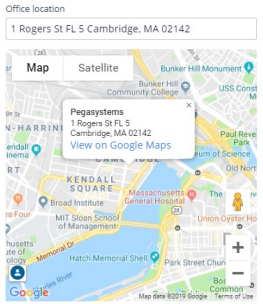
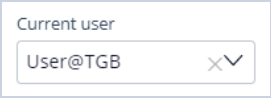
Description

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

Field type	Type of data
Email	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.
Integer	Positive and negative whole numbers, including the value zero (0).
Phone	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.
Picklist	A list of predefined values.
Time only	UTC (Coordinated Universal Time) value normalized to Greenwich Mean Time (GMT).
URL	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.	
Location	Address input or automatic geolocation.	
User reference	Enter or select a user ID that exists in the system.	

Field types and data object location

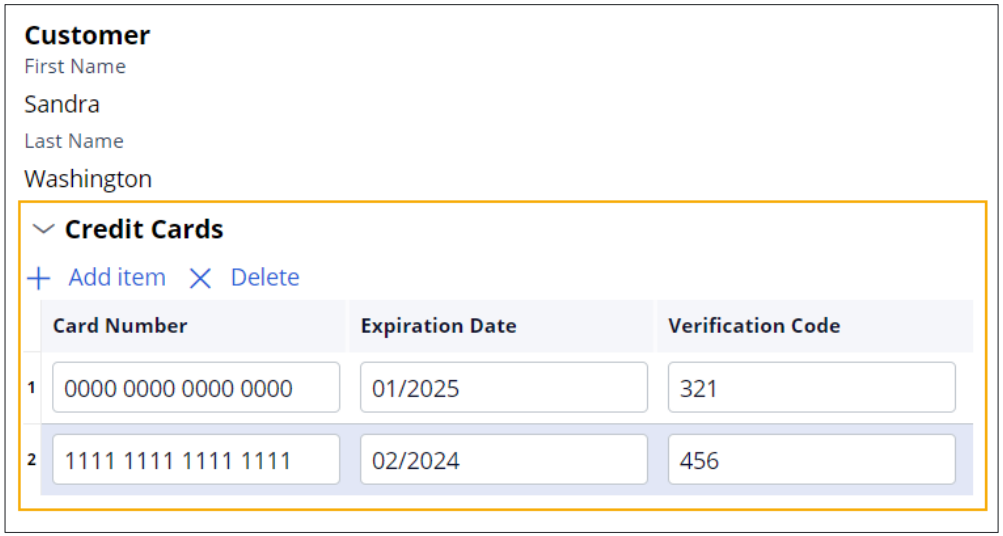
Implementation

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()



Data and Integration - Referencing a Property

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

Value mode properties	**.OrderDate**	• A single value property named ***OrderDate***
	.Phone(Mobile)	• An entry in a value group property, such as the mobile phone number where Mobile is the group subscript
	.DiscountCode(1)	• The first entry in a value list property, such as one of the discount code
Page mode properties	**.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

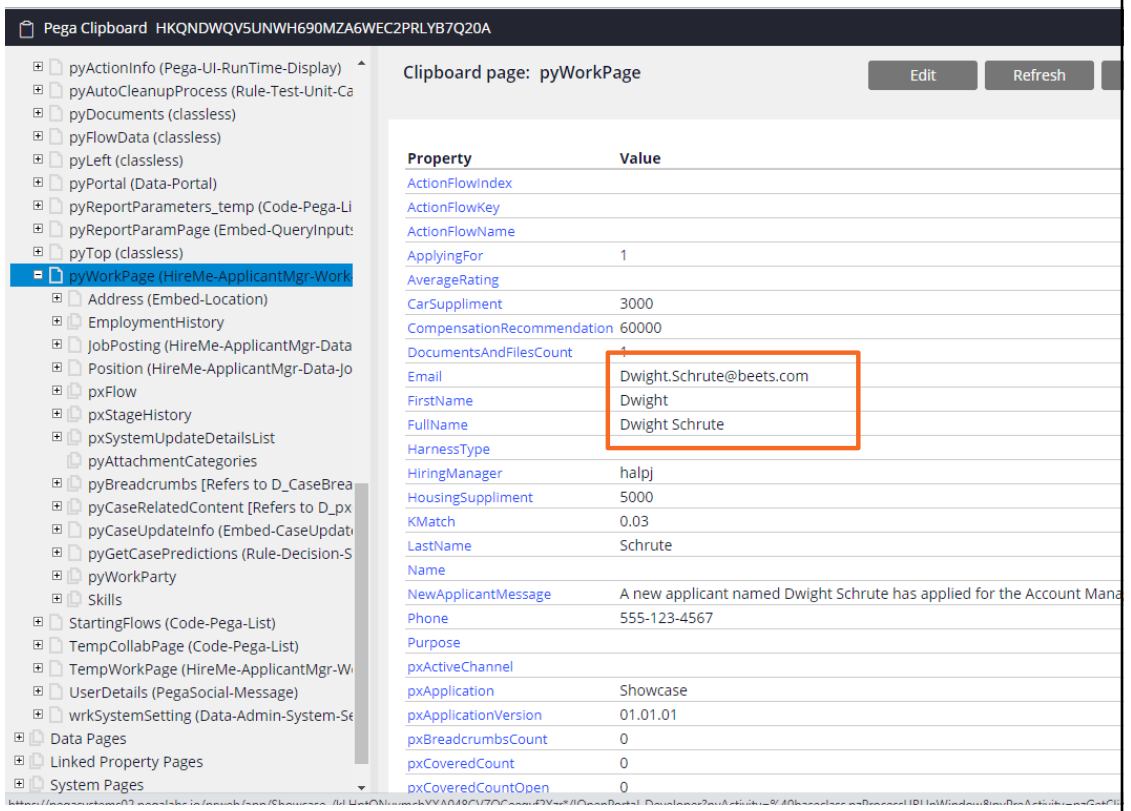
- Pega comes with a set of standard property rules.
- The prefix identifies how the standard property can be used.
- System property names start with ***px***, ***py*** or ***pz.***
  - **Px** - properties that users see on a form, but cannot directly enter or change values, i.e. ***pxCreateDateTime***.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

pxUpdateSystemID	pega
pxUrgencyPartyTotal	0
pxUrgencyWork	10
pxUrgencyWorkClass	10
pxUrgencyWorkSLA	
pxUrgencyWorkStageSLA	0
pxUrgencyWorkStepSLA	
pyAgeFromDate	20200708T164935.621 GMT
pyCancelLabel	Cancel
pyConfirmationNote	pyStepRoutedConfirmation
pyCustomerSatisfiedTimestamp	
pyDocumentTitle	
pyElapsedCustomerUnsatisfied	6.0
pyElapsedStatusNew	6.0

Clipboard tool

Definition

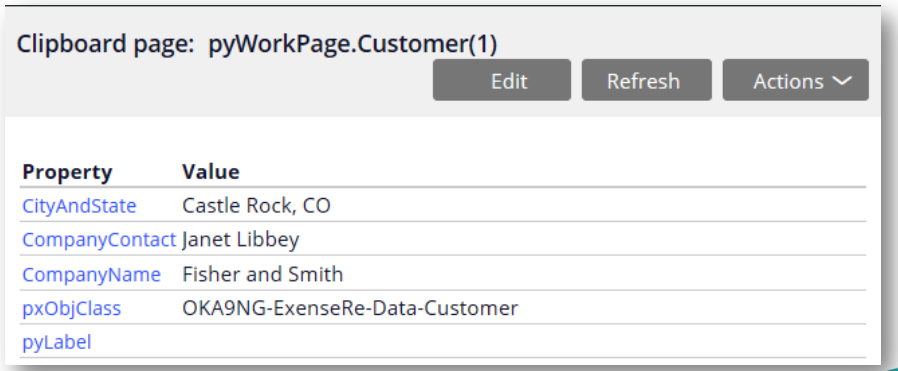
- A debugging and troubleshooting aid for application developers
- Acts as a temporary memory for both the case and the entire application
- Allows an application developer to examine a snapshot of the data structure and contents and sometimes change values
- Has a collection of pages containing properties
- The clipboard resides on the server



pyWorkPage

Description

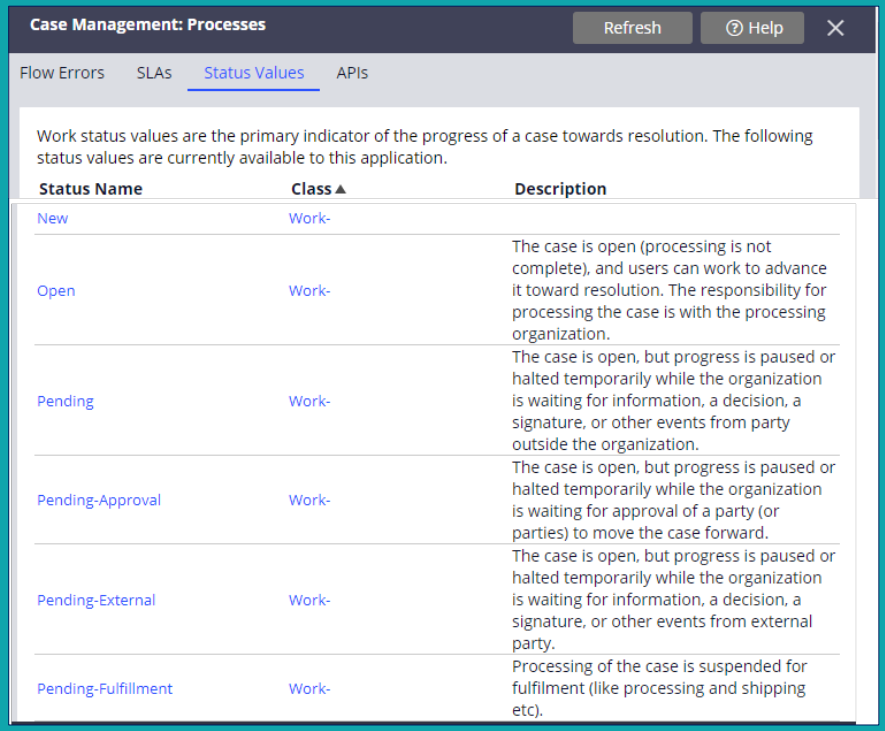
- **pyWorkPage** is a top-level page on the clipboard that stores case data.
- Whenever a user is working on a case, the case will ALWAYS, reside in **pyWorkPage**.
- An **embedded page** is a page that is NOT a top-level page, that typically stores data describing a data type.
  - For example, the .Customer(1) page is an embedded page
- The **pyWorkCover** is a page that contains the case data for a parent case when the associated child case is being processed in **pyWorkPage**.



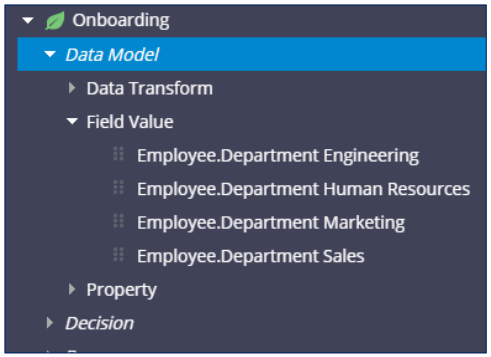
Field Values

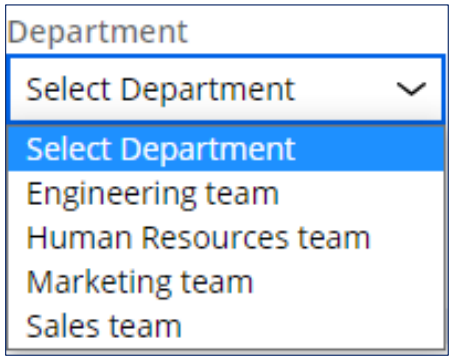
Definition

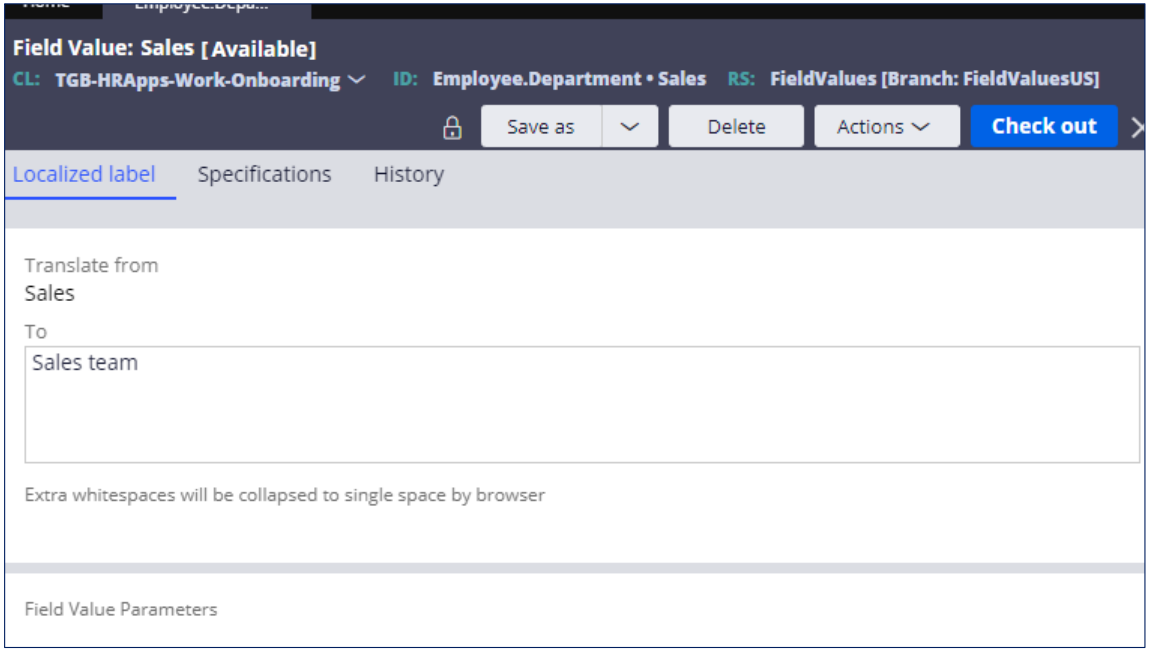
- **Field values** provide a method for defining allowed values for properties.
- Field values are used if the list of allowed values is large, expected to change frequently, or specific for each case type.
- For example, case instance status is set using the property ***.pyStatusWork***. The list of allowed values for setting ***.pyStatusWork*** is defined by using Field values.



Configuring a Field Value



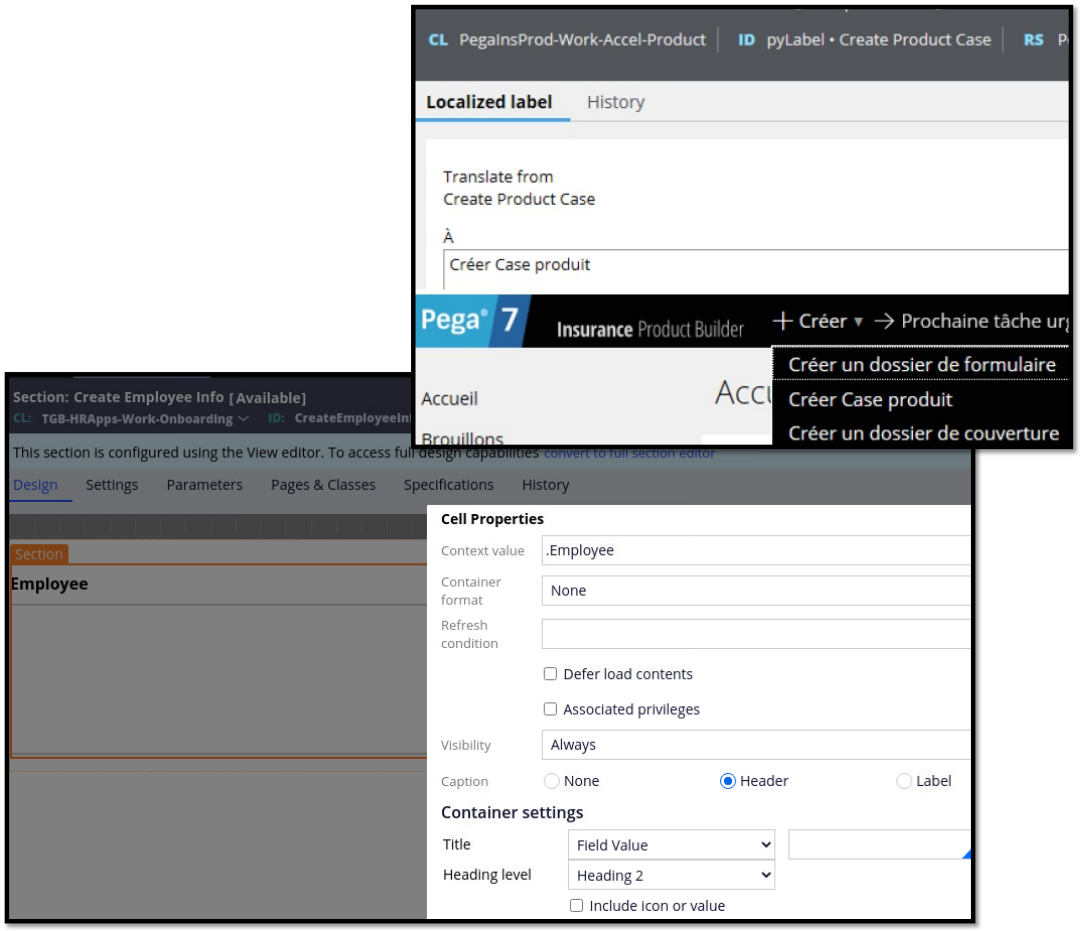




Localization

Implementation

- Localization wizard identifies field values and text strings used in your application ruleset user interface rules.
- For localized text to be displayed in the menu, create field values for .pyLabel property.
- Section header names with field label values can use Field values instead of text.



Configuration sets and settings

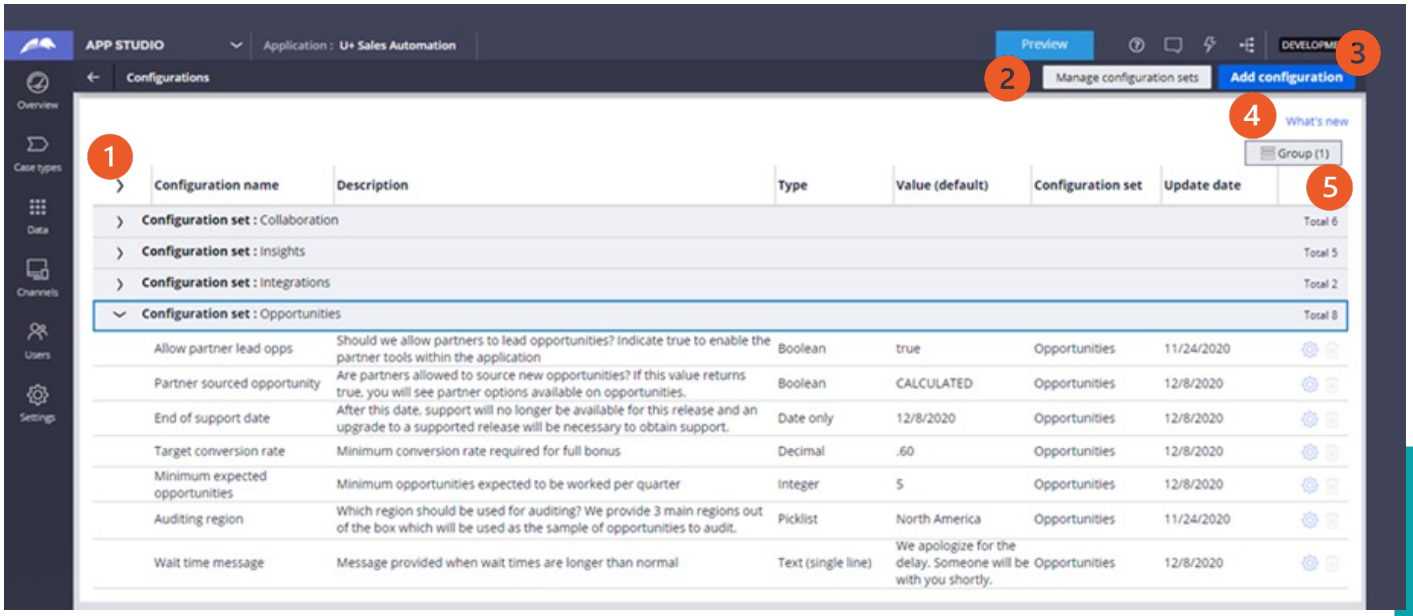
Definition

You can use Configuration settings in the following ways:

- **Control the use of features in an application** - When a feature has dependencies, you can use Configuration settings to turn features off until the dependencies are met.
- **Determine which process in a flow should be followed** - If approval is needed based on a monetary limit.
- **Control UI experience** - When an agile development methodology is used, process changes are introduced with each release. You can use a Configuration setting to control the display of instructions for a process change in an existing process or for new processes. Once users adopt the process, you can update the Configuration setting to hide the additional instructions.

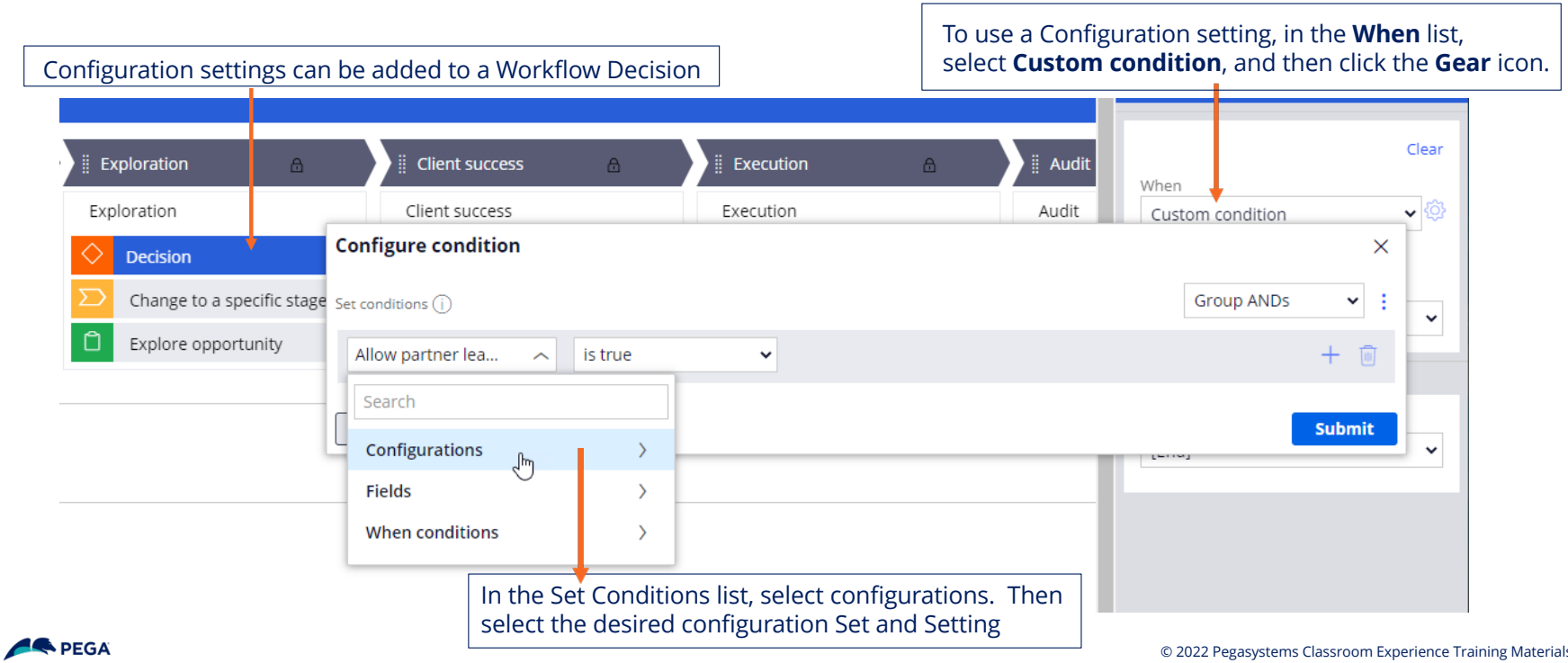
Configurations Landing Page

1. Grouping
2. Manage Configuration Sets
3. Add Configuration
4. See What is new
5. Group



Referencing Configuration settings

- Configuration settings can be accessed in an application through Condition Builder.
- Condition Builder provides a menu to select a configuration set and a configuration setting.



Data and Integration 23% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=96b6227657fc0a9b

User Experience 12%

Exam Blueprint

User Experience 12%

- Customize user interface elements, dashboards, portal content
- Configure action sets
- Customize form appearance, visibility settings, controls
- Add and remove fields
- Group fields in views
- Display list data in views; configure repeating dynamic layouts
- Localize application content
- Enable accessibility features in an application

User Interface Concepts

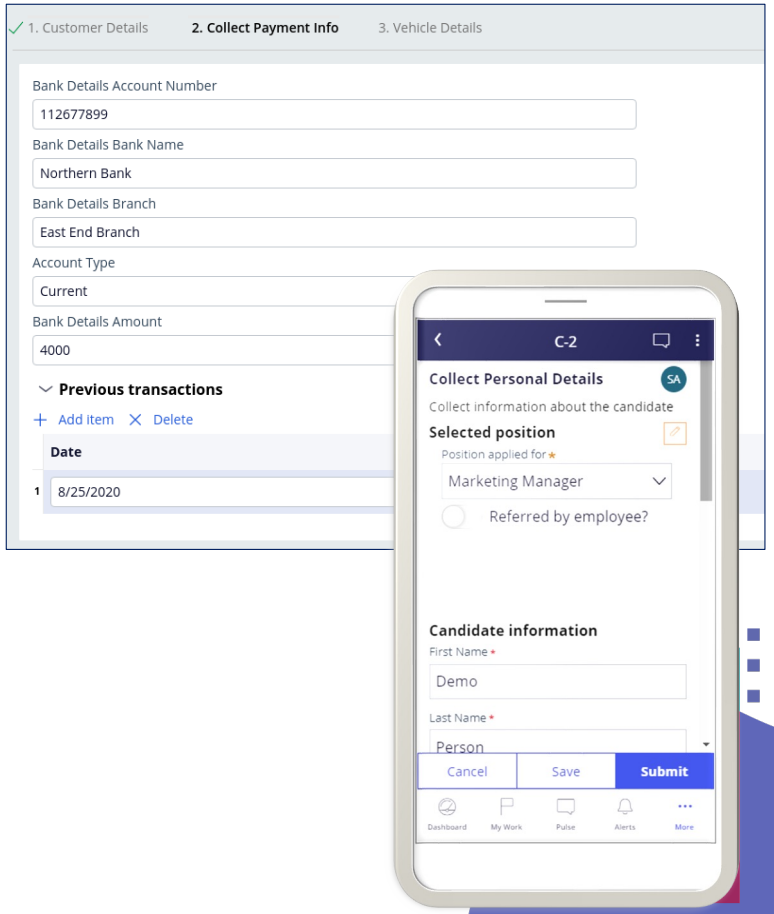
Description

- 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

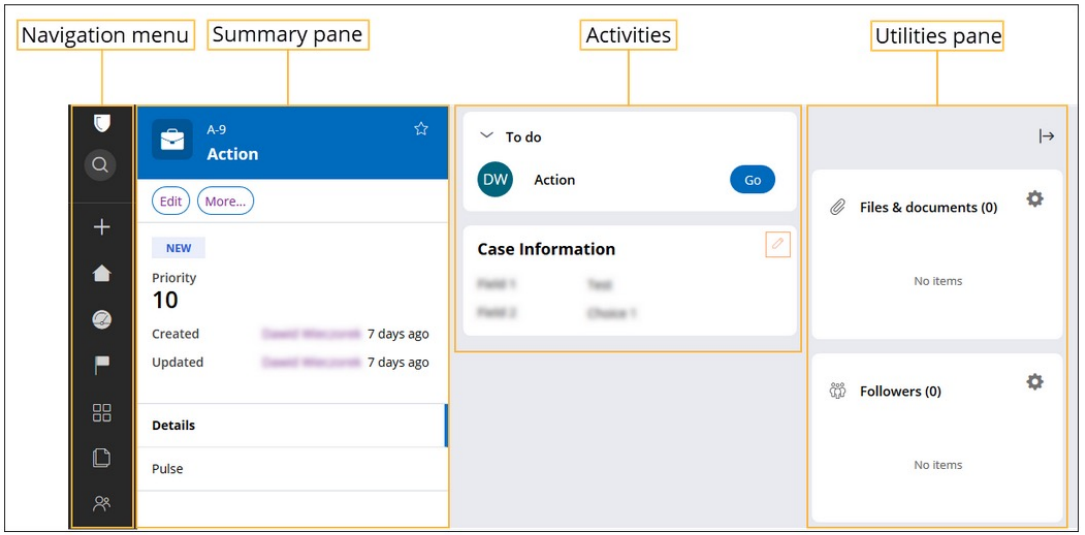
- Users of Pega applications can interact with the application through a web browser, tablet or mobile application.



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

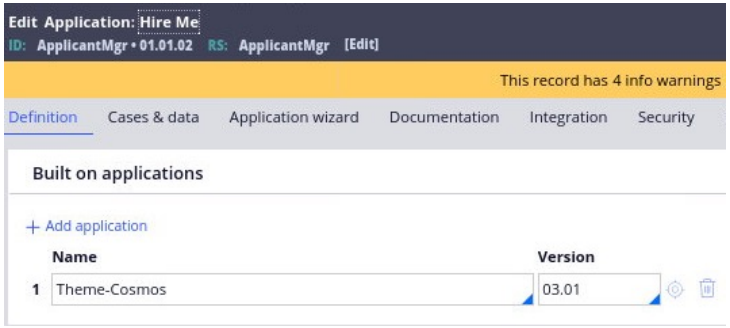


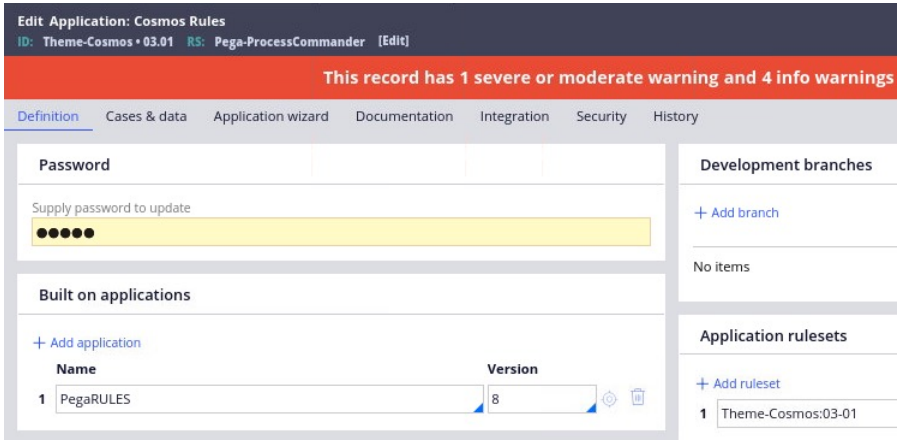
An example case created in a Cosmos-based application

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

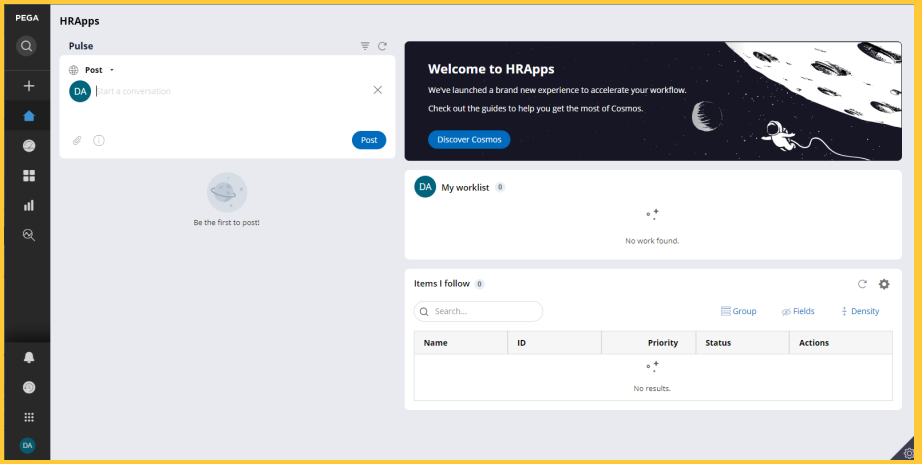




Portals

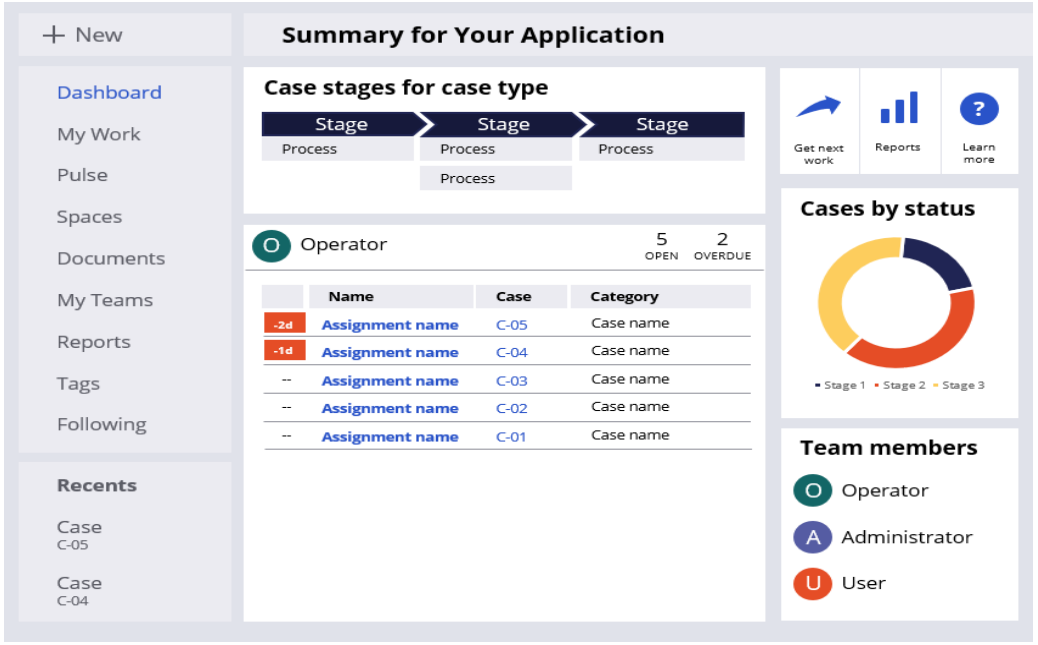
Definition

- 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.



Pages

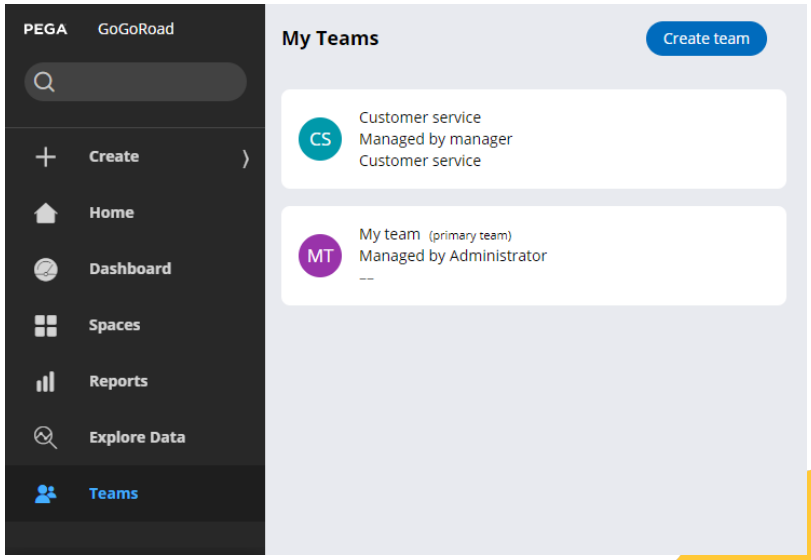
Definition

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

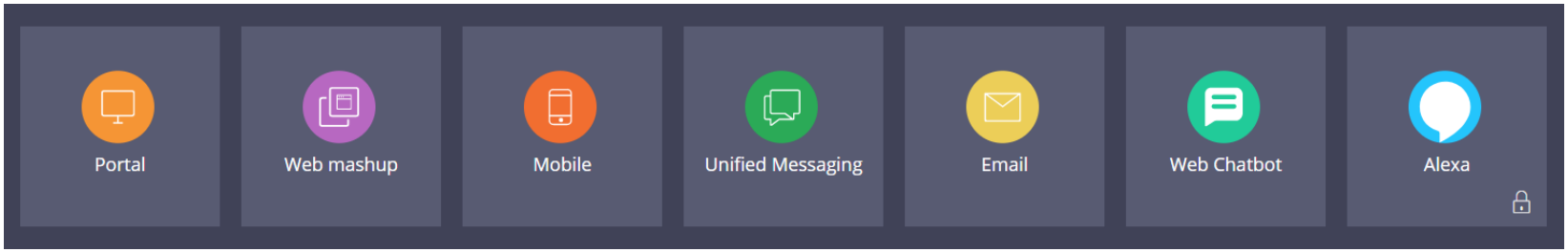
- **Landing pages** are created to build an application that matches the needs of your users and consists of fields, control and resources presented as images and text.
- **Custom pages** are Pega's out-of-the-box landing pages that are built as part of the application and are not customizable by end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

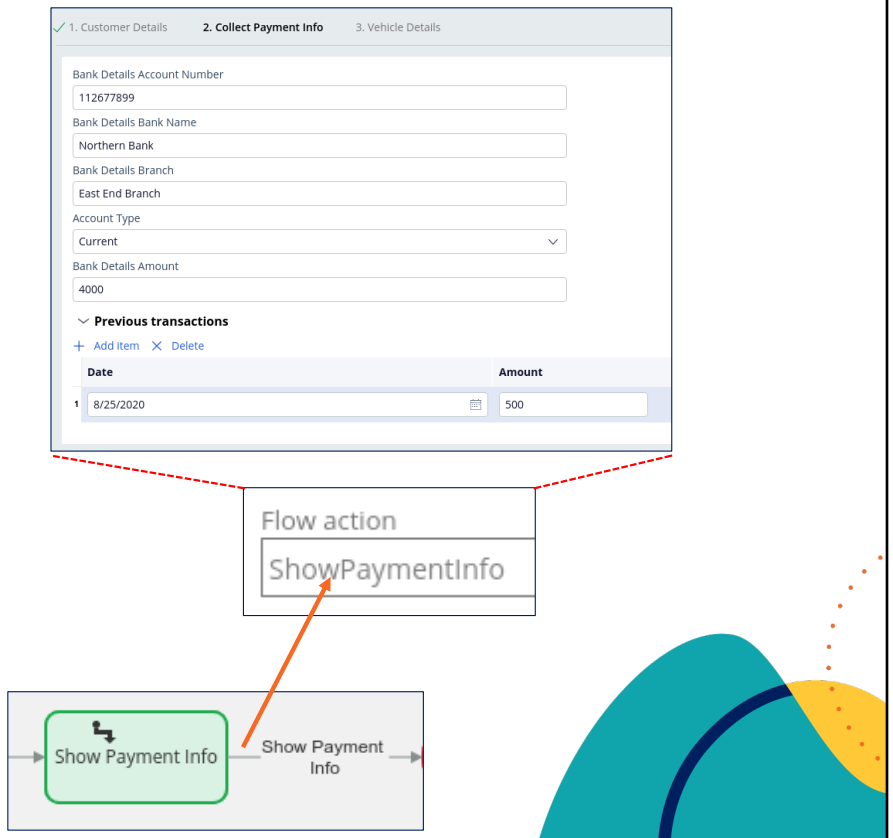


- New components can be downloaded from Pega Marketplace.
- The Lock icon on a channel indicates that the channel component is not yet added to the application.

View – User Interface

Definition

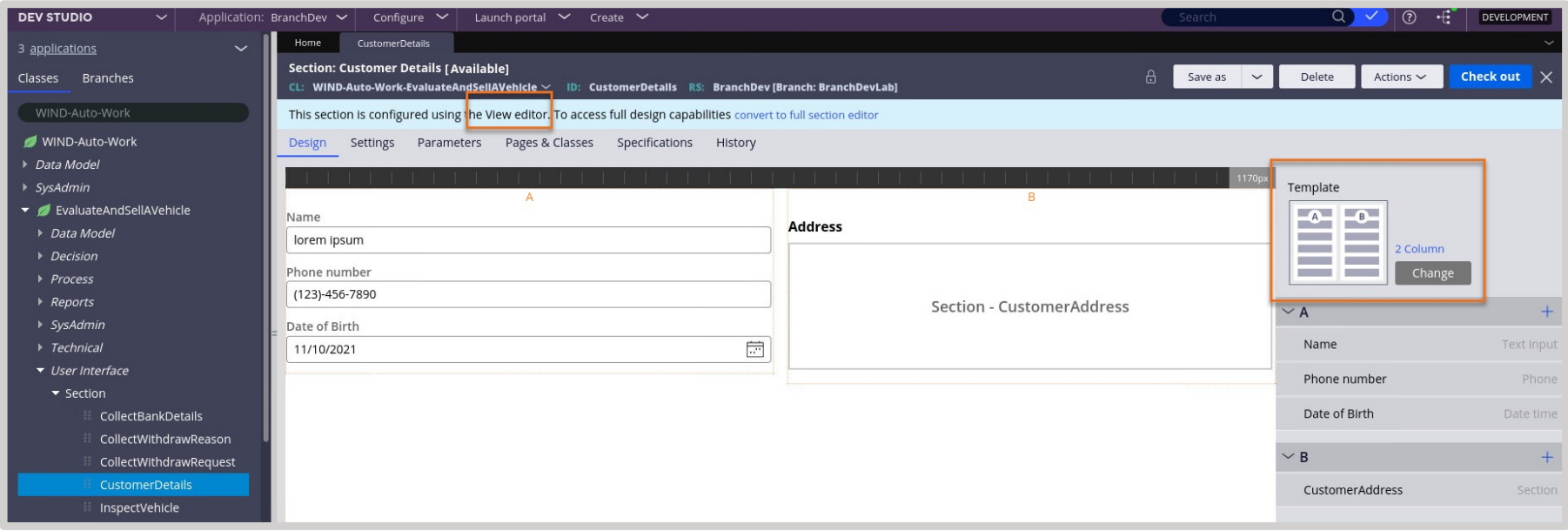
- A **View** is associated with a collect information step (assignment shape) in the flow rule of a case
- A **Flow Action** is associated with the Connector leaving the shape in the Main or Sub flow
- A **Flow Action** is associated with the Assignment shape in a Multi-Step Form
- A **View** may be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



Design Templates

Navigation

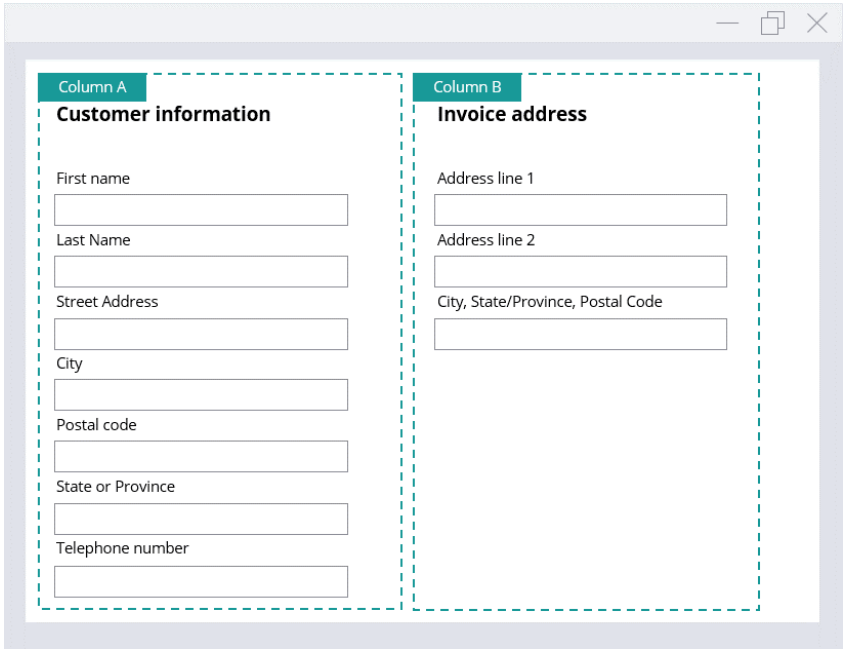
Dev Studio > App Explorer > User Interface > Section > open section > View Editor > Template

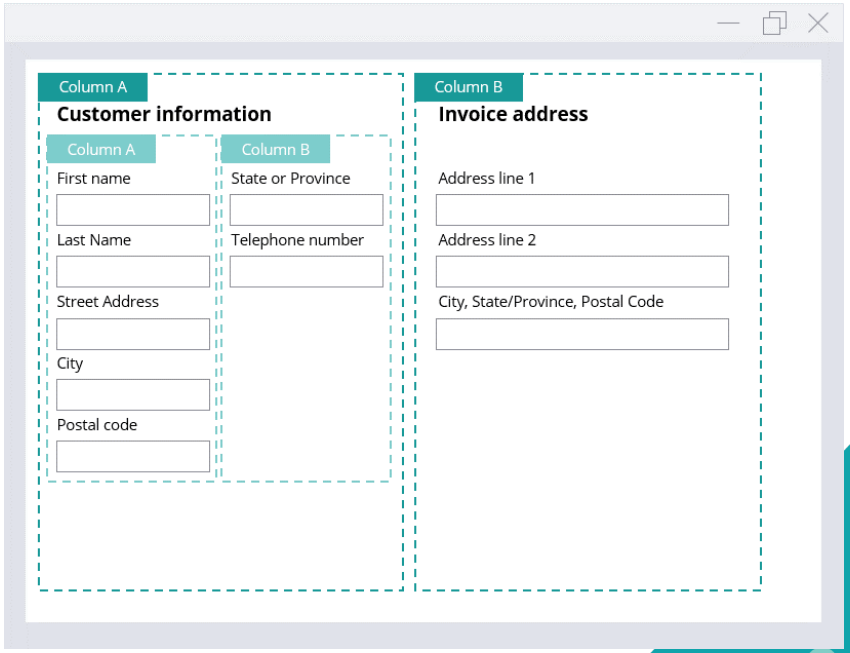


Design Templates

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.

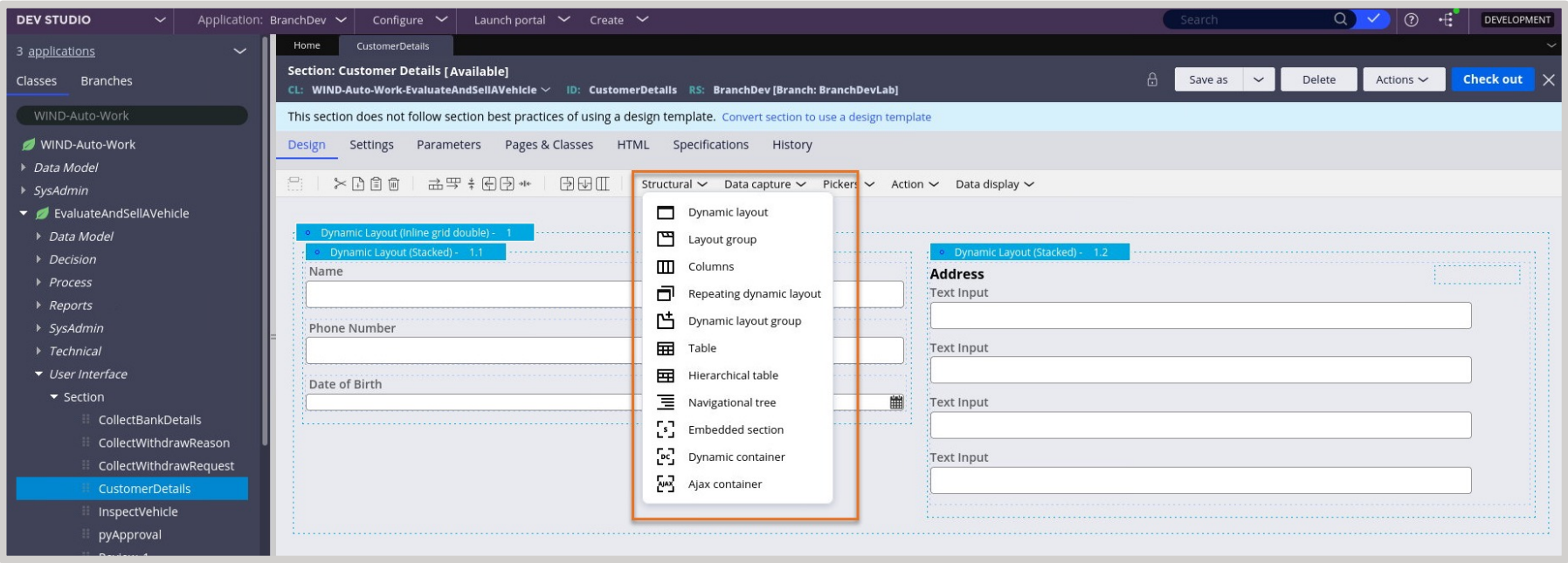




Layouts

Navigation

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

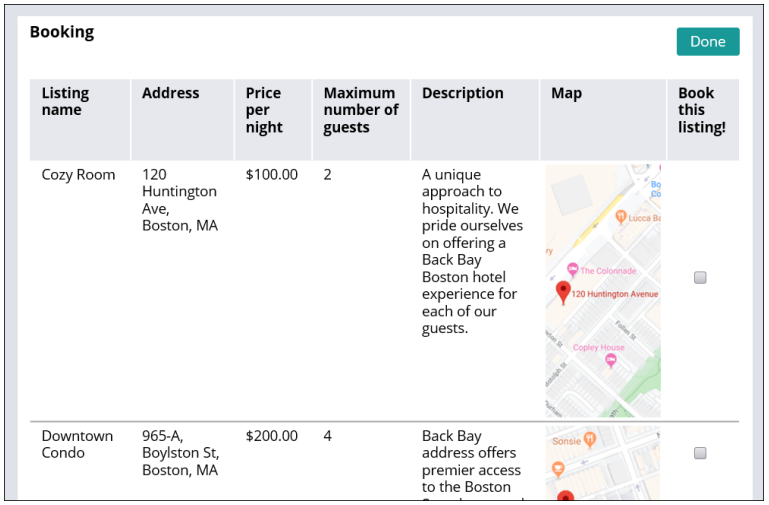


Repeating Layouts

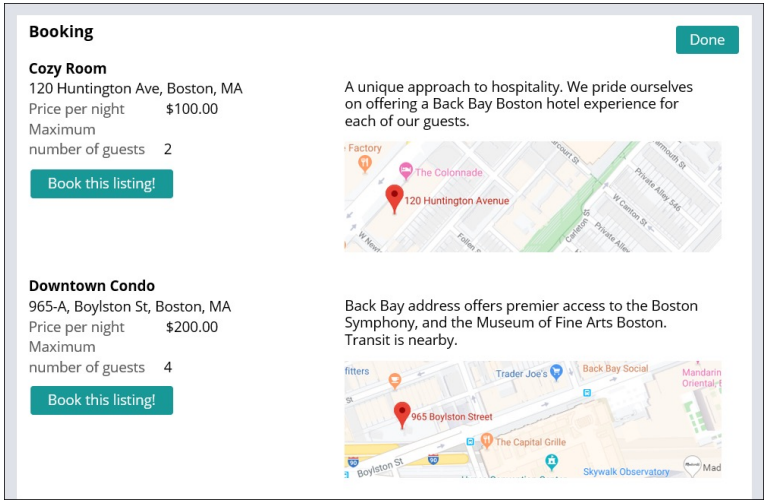
Description

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.



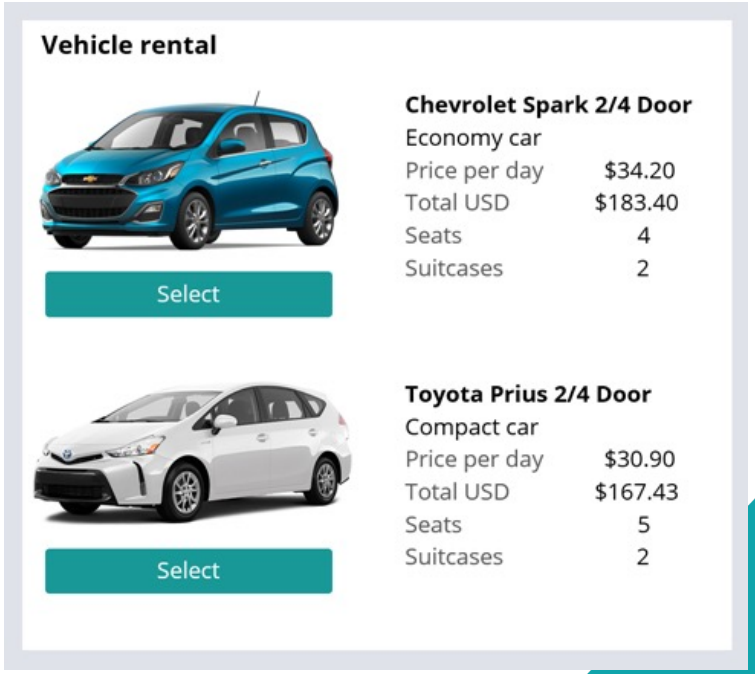
- A **repeating dynamic layout** is useful when you want to group and present content in a nonlinear, more aesthetic format.



Repeating Dynamic Layouts

Description

- The **repeating dynamic layout** option is a more flexible and aesthetic format. Repeating dynamic layouts are better when you want:
  - To use the formatting capabilities of a dynamic layout within a repeating structure
  - To use lists to display a section for each item on the list. Each section can contain any number of layouts
  - To include an image or other graphics, for example, an image of a vehicle or interactive map
- **Repeating layout configuration and behavior at run time**
  - In a repeating layout, when you configure a single layout as a template, Pega Platform™ automatically applies the template to each item in the list at run time.



Responsive UI and tabular data

Description

In Pega Platform™, a table automatically displays data from a field group list. You can configure responsiveness in tables to limit the information that is displayed when the display size changes.

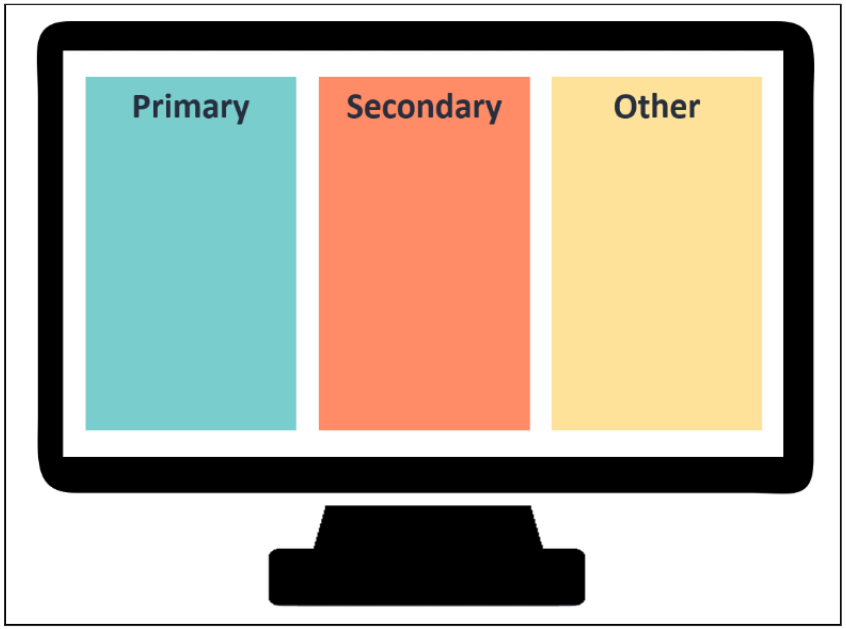
- Responsive table design minimizes horizontal scrolling by removing the less important information from the user interface.
- When designing responsive behavior, it is a best practice that the displayed information aligns with the business needs.
- The order in which information is presented to end users affects the user experience. For example, in the following table, the Part number column and Name column should always be displayed, and the Line total column should always be presented at the end of the displayed information.

Column name:	Part number	Unit cost	Name	Description	Quantity	Line total
	333-88	1.99	InkPen	2-pack ballpoint black	100	199.00

Column importance

Description

- A table has one or more columns, each with a configured importance setting.
- Column importance defines how Pega Platform presents columns as the display size changes.
- The column importance options are **Primary**, **Secondary**, and **Other**; by default, the left-most column is set to Primary.
- You configure column importance in **Dev Studio**.
- In **App Studio** you can set or change a column to primary importance while editing the form configuration at runtime.
- You can set a column's importance to primary by moving that column to the leftmost place in the table.
- When you change the order of columns in a table, Pega Platform automatically sets the leftmost column to primary importance and the remaining columns are set to secondary importance.



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

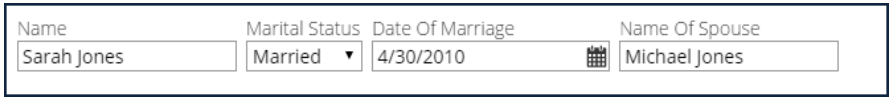
Description

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.



Control Field Display

Description

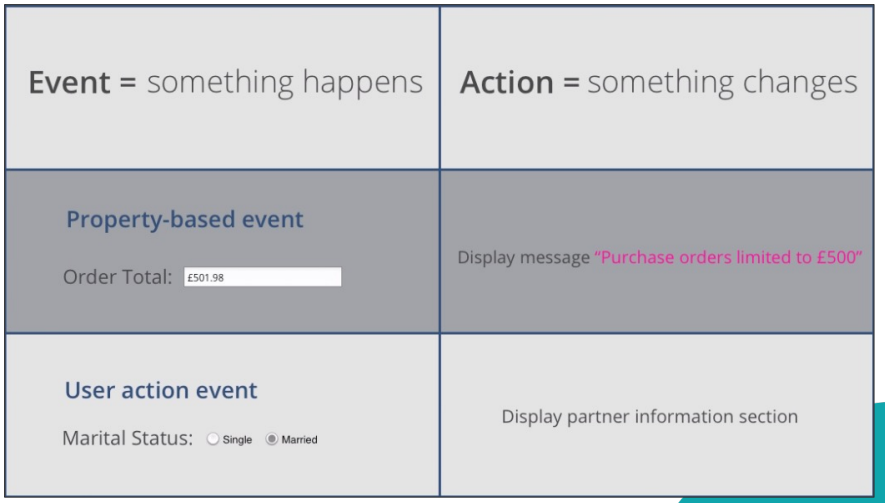
- Pega Platform provides the ability to control how fields are displayed. Application Developers create field attributes for dynamic display.
- You can add conditions to UI elements that affect visibility and user interaction.

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

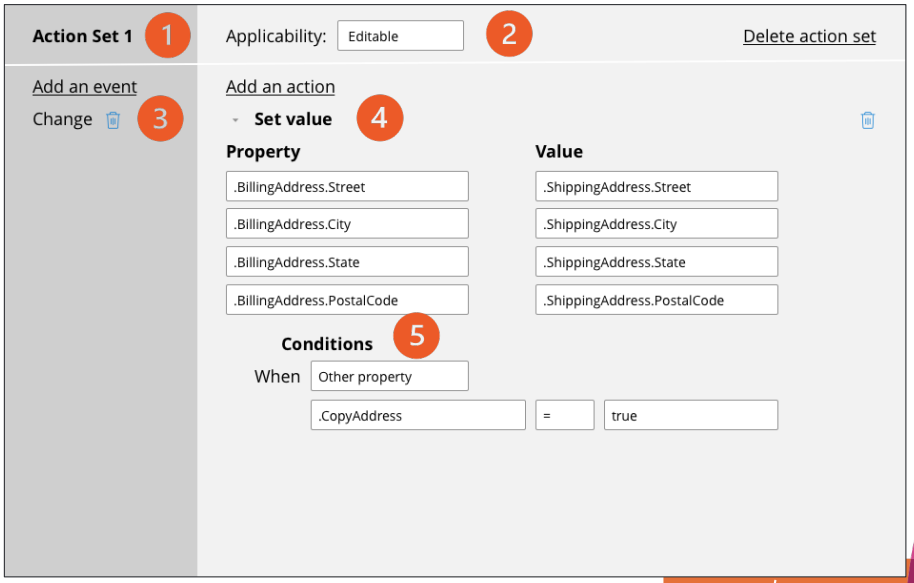
- Dynamic UI behavior is implemented with an event-action model.
- An action set is comprised of an event, an action, and (optionally) conditions.
  - **Event** - a trigger performed by the user such as click, double-click, hover, focus, and keyboard entry
  - **Action** – a response performed by the 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



Action Sets Configured

Description

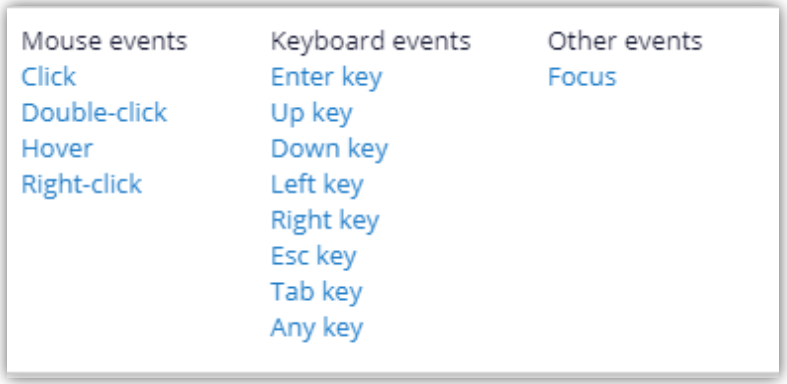
1. **Identifier** – Identifies the action set.
   - Action sets are evaluated in numerical order, starting with Action Set 1.
2. **Applicability** – Determines when the action set is applied.
   - Action sets can be configured for use only when a field is in an editable state, read-only state, or both 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.



Action Sets - Event

Description

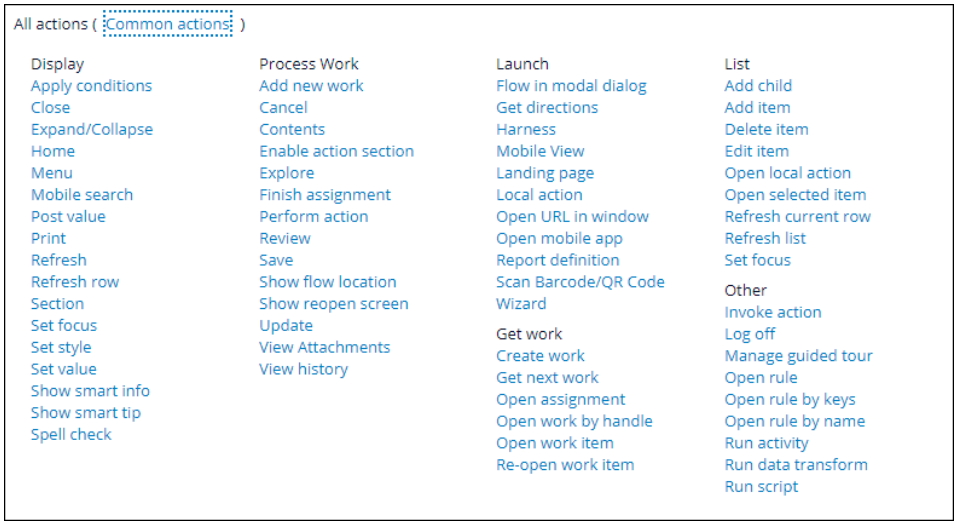
- The trigger for the action set.
- **Mouse events**: events triggered by one of the following actions performed with a mouse or other pointing device
- **Keyboard events**: events triggered by pressing one of the following keys on the keyboard
- **Other events**: events that do not fall into another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



Action Sets - Action

Description

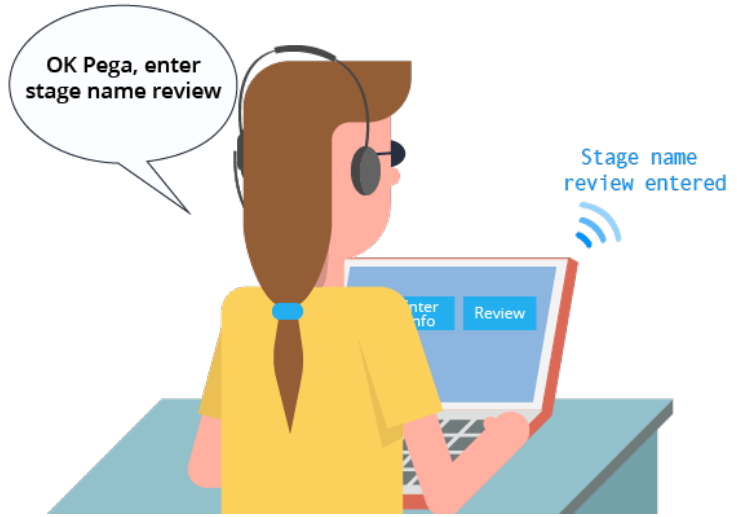
- The action or actions performed when the specified event occurs.
- An action may require that you provide values for needed parameters.
- If an action set lists multiple actions, actions occur in the order listed, starting with the topmost action in the action set.
- Any implicit action triggered by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



Application Accessibility in Pega

Description

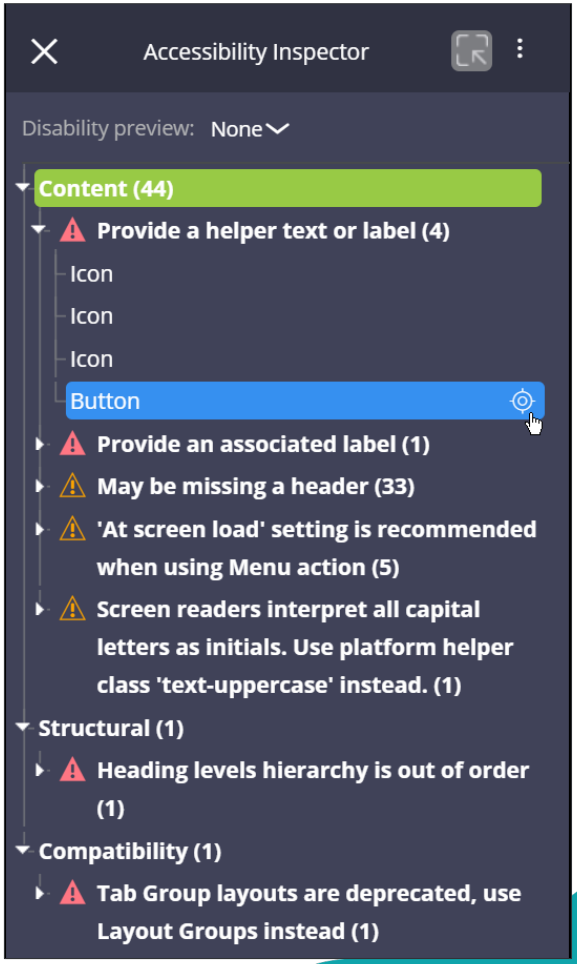
- Pega supports the Web Accessibility Initiative-Accessible Rich Internet Applications (WAI-ARIA) standard.
- WAI-ARIA is a technical specification that defines ways to make Web content and Web applications more accessible to people with disabilities using **accessibility roles.**
  - Accessibility roles are specific attributes applied to user interface elements that enable communication about the UI elements between assistive devices and Pega applications.
  - A common accessibility device is a screen reader (such as JAWS) for the blind, which describes a page of content orally.



Accessibility Inspector

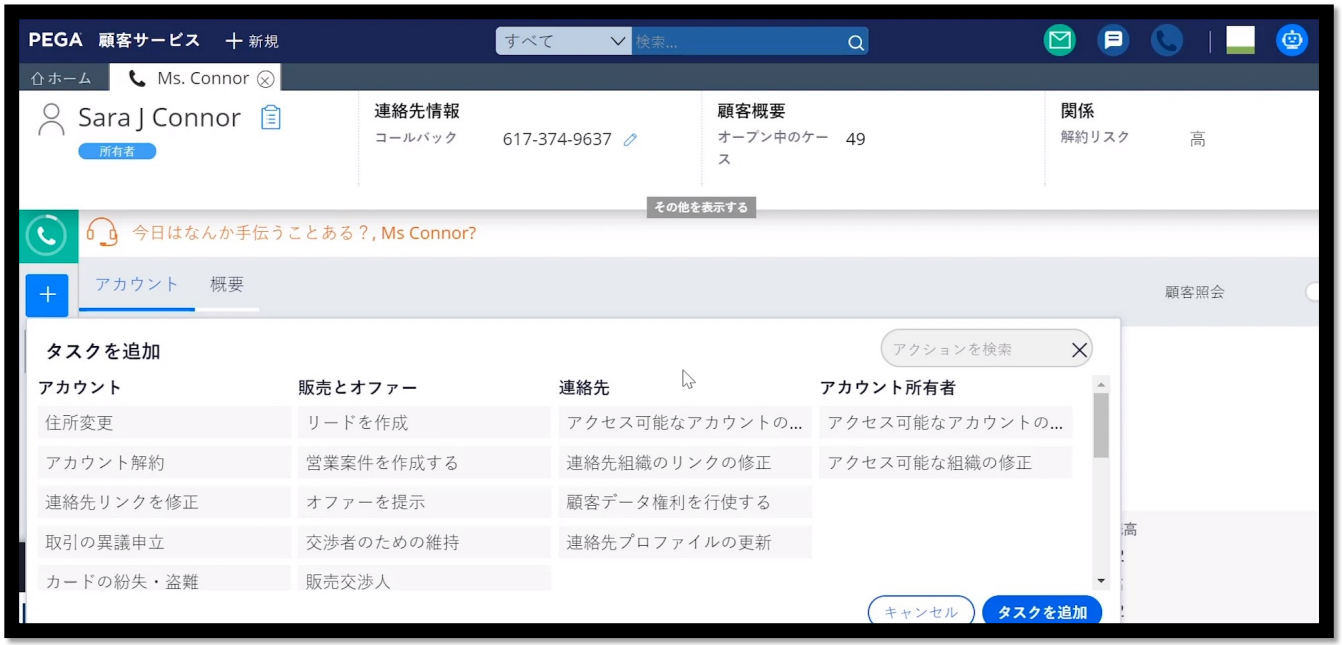
Definition

- The Accessibility Inspector tool allows you to identify and rectify some of the accessibility issues within your application.
- The Accessibility Inspector is available in both App Studio and Dev Studio, and the User portal when launched from Dev Studio.
- The Accessibility Inspector organizes the issues found into four categories.
  1. Content – Issues with the configuration of controls.
  2. Structural – Issues with the organization of content within the portal.
  3. Interactivity - Issues affecting how assistive devices navigate the UI.
  4. Compatibility – Issues with the Pega UI elements used on a form.



Localization - Use Case

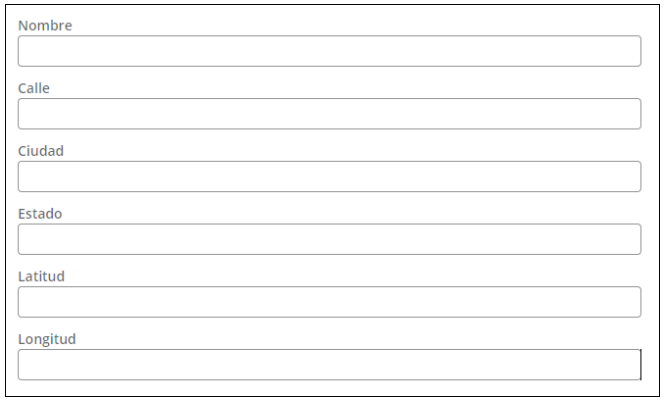
- Localizing your application helps users work in their preferred language, which improves their experience and ensures a better understanding of the product.
- By localizing an application, you can expand your operations to new markets regardless of language barriers.



Localization Design

Definition

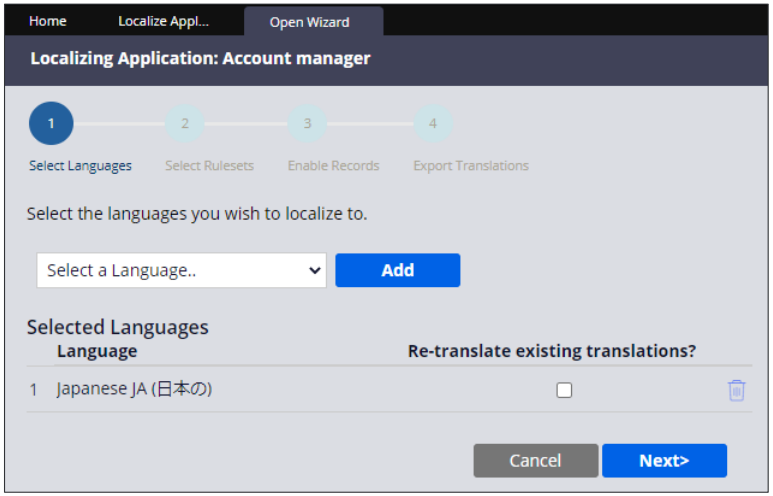
- Ensure that you design your application for localization.
  - create **field value** rules for capturing labels and notes
  - **paragraph** rules for instructions and messages
  - **correspondence** rules for emails and other correspondence
- Once these elements are added to your application, they can be localized for most languages.
- Set the Locale setting to a target language (for example, Spanish) and run your application to verify that all labels, notes, instructions, messages, and emails are successfully localized.

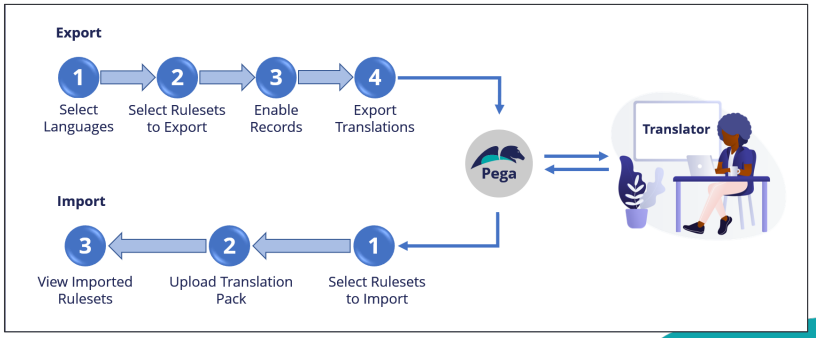


Language Pack Installation

Description

- Pega Platform provides language packs to localize the Pega rules (except the content on your UI forms).
- Language packs are collections of language-specific rulesets that support the localization of Pega Platform.
- Use the Localization wizard to export and translate the Pega rulesets and content on your UI forms.





User Experience 12% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=93g622778c024c2e

Application Development 12%

Exam Blueprint

Application Development 12%

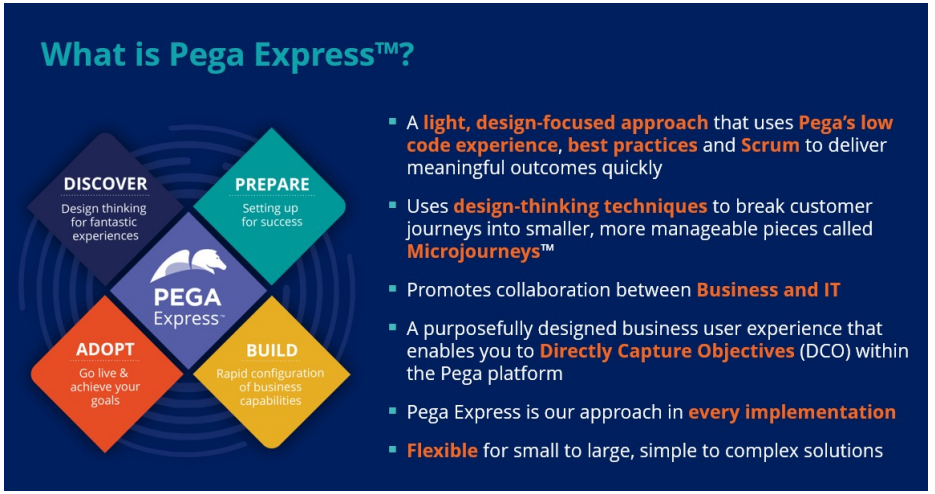
- Manage application development; user stories, feedback, bugs
- Use the Estimator to scope a Pega Platform project
- Create and maintain rules, rulesets, classes, inheritance
- Debug application errors

Pega Express as a Delivery Approach

Definition

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

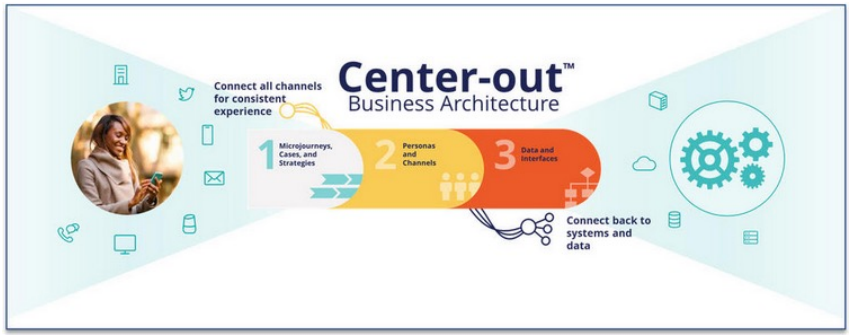
- The approach is a fusion of:
  - Pega best practices for example;
    - Directly Capture Objective (DCO) sessions
    - Concept of the microjourney
    - Strong business and IT collaboration
  - Design thinking
  - Pega's Low code capability and features enabled in the platform.
  - Scrum
- Whether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



Pega Express Delivery – 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

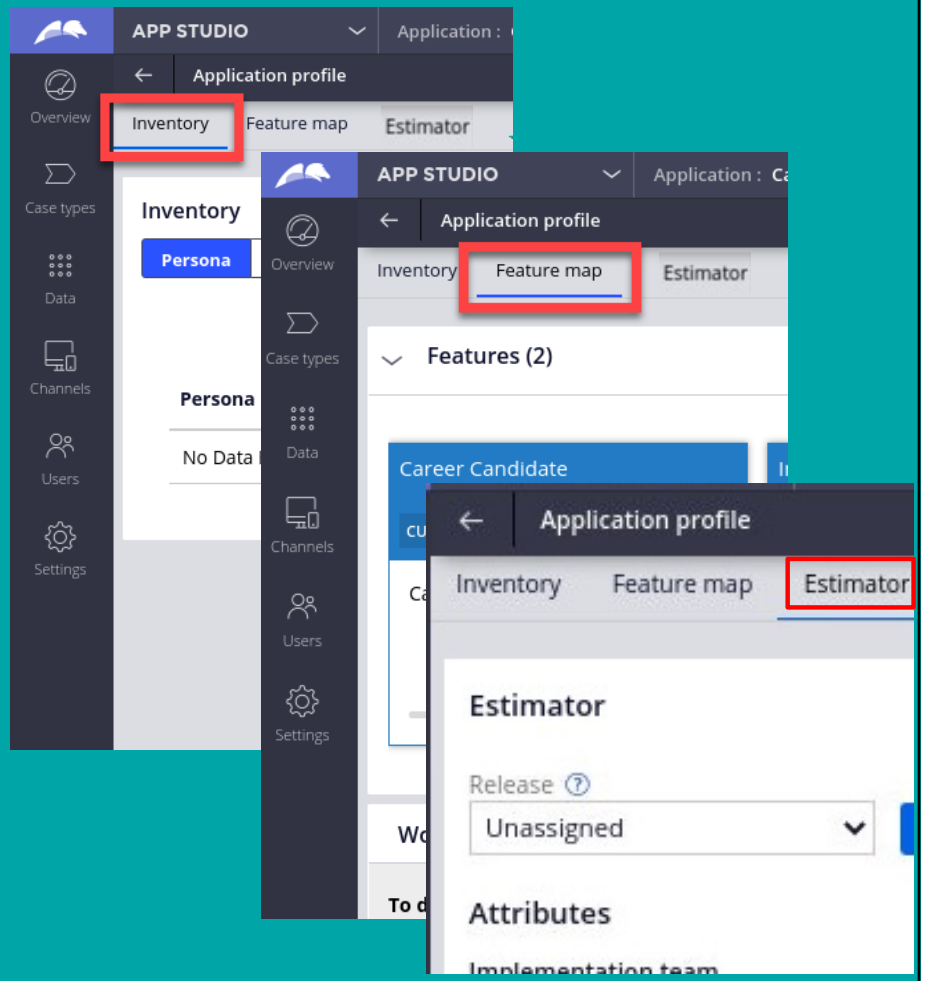
Application Profile

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

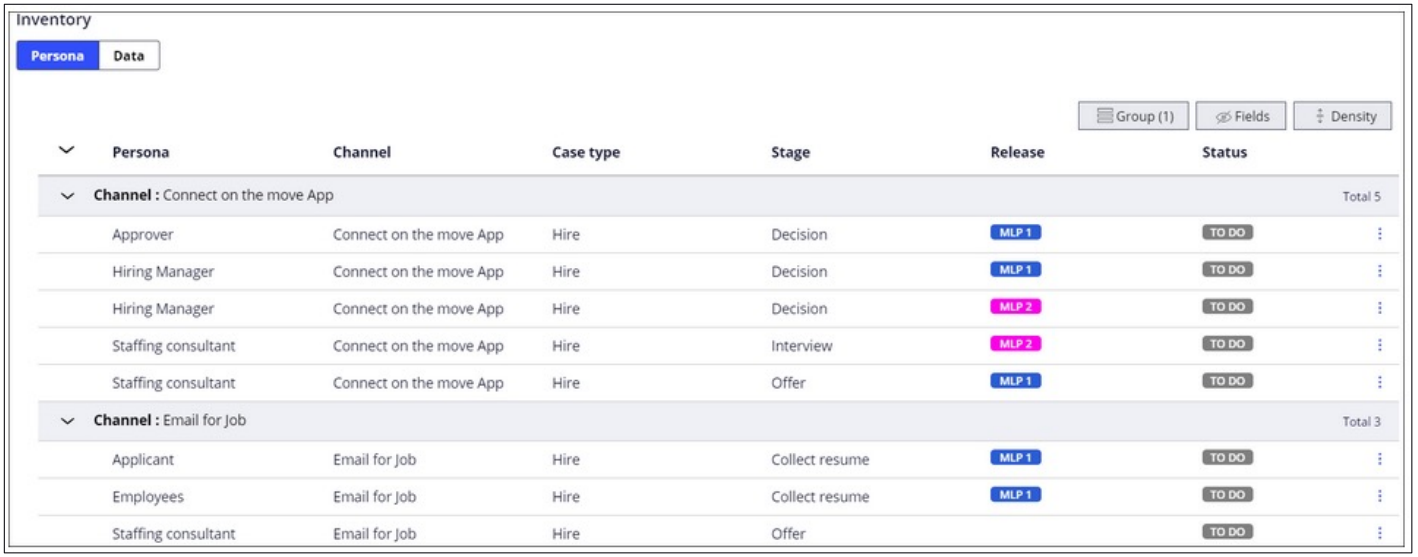
Improves communication with interested parties and provides a universal understanding of requirements.



Inventory

Description

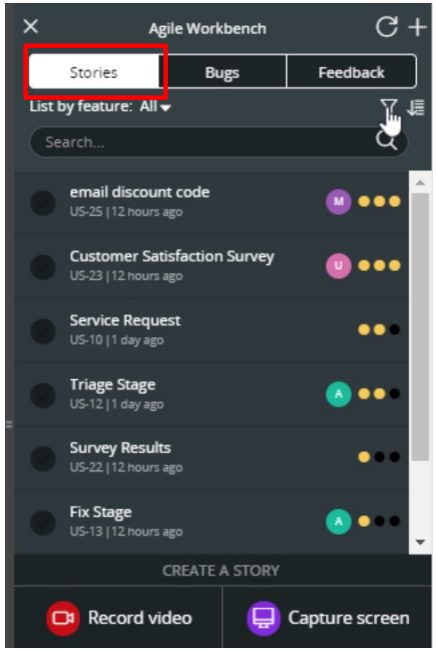
- The Inventory landing page is a tool for implementing the Pega Express methodology in delivering your projects in a goal-oriented and no-code way
- With the Inventory page, you can view lists of all personas and data objects in your application.
- You can quickly access the information that you need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

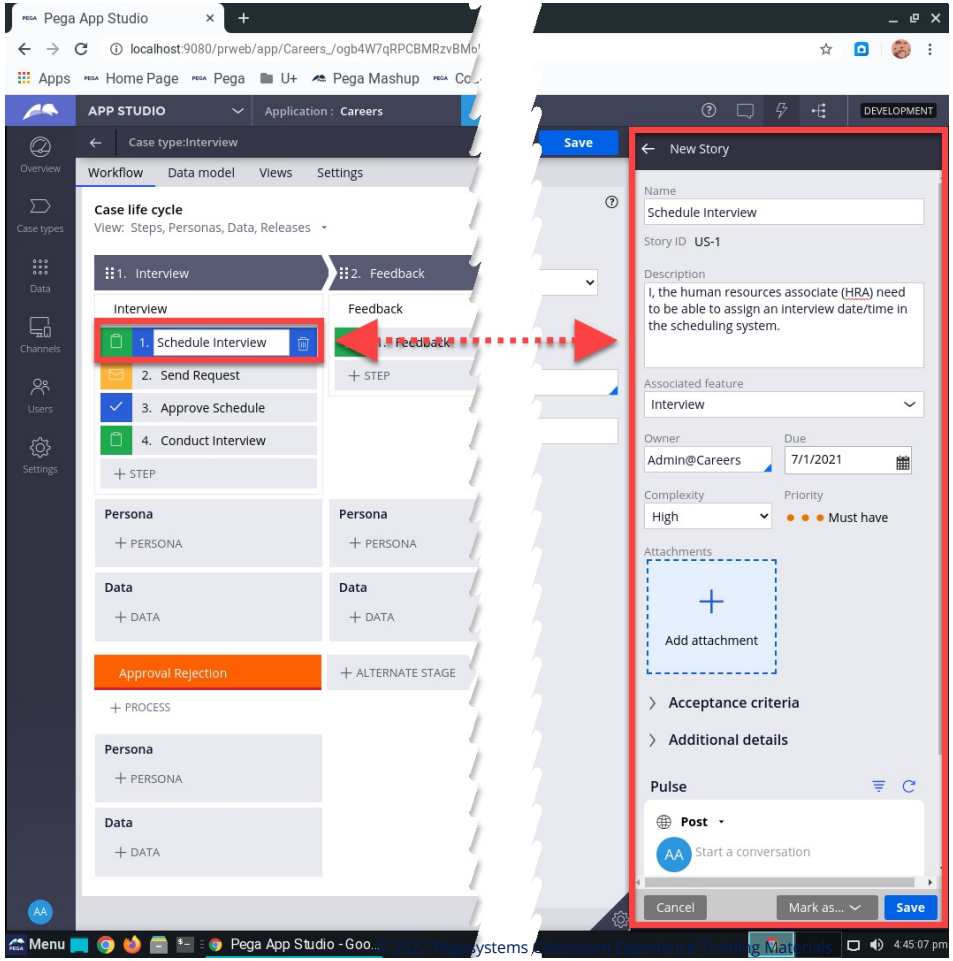


Add Stories

Description

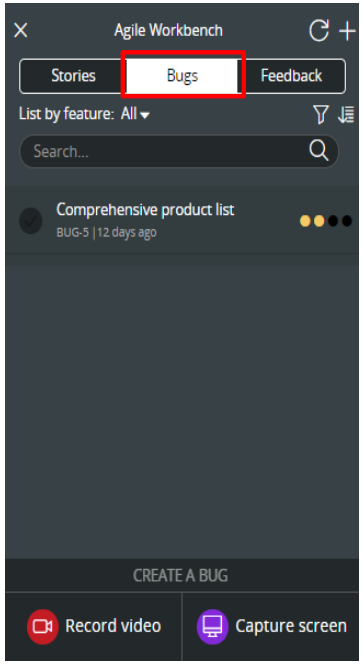
Team members may add user stories to the backlog and refine existing items for each application release.





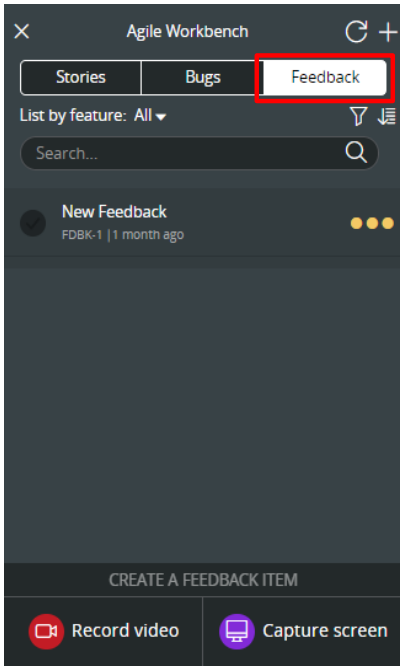
Record Bugs

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.

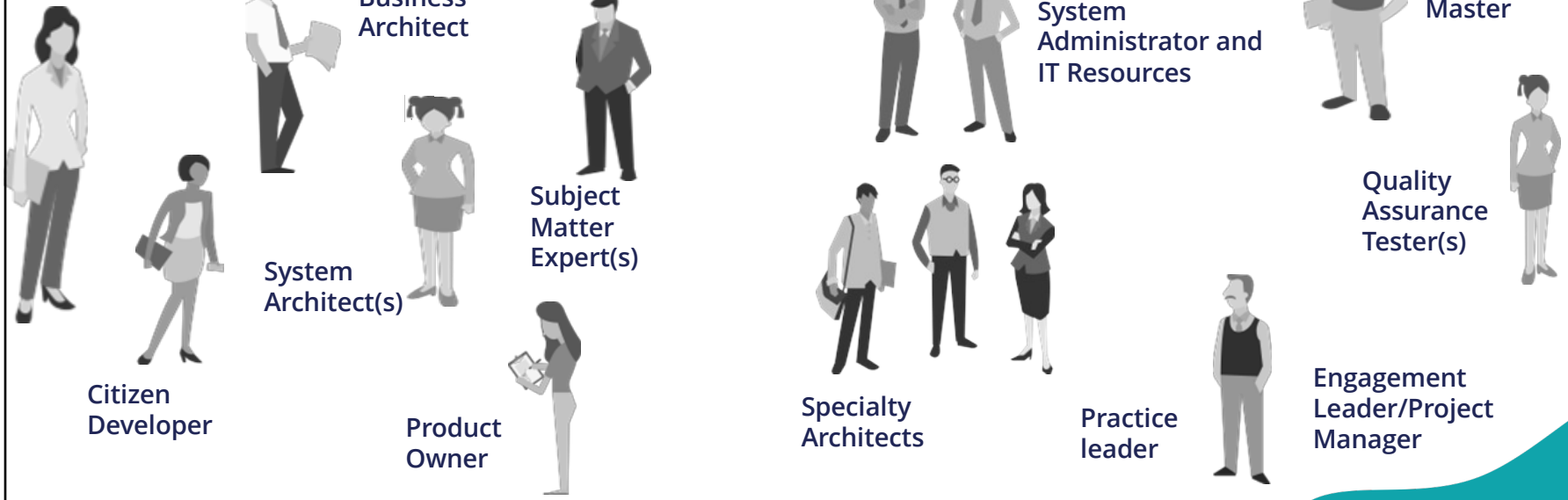


Roles

Description

Common Project Roles

The mix of team members and roles differs depending on the project goal. Some common roles found on a Pega project are shown below. Project roles should not be confused with sprint team members, although the roles may overlap



Pega Express Delivery – Design Thinking

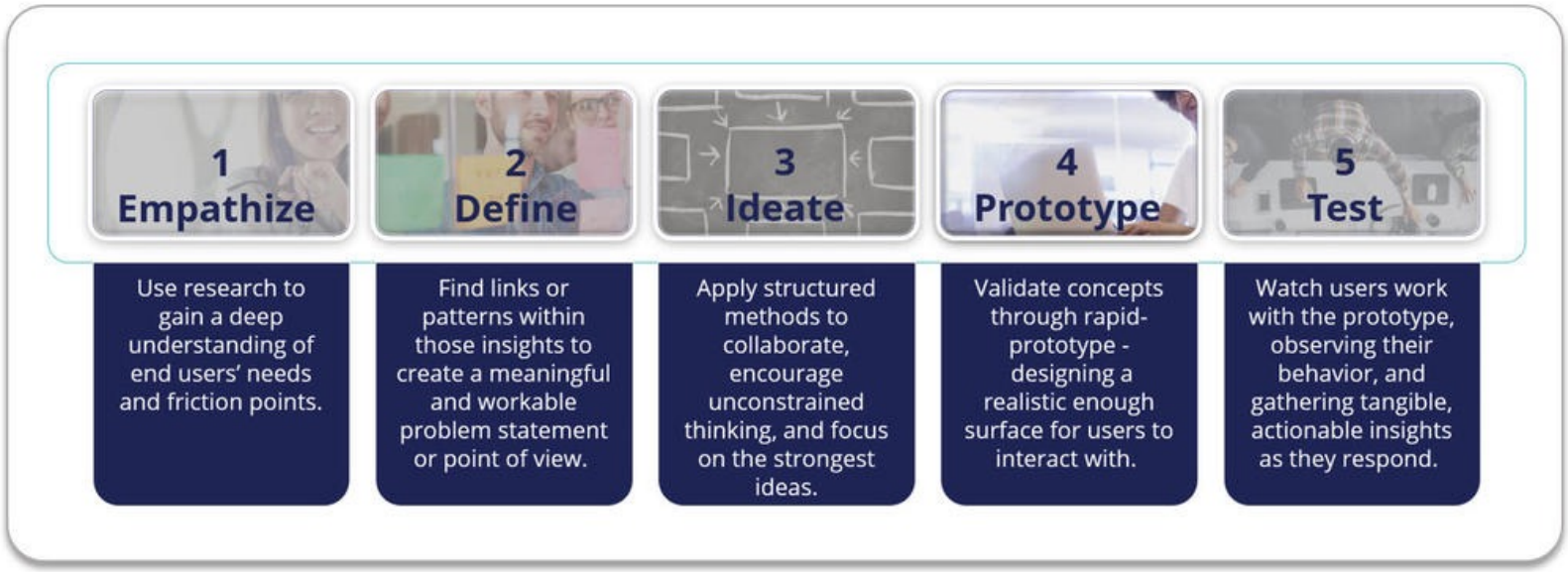
Description

- 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

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

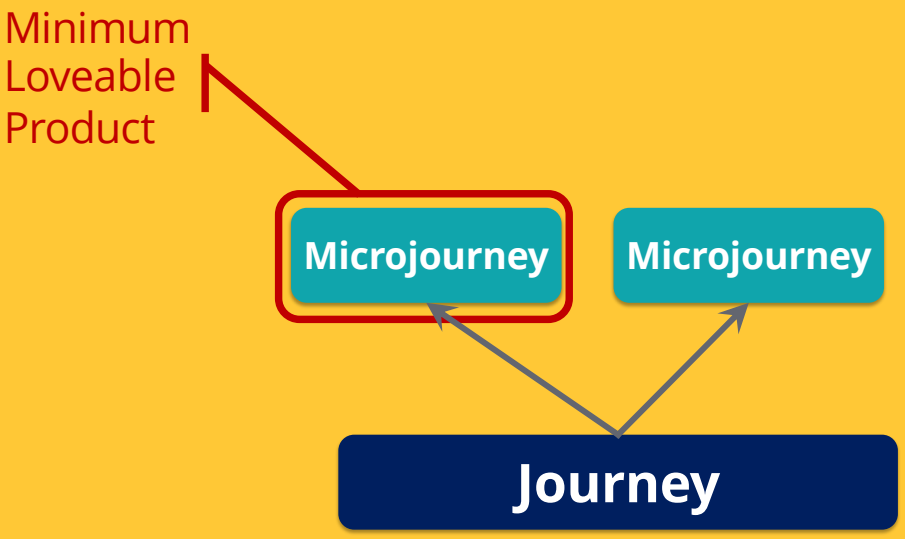
Description

- **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

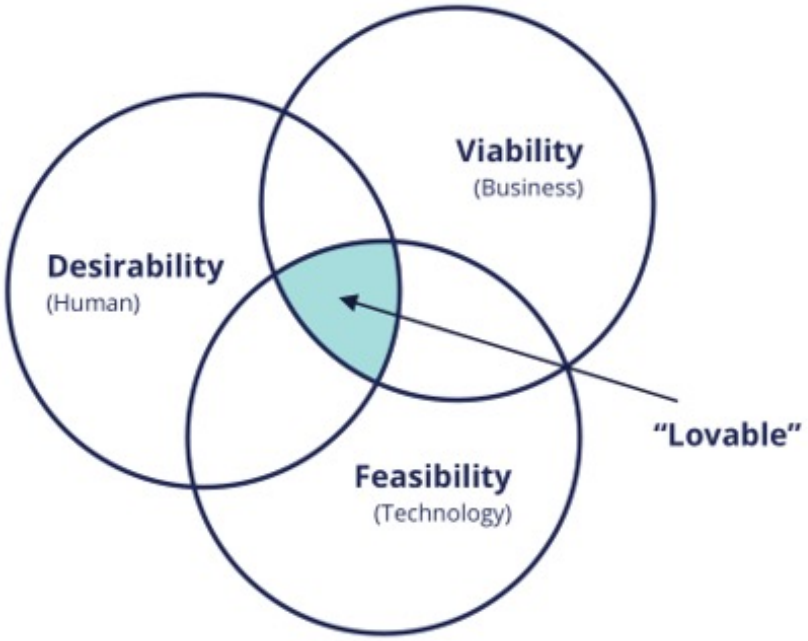
Definition

- 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**

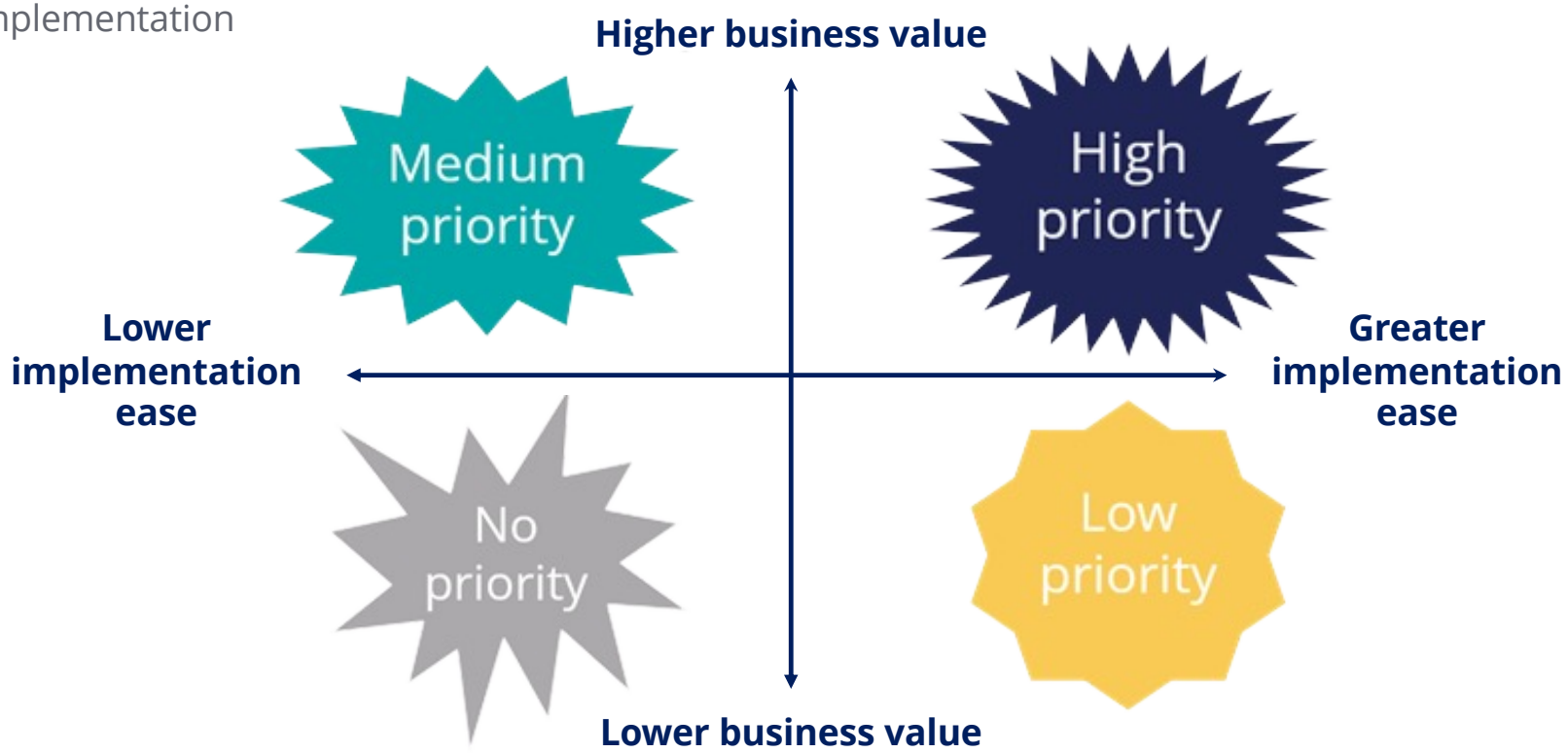
   Focus is on the business viability of projects for which the sponsors are paying to ensure return on investment

3. **Feasibility**

   Focus is on technical feasibility and effort of developing the software

MLP Priority

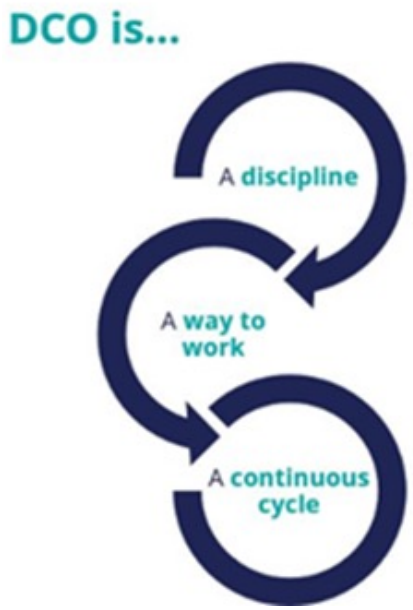
Implementation



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



Driven by...

- Collaboration
- Iteration
- Validation

Application of DCO tools

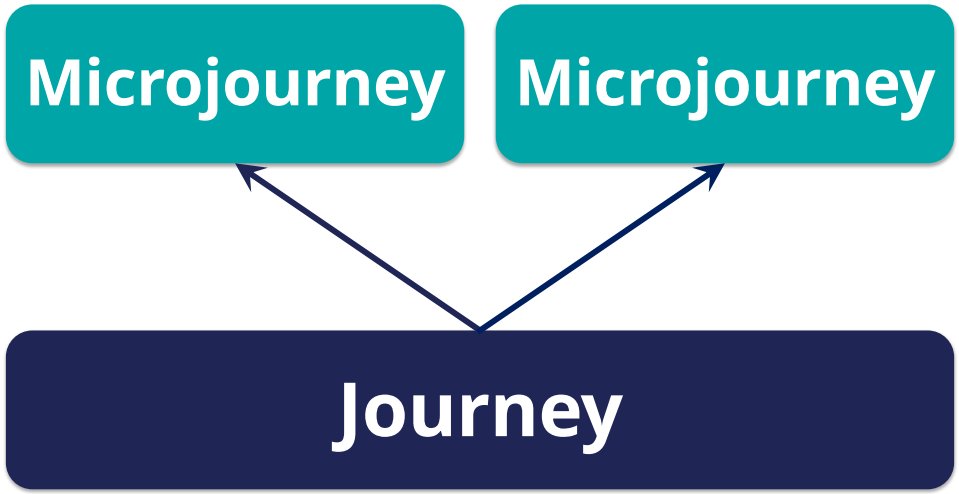
Description

DCO-related Tool	Purpose	Benefits
Case Designer	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.
Agile Workbench	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.	• Create and edit items while working in the application rather than having to access another system. • Capture a screenshot or video to include with the item.
Agile Studio	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.

Journey and Microjourney

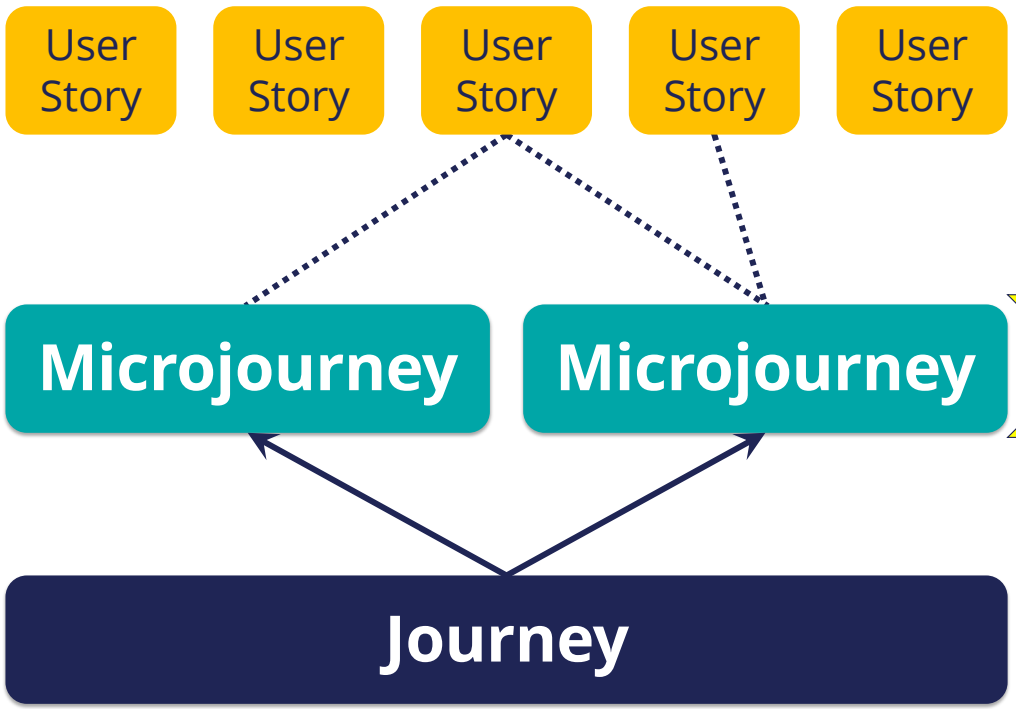
Definition

- A **Journey** is the series of interactions between a customer and an organization that occur as the customer pursues a specific goal. (Journeys are too big for an MLP.)
- **Microjourneys** are the series of interactions between one type of customer and an organization through a specific delivery channel that occur as the customer pursues a specific goal

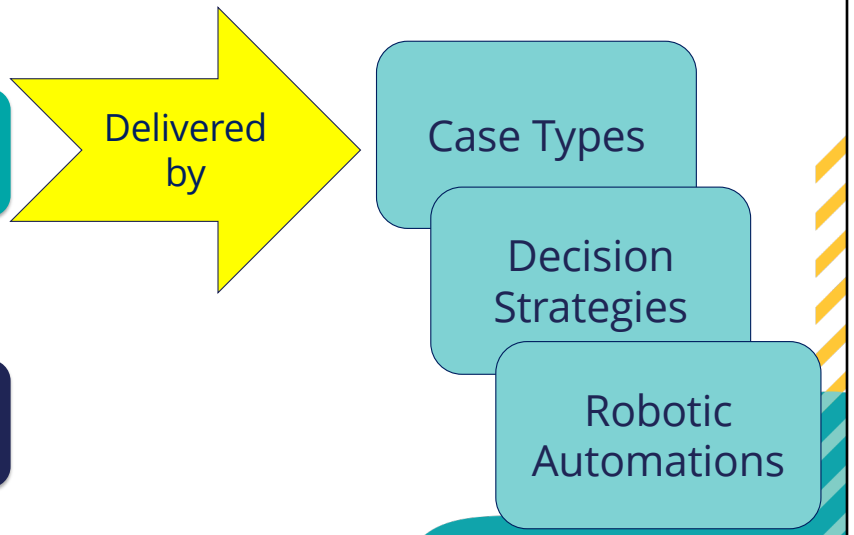


Microjourney prototype

Microjourney relationships:



Let's concentrate on completing the microjourney by prototyping a case type and modeling its processes.

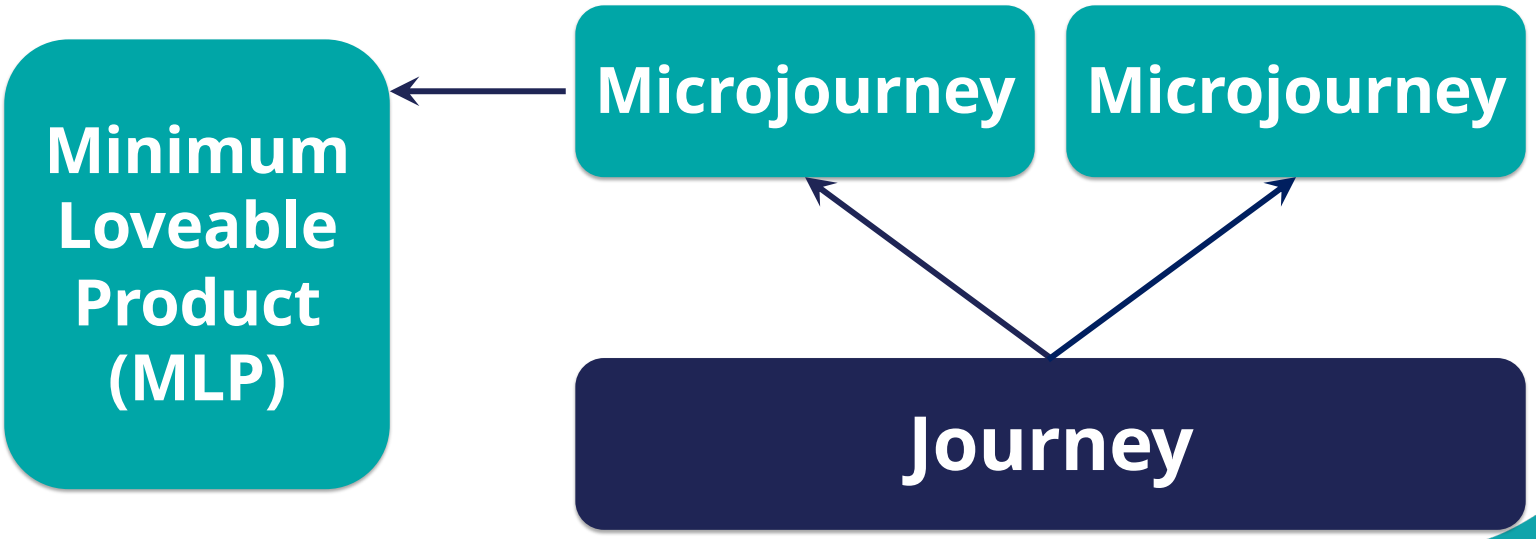


Implementation

Definition

Identify your MLP

- A program begins with a definition of its Minimum Loveable Product (MLP)
- An MLP is composed on one or more Microjourneys supported by in the MLP release



Application Profile Estimator

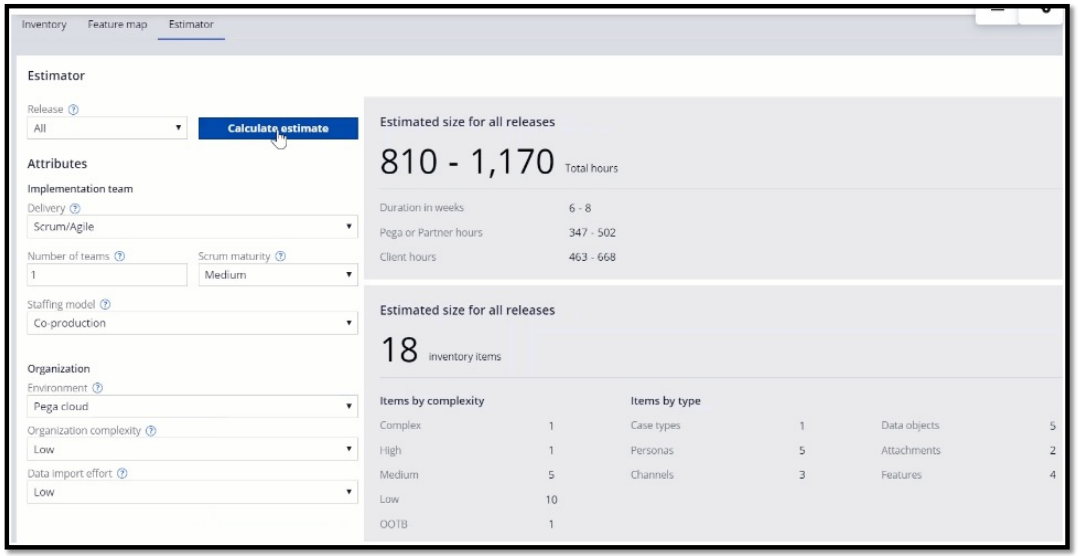
Description

- **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.

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.



Class Hierarchy

- The **Enterprise Class Structure** design pattern can be used to implement the reuse of rules through inheritance.

- For example, creating a new case type allows reuse of rules found within an application's class hierarchy, as well as standard rules from Pega, or rules contained in other applications.

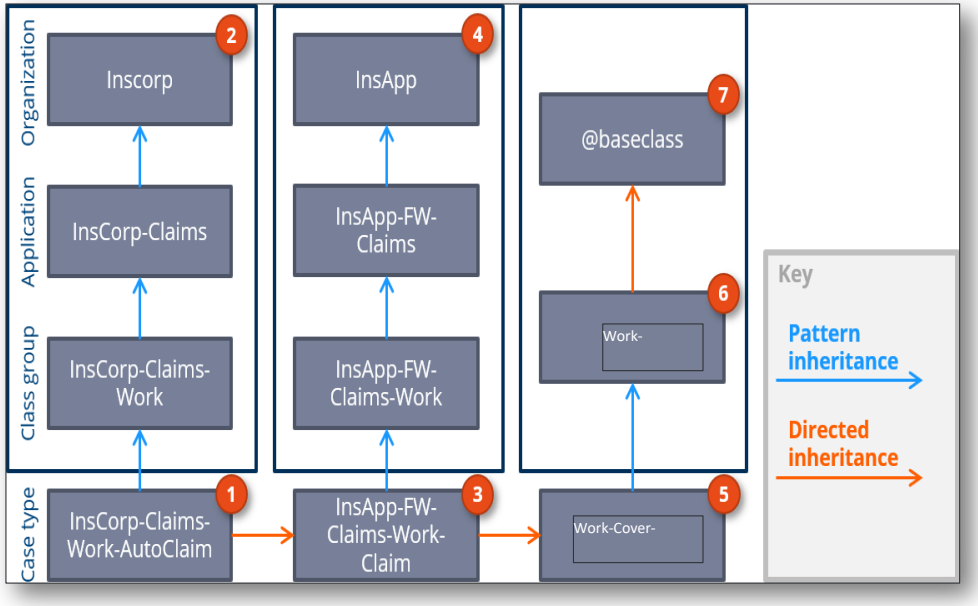
Inheritance: Job Applicant (HireMe-ApplicantMgr-Work-JobApplicant)

	Name	Label	Inheritance type
1	HireMe-ApplicantMgr-Work-JobApplicant	Job Applicant	Pattern
2	HireMe-ApplicantMgr-Work	Hire Me Reuse Layer	Pattern
3	HireMe-ApplicantMgr	ApplicantMgr Namespace	Pattern
4	HireMe	Top Level Class	Directed
5	Work-Cover-	Cover classes	Both
6	Work-	Case	Directed
7	@baseclass	@baseclass	NA

Inheritance algorithm

Implementation

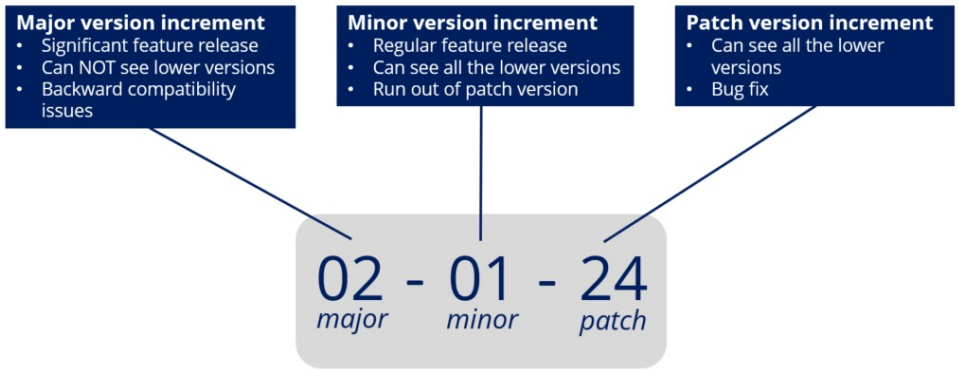
- Pega searches through the parent classes indicated by ***pattern*** inheritance first, followed by ***directed*** inheritance.
- This process repeats until Pega reaches the ultimate base class.
- If the rule cannot be found after searching ***the base class***, Pega returns an error.



Ruleset versioning

Description

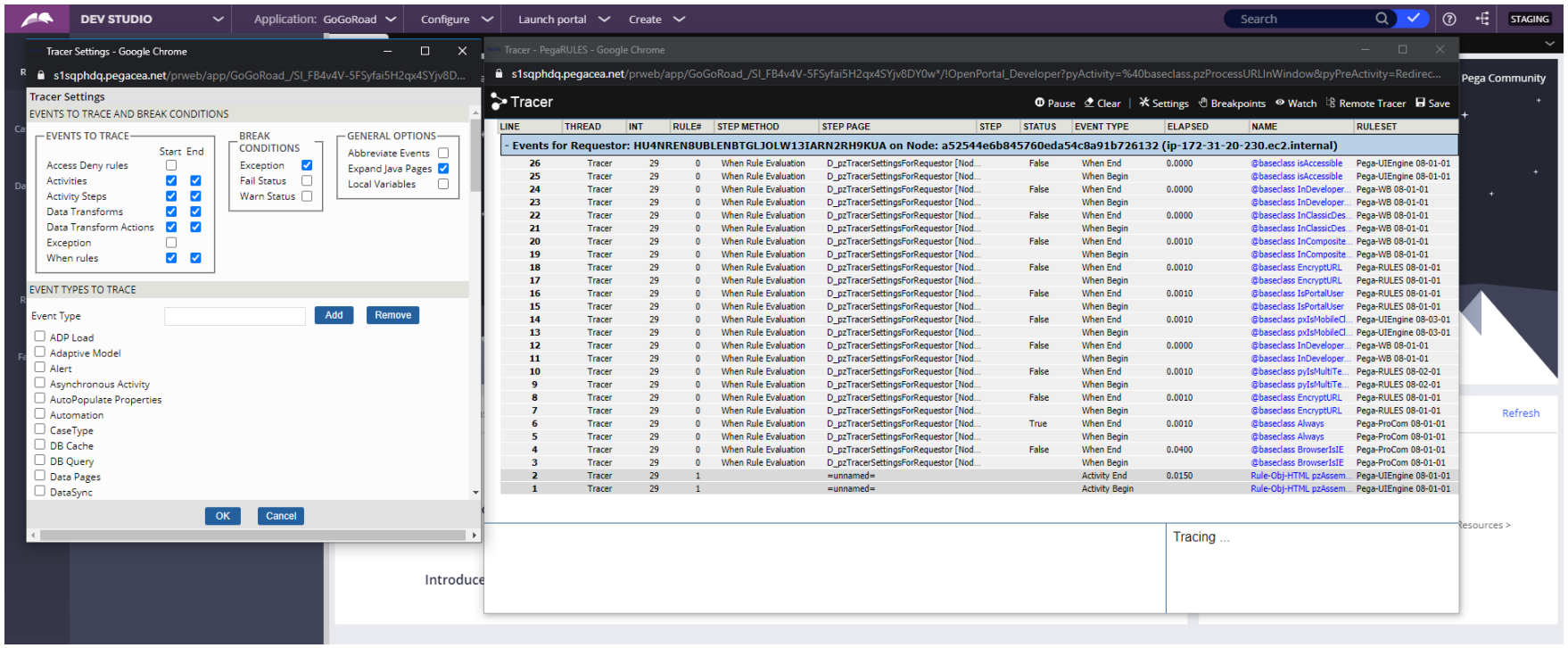
- A ruleset version has 3 parts –major, minor, patch.
- By default, for a new ruleset version, the system enters 01-01-01 as the version number.
- For existing rulesets, the system increments the highest version by one patch number.



- Major
  - –Represents a substantial release of an application
  - –Extensive changes to application functionality
- Minor
  - –Represents an interim release or enhancements to a major release
- Patch
  - –Consists of fixes to address bugs in an application

Overview

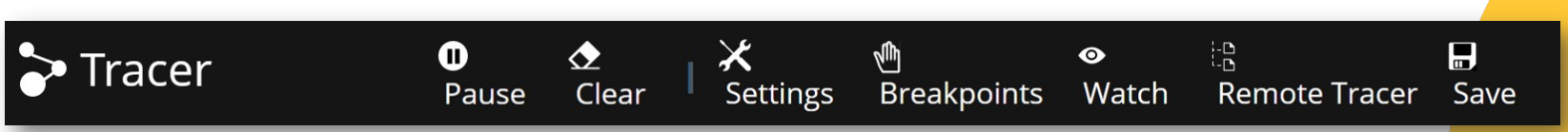
The tracer tool provides full debugging facilities, including step-by-step execution, breakpoints, and watch variables.



Tracer Toolbar

Description

- The Tracer toolbar is above the log display on the upper right.
- The toolbar consists of buttons that allow managing the events it captures.
  - ***Pause*** – suspends event logging in the Tracer
  - ***Clear*** – removes all logged events from the screen
  - ***Breakpoints –*** Identifies when to stop execution
  - ***Save*** – generates an XML file listing the events on screen
  - ***Watch*** – monitors property values or pages to detect when the value changes
  - ***Remote tracer*** – Trace events that are generated by a service requestor or another user. The user must be logged into the application to trace events.



Log files

Description

- The Pega Platform writes errors, warnings, and other debug information to log files.
- These logs track exceptions and other events that impact your application and provide insight into causes.
- Each log is managed by an appender, which determines the type of events written to the log file.

AdminStudio > Resources > Log Categories > Actions > Create Log Category. It will enable you to override the log level settings defined in the Appender, prlog4j2.xml file, and control which logging events appear in the Pega log. It will affect all the nodes in the environment.

- Depending on the settings, the following log files will display:
  - **PEGA log** contains warnings, errors, and information messages about internal operations.
  - **ALERT log** contains performance-related alerts.
  - **ALERTSECURITY** log contains alerts that suggest improper configuration of Internet Application Composer facilities, or overt attempts to bypass system security features through URL tampering.
  - **BIX log** is an optional add-on product that provides the extract functions of an ETL (extract, transform, and load) utility.
  - **SERVICES-PAL** log contains performance data saved from services.
  - **CLUSTER log** contains information about the setup and run-time behavior of the cluster.

Application Development 12% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=htk6227873f4440f

Reporting 5%

Exam Blueprint

Reporting 5%

- Create business reports
- Identify types of reports
- Use 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

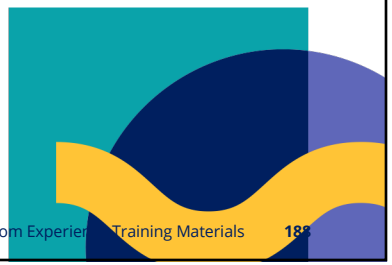
Use a report definition to identify the data to retrieve from records from a database.

- 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.

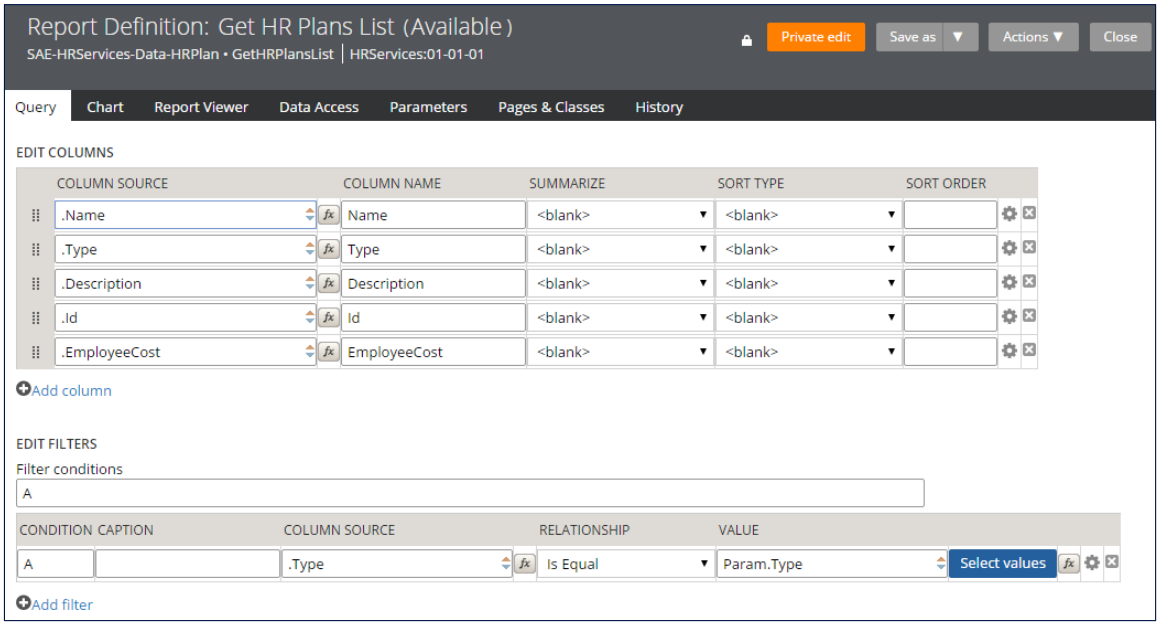
Case ID	Employee	Hire Date	Location
O-101	James Martin	2/21/16	Atlanta
O-104	Anne Walker	O-104	Anne Walker
O-746	Julia Phagan	1/12/16	Atlanta
O-983	William Kirk	9/8/16	Atlanta
O-171	Leonard Kelley	O-171	Leonard Kelley
O-623	Kate Picardo	2/25/16	Atlanta
O-421	Robert Wang	O-421	Robert Wang

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 you can make your report more performant by adding filters.

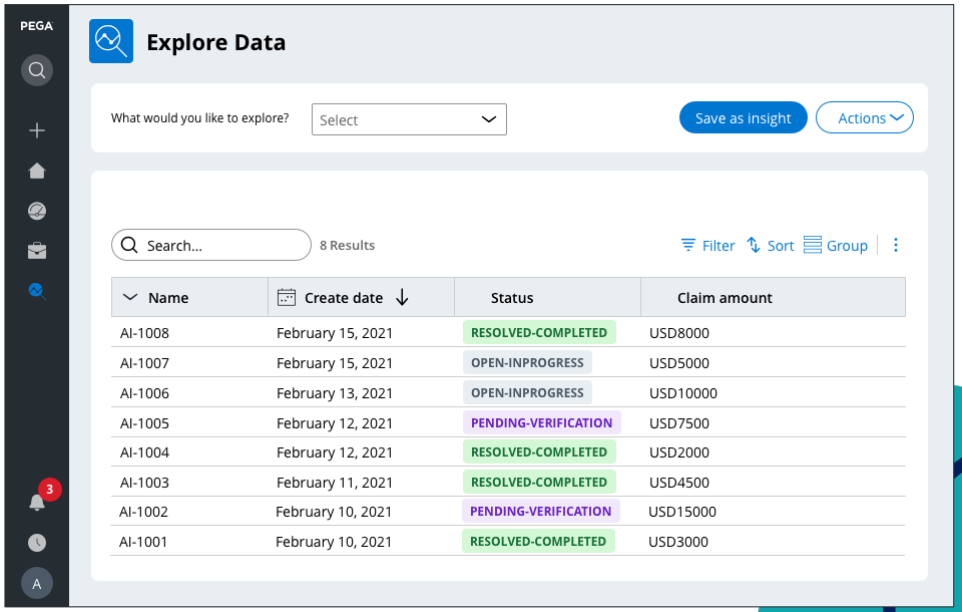


Data Explorer and Insights

- 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



Reporting 5% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=4qg62278d9a0b927

Mobility 5%

Security 5%

DevOps 5%

Exam Blueprint

Mobility 5%

- Configure mobile app channels
- Leverage Pega Mobile Preview

Security 5%

- Manage user and role assignments
- Configure security policies
- Track and audit changes to data

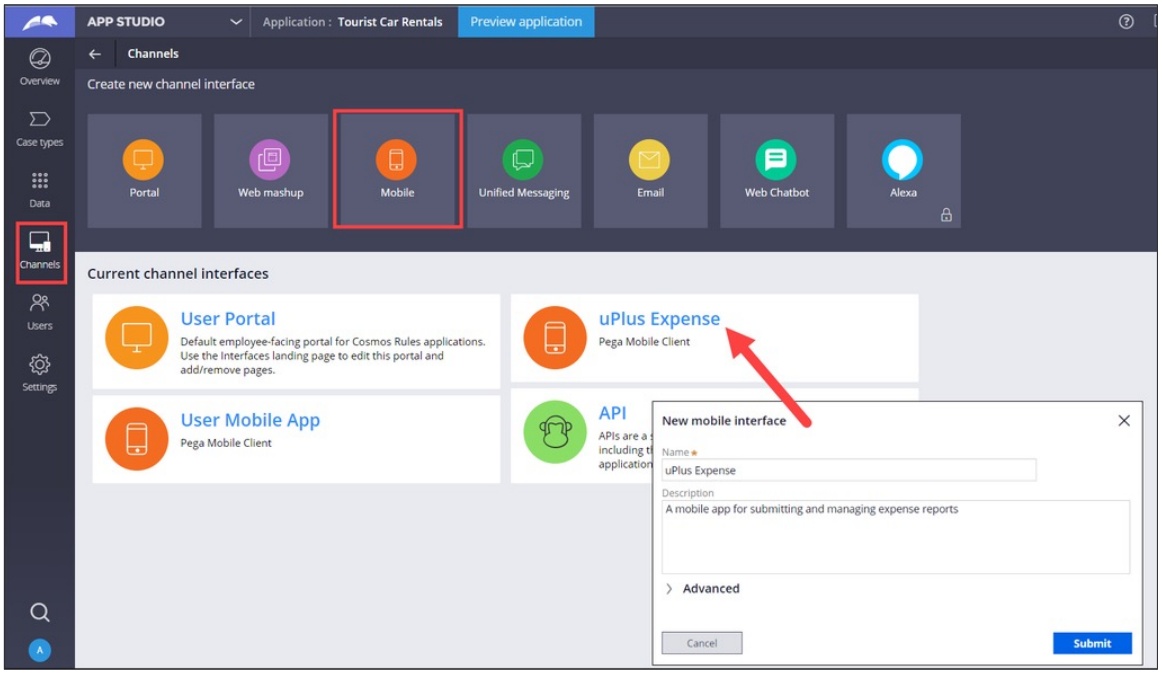
DevOps 5%

- Record a unit test
- Create and execute scenario-based test cases
- Identify best practices for configuring unit tests

Mobility

Mobile app channels

A mobile channel provides developers a way to intuitively configure and customize various aspects of mobile app behavior.



Mobile Channel

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

Security is critical

Inadequate security can prevent your application from being deployed.

Security Areas

- **Data Encryption** protects sensitive data within your application without affecting the functionality of Pega Platform
- **Authentication** ensures that only users and systems whose identities have been verified can access your application.
- **Authorization**, also called **access control**, ensures that after logging in, a user only has access to the allowed platform features, interfaces, or data.
- **Auditing**, also called **accountability**, is a systematic evaluation of the security of company information systems. Security is measured by how well it conforms to a set of established criteria.

Application-level and Feature Security

- The time to consider application security is early on, such as during the **Prepare** phase of your project, before you begin to build and configure the application.
- To protect your application from hackers and prevent unauthorized access and use, you need to manage two types of security:
  - **Application-level** - Application-level security focuses on protecting the application from outsiders and unauthorized users.
    - The goal of application-level security is to make it impossible for non-authorized users to break into, read, or otherwise access your application.
  - **Feature security** - Feature security focuses on the application by determining the case types, features, and data that authorized users can or cannot access.
    - With feature security you set up security roles for personas identified in each case type so that authorized users can access the application features they need.
    - Prevent users from viewing features or accessing data to which they should not have access

Unauthorized system access

- **Cracking** – the act of guessing a user's password
- **Brute-force attack** – a hacker makes a series of attempts to crack a user's password and achieve a successful login
- **Dictionary attack** – the hacker uses a wordlist containing known or suspected passwords to increase the chance of a successful guess. A hacker can add entries to their dictionary by:
  - –Identifying common words, such as password
  - –Spidering, or searching an organization's website to identify common terms within the organization
  - –Obtaining known passwords from a successful hack of a different site
- **Rainbow table attack -** a hacker creates a lookup table containing a list of possible passwords and hashes, then compares each hash against a list of hashed passwords. When the hashes match, the hacker identifies the password from their lookup table.
- **Hashing** – an algorithm is used to generate a number called a hash, from a text string which helps to secure passwords because a hash cannot be decrypted back to its original form

Security Policies

- **Password policies** govern the strength of user passwords.
- **CAPTCHA policies** test whether a person entered the password.
- **Lockout policies** define system behavior when a user enters an incorrect password.
- **Audit policy** determines the amount of detail written to the system log for a security issue.

Audit policy and other policies

- **Audit policy**
  - Use the Audit Policy section to customize the level of detail captured for login attempts.
- **Multi-factor authentication policies**
  - Passwords represent one way to authenticate a user. To increase security, enable **multi-factor authentication** to authenticate users. With multi-factor authentication, a user gains access only after providing multiple factors, or pieces of evidence, to verify their identity.
- **Operator disablement policy**
  - Use the **Operator disablement policy section** to automatically disable access for users who are inactive for the specified number of days.

Multi factor authentication policies

- Passwords represent one way to authenticate users.
- To increase security, enable **multi-factor authentication** to authenticate users.
- With multi-factor authentication, users gain access only after providing multiple **factors**, or pieces of evidence, to verify their identity.

Knowledge – Something that only the user knows, such as a password

Possession – Something that only the user has, such as a mobile device

Inheritance – Something that represents a characteristic of the user, such as their location

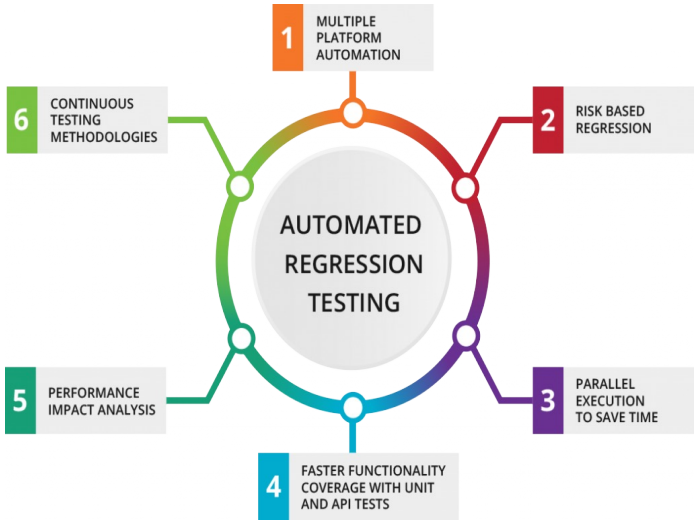
Guardrail compliance

- The most important security requirement for any Pega Platform application is to maintain guardrail compliance. Pega Platform security features are not always successfully enforced when using custom code.
- To protect your application, use the built-in security configuration features in Pega Platform. Do not rely on custom code built by developers who are not security experts.

Dev Ops

DevOps is a set of practices that bridge application development and operational behavior to reduce time to market without compromising on quality and operational effectiveness.

- During development changes are made to existing case types and their processes. If those processes change the user experience, new scenario tests should be created.
- If the user experience is inadvertently changed, automated regression testing using the existing scenario tests will fail and bring this issue to the attention of the development team.



Unit Testing

- **UNIT TESTING** is a level of software **testing** where individual components of a software are tested
- The purpose is to validate that each **rule** of the software performs as designed
- A **rule instance** is the smallest testable part of any Pega application. It usually has one or a few inputs and usually a single output
- The results of a unit test can be converted into a **Test Case**
- To protect your application, use the built-in security configuration features in Pega Platform. Do not rely on custom code built by developers who are not security experts.

Converting unit tests into test cases

Description

- For most rules, unit tests can be converted into test cases that can be reused to test the rule.
- By configuring case details, and then defining the expected results with **assertions** (test conditions) the unit test can be reused as a test case
- When the test case runs, the test results are comparted to the expected results defined for the rule's assertions. If the test results do not meet the defined assertions, the test case fails.

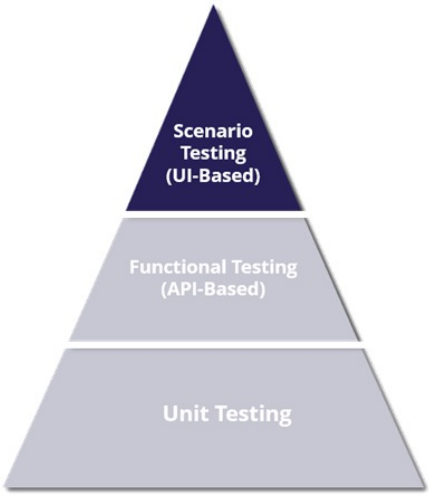
Examples of assertions and their uses

Assertion type	Usage	Example
Property	Tests the value of the specified property. Requires the page on which the property is defined, a comparison operation, and a comparison value.	pxUrgencyWork is equal to 10
Decision result	Tests the value returned by a decision rule. Requires values for each input property needed by the decision rule to return the expected result.	When Referred by employee is false, return RecruitingWB
Expected run time	Tests whether a rule executes within an allowed amount of time. Requires a comparison operation and an allowed time in seconds.	Expected run time is less than or equal to three seconds

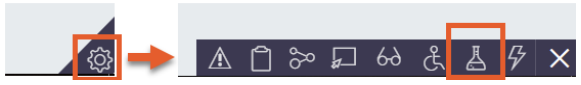
Scenario Testing

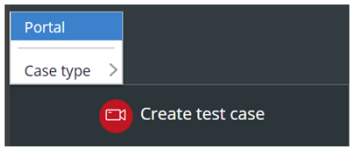
Description

- **Ideal Test Pyramid:**
  - Bottom of the pyramid the least expensive to run
  - Top of the pyramid is the most expensive
- UI-Based functional test and scenario tests verify that cases are working as expected.

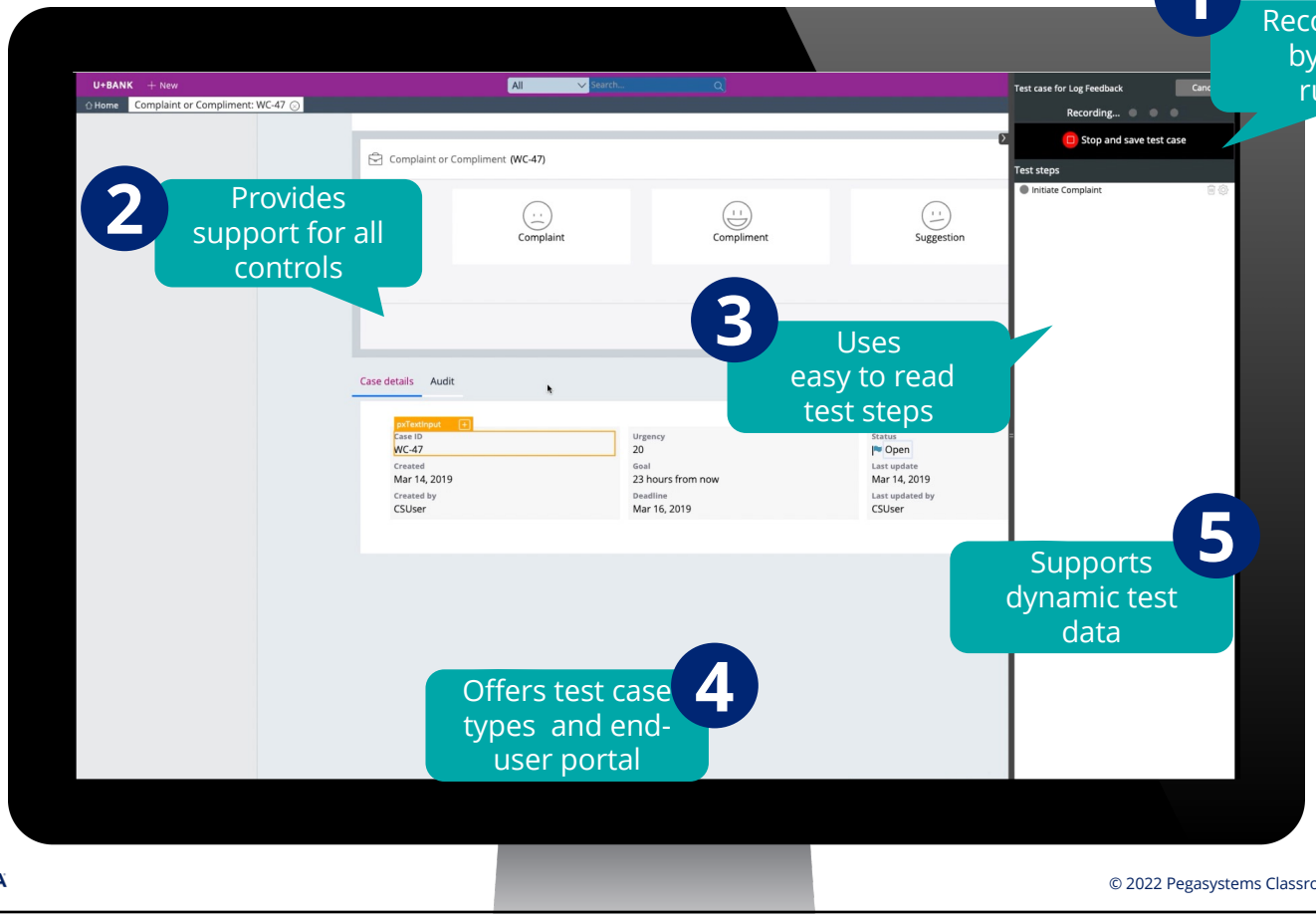


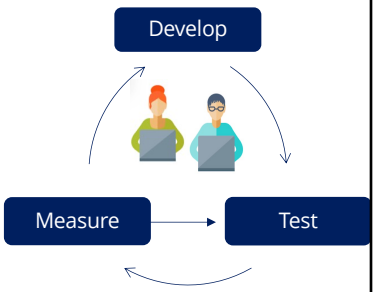
- **Scenario tests** are captured in the context of the application portal only, such as the **User** or a similar application portal.
- You use the **Automation Recorder** on the runtime toolbar to create or modify a scenario test.





Scenario testing





Scenario Testing

Tests are saved in a dedicated test ruleset that is defined in the application rule.

Scenario tests for features

- With scenario testing, you can create UI-based, end-to-end scenarios to test your application. The user who has access to the run-time toolbar captures scenario tests in the context of the application portal.
- In the Dev Studio application portal, use the Launch portal menu to select and navigate to the specific portal (such as the User Portal) to initiate test recording. Use the Automation Recorder on the run-time toolbar to create or modify a scenario test.

Best Practices for Unit and Scenario testing

- Saving the test case requires access to a ruleset that is configured to store test cases.
- If the ruleset selected is not configured to store test cases, Pega Platform returns an error.
- Before you record a unit test, System Administrator should provide a suitable ruleset.
- Save test cases and all related rules to a dedicated testing ruleset for maintenance and packaging

Mobility, Security, DevOps 15% - Quiz

ClassMarker Quiz

ClassMarker Quiz link:

https://www.classmarker.com/online-test/start/?quiz=39762278f0dc4d20


PEGA®