3-2016

Automation of Global Hiring Process

Sandeep Reddy Basireddy
St. Cloud State University

Automation of Global Hiring Process

by

Sandeep Reddy Basireddy

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

In Partial Fulfillment of the Requirements

For the Degree

Master of Engineering Management

May, 2016

Starred Paper Committee:
Hiral Shah, Chairperson
Ben Baliga
Balasubramanian Kasi

Abstract

This project was implemented in a company is a business aligned technology services provider with proven capabilities in software application development, application management and systems and technology integration services and an expert in building business value for global financial services and Fortune 500 companies. This project proposal will focus on implementation of algorithms to automate hiring process. As this project involves collecting information about the candidate, matching the candidate to an open position, scheduling an interview with a hiring manager, collecting and processing the feedback upon which a hiring decision is made, and then extending an offer to the candidate.

Acknowledgments

This project would not be possible without the help and encouragement from the professors of my committee Dr. Hiral Shah, Dr. Ben Baliga and Prof. Balsy Kasi. The professors really helped me in gaining knowledge in every subject that was taken during my Master’s in Engineering Management at St. Cloud State University. I would also like to thank Dr. Ben Baliga. The resources that were provided during the study at St. Cloud State University were tremendous. Finally, I would like to thank my friends and family for providing a great support in the fulfillment of my dream study at St. Cloud State University.

Table of Contents

Page

List of Tables

List of Figures

Figure Page

Chapter 1: Introduction

Introduction

This company, the largest independent pure-play technology consulting and outsourcing provider for the financial services industry is a $300-million firm based in New York. Since inception in 2001, This company has been on a steep growth trajectory. With 5,500+ professionals operating in 16 countries across the world, it has presence across US, Australia, Canada, UK, Japan, The Netherlands, Hong Kong, Singapore, UAE, Ireland, Germany, Switzerland, Luxembourg, Italy, France, and Development Centers in India.

This company is a business aligned technology services provider with proven capabilities in software application development, application management and systems and technology integration services and an expert in building business value for global financial services and Fortune 500 companies. Its services are strategically aligned with clients' business needs for growth, profitability optimization, efficiency, and compliance transparency. This company meets the challenges of its clients in industries such as Capital Markets, Insurance, Banking and Digital.

This capstone project proposal will focus on implementation of algorithms to automate hiring process. As this project involves collecting information about the candidate, matching the candidate to an open position, scheduling an interview with a hiring manager, collecting and processing the feedback upon which a hiring decision is made, and then extending an offer to the candidate.

Problem Statement

This company uses Applicant tracking system to manage hiring process. Over time, some business processes were modified in order to fit into the Advantage mold. A new system has to be implemented from the ground up that will allow us to easily configure/ make changes on the go and efficiently address hiring services needs in a nimble fashion. In hiring, there are a lot of manual processes that include extending an offer to the candidate. Also, the existing legacy applications are unable to meet the modern day demand leading to several issues.

Nature and Significance of the Problem

Present system has traditionally been thought of as a way to manage the HR functions and employees of a business. As the technology developed and HR functions changed, there was a need to develop software that handled the process of recruitment and hiring. Having the right recruitment software can change the way you recruit, save you time and money during the recruiting process, and most importantly, ensure that you are hiring the best people.

Also, many of the administration tasks that have been strangling recruiters and HR professionals. The amount of time spent on paper based systems and administrative tasks have caused many HR managers to look carefully at the benefits software.

In another case, the modern day web applications should support cross browser compatible to mobiles and tablets. Our current legacy application has some constraints when it comes to this feature. So, there is a need to address this to make it user friendly.

Objective of the Project

The objectives of the project were:

- Decrease turnaround time of the hiring process by 40%.
- Cut cost of the overall hiring cycle.
- Decrease the maintenance required.
- Making the application user friendly by the introduction of cross browser compatibility feature.

Project Questions

1. What are the time savings after implementing the automation?
2. What is the cost savings after implementing the Automation?
3. What is the maximum limit to which the performance can be improved?
4. What needs to be done to implement the cross browser compatible feature?

Limitations of the Project

This project was implemented with limited real time data and the results that are obtained for the maximal product quantity will be available in real time implementation where in which it also considers the platform on which the project is executing and the resources that are available at that point of time. However, the calculations are accurate at every point in measuring irrespective of the quantity that is available.

Summary

This chapter briefly covered many aspects of this project prominently to determine the actual problem that exists and how it affects in real time, main motive of the project, list of questions that are going to be answered at the end of the study, basic limitations of the project and, finally, the definition of all the terms that are used in this project to fully understand the meaning of each term. The next chapter covers the literature background knowledge associated with this project.

Chapter 2: Background and Review of Literature

Introduction

This chapter focuses towards reviewing the literature of the problem, literature related to the methodology that has been implemented in the process of solving the problem and the background of Company and the issues related to it.

Background Related to Problem

This company is a business aligned technology services provider with proven capabilities in software application development with 5,500+ professionals operating in 16 countries across the world.

As the technology developed and HR functions changed, there was a need to develop software that handled the process of recruitment and hiring. The amount of time spent on paper based systems and administrative tasks have caused many HR managers to look carefully at the benefits software and current legacy application has some constraints when it comes to this features.

Literature Related to the Problem

To reduce the time and increase the automation of the process Pega software is used which is developed by Pegasystems. Pegasystem develops, markets, licenses, and supports software, which allows organizations to build, deploy, and change enterprise applications. The Company also provides implementation, consulting, training, and technical support services to its customers. Its PegaRULES Process Commander (PRPC) software provides a unified platform that includes all the necessary elements and technologies to build enterprise applications. These elements, which include the direct capture of business objectives, automating programming, reporting, predictive analytics and decisioning, dynamic case management, mobile device access, generated user interface, process automation, business rules, social media technologies and automating work, are all part of its integrated architecture. The Company has customers in sectors, such as financial services, healthcare, insurance, communications and media, Government and other industries (Pegasystems, n.d.).

Description of the process that has to be automated:

First, a candidate applies for an open position. To do this, some basic contact information has to be provided, along with the educational background and work experience where back group, and perhaps highlight some relevant skills that support the requirement.

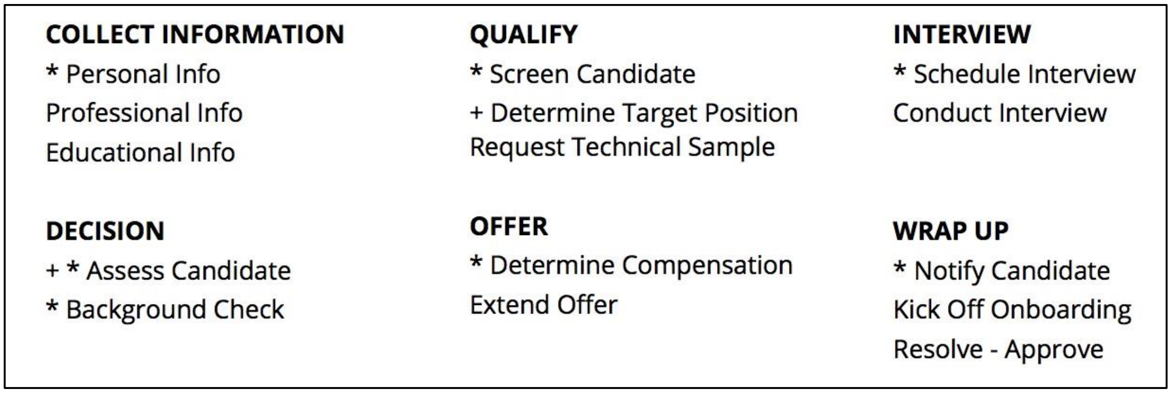
Once the candidate submits their information, a recruiter needs to review it, before scheduling an interview. The recruiter should screen the candidate to better understand their experience and qualifications. If the candidate is not a good fit, the system runs to check if the candidate qualifies for a different position, or elect not to pursue the candidate.

Now that the recruiter has qualified the candidate, an interview can be setup with the hiring manager. Once the hiring manager completes the interview, feedback and recommendation are provided, which will be the basis to make a hiring decision. Once the interview is complete, the candidate is assessed, a hiring decision is made, and the offer of employment is extended to the best candidate. Since the approvals process can be time-consuming–and may involve repetition if the candidate negotiate compensation, joining etc.–this is split into two stages: Decision and Offer.

Once the candidate accepts the offer, the process is just about done baring a few finishing tasks–update the open position so that someone else doesn't apply for it, close the candidate case, and initiate the onboarding process.

Also, another stage needs to be created to reflect the other task hinted above: rejecting a candidate. This stage isn't part of the primary path of our case.

Figure 1 gives overview of Process outline which is divided in to six stages- Collect Information, Qualify Interview, Decision, Offer and Wrap-up.



COLLECT INFORMATION * Personal Info Professional Info Educational Info	**QUALIFY** * Screen Candidate + Determine Target Position Request Technical Sample	**INTERVIEW** * Schedule Interview Conduct Interview
DECISION + * Assess Candidate * Background Check	**OFFER** * Determine Compensation Extend Offer	**WRAP UP** * Notify Candidate Kick Off Onboarding Resolve - Approve

Figure 1: Process Outline of Project

Later out of the box custom reports are generated by pega to analyze performance, quality, monitor assignments, processes, etc.

Literature Related to the Methodology

Agile methodology is an alternative to traditional project management, typically used in this software development.

Agile software development is a group of software development methods based on iterative and incremental development in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen tight iterations throughout the development cycle (Highsmith, 2001).

Agile methodology is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development (Agile methodology, 2008).

Scrum is the most popular way of introducing Agility due to its simplicity and flexibility. Because of this popularity, many organizations claim to be “doing Scrum” but are not doing anything close to Scrum’s actual definition. Scrum emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short iterations. Doing Scrum as it’s actually defined usually comes into conflict with existing habits at established non-Agile organizations (Scrum, n.d.).

Adaptive Software Development (**ASD**) is a software development process that grew out application development work by Jim Highsmith and Sam Bayer. It embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs (Pressman, 2008).

Agile development provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development—requirements, design, etc.—is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there’s always time to steer it in another direction (Agile software development, n.d.).

Adaptive Software Development replaces the traditional waterfall cycle with a repeating series of speculate, collaborate, and learn cycles. This dynamic cycle provides for continuous learning and adaptation to the emergent state of the project (Manifesto for Agile Software Development, n.d.)

The characteristics of an ASD life cycle are that it is mission focused, feature based, iterative, time boxed, risk driven, and change tolerant (Highsmith, 2000).

Amazon Data base services are used capture the data produced during the development process. **Amazon Relational Database Service** or **Amazon RDS** is a distributed relational database service by Amazon.com. It is a web service running “in the cloud” and provides a relational database for use in applications. It is aimed at simplifying the set-up, operation, and scaling a relational database. Complex administration processes like patching the database software; backing up databases and enabling point-in-time recovery are managed automatically (Amazon Web Service, 2010).

Various tasks that will be executed through the Agile Methodology are:

1. Planning
    - Identification of stakeholders and sponsors
    - Infrastructure Requirements
    - Security related information and gathering
    - Service Level Agreements and its conditions
2. Analysis
    - Capturing of Stories in Parking lot
    - Prioritize stories in Parking lot
    - Scrubbing of stories for estimation
    - Define Iteration SPAN (Time)
    - Resource planning for both Development and QA teams
3. Design
    - Break down of tasks

- Test Scenario preparation for each task
- Regression Automation

4. Execution
    - Coding
    - Unit Testing
    - Execution of Manual test scenarios
    - Defect Report generation
    - Conversion of Manual to Automation regression test cases
    - Mid Iteration review
    - End of Iteration review
5. Closure
    - Pilot Launch
    - Training
    - Production Launch
    - SLA Guarantee assurance
    - Review SOA strategy

Summary

This chapter briefly covered the background and literature of the problem. It also described the methodology used and explained different phases of the Agile methodology.

Chapter 3: Methodology

Introduction

This chapter covers the design of study, data collection and data analyses of the project.

Design of Study

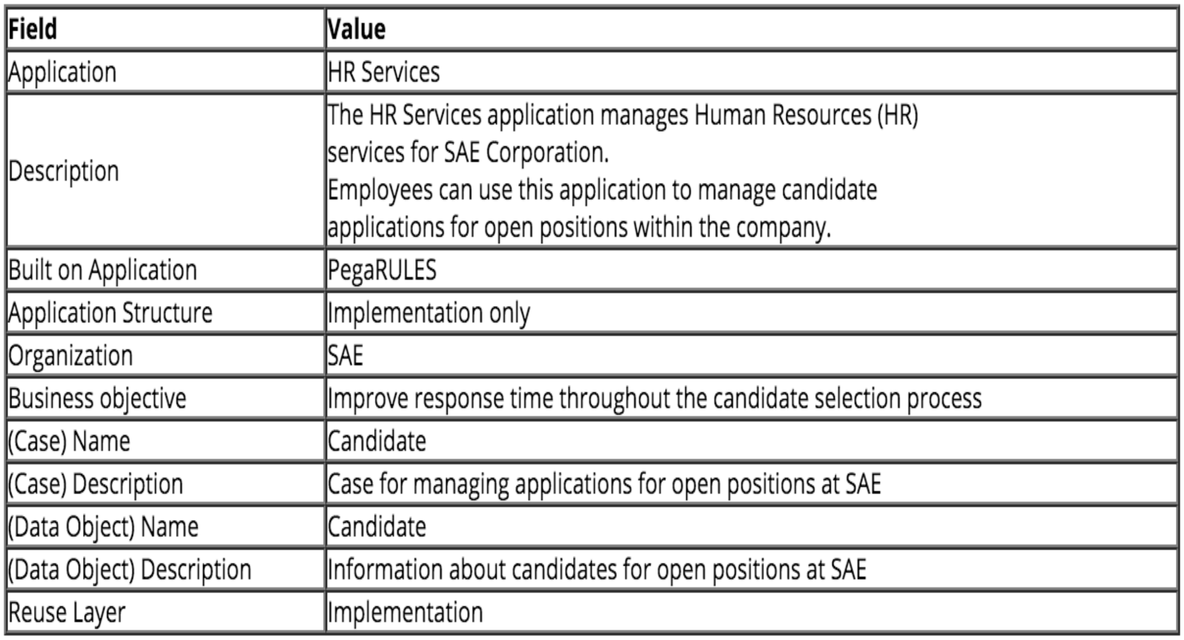
It covers the architecture used for the design of the project.

Creating New Application

Create the HR Services Application. New Application wizard was used and provided the following information to create an application:

- Basic application information, such as the name and description of the application.
- Business objectives that the application must satisfy.

Table 1: Hr. Services Application Basic Details



Field	**Value**
Application	HR Services
Description	The HR Services application manages Human Resources (HR) services for SAE Corporation. Employees can use this application to manage candidate applications for open positions within the company.
Built on Application	PegaRULES
Application Structure	Implementation only
Organization	SAE
Business objective	Improve response time throughout the candidate selection process
(Case) Name	Candidate
(Case) Description	Case for managing applications for open positions at SAE
(Data Object) Name	Candidate
(Data Object) Description	Information about candidates for open positions at SAE
Reuse Layer	Implementation

Table 1 shows basic details of Hr. Services application with application name, description, organization structure, objectives and reuse layer.

Identifying Case Stages. Used the Stage Designer to add the following six primary stages to your case: Collect Information; Qualify; Interview; Decision; Offer; and Wrap Up.

Figure 2 shows case stages of application is divided in six stages–Collect Information, Qualify, Interview, Decision, Offer and Wrap Up.

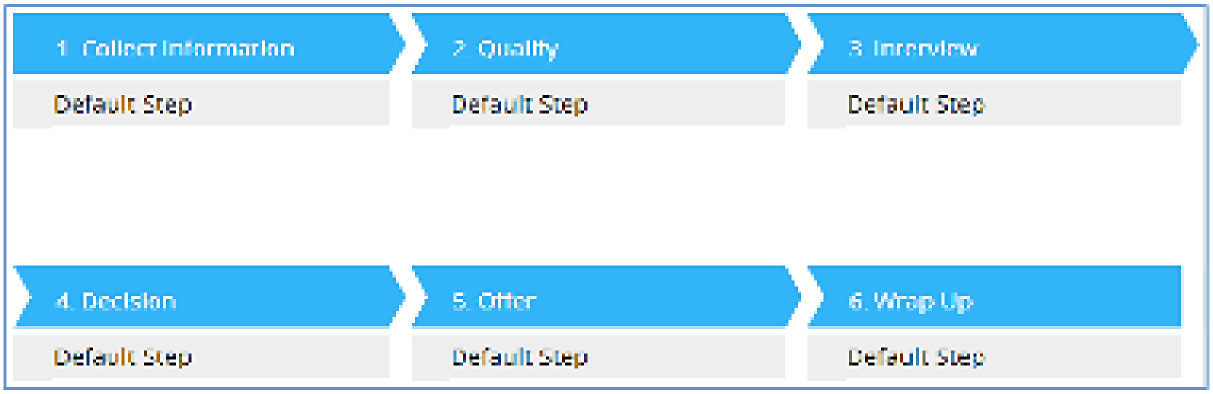
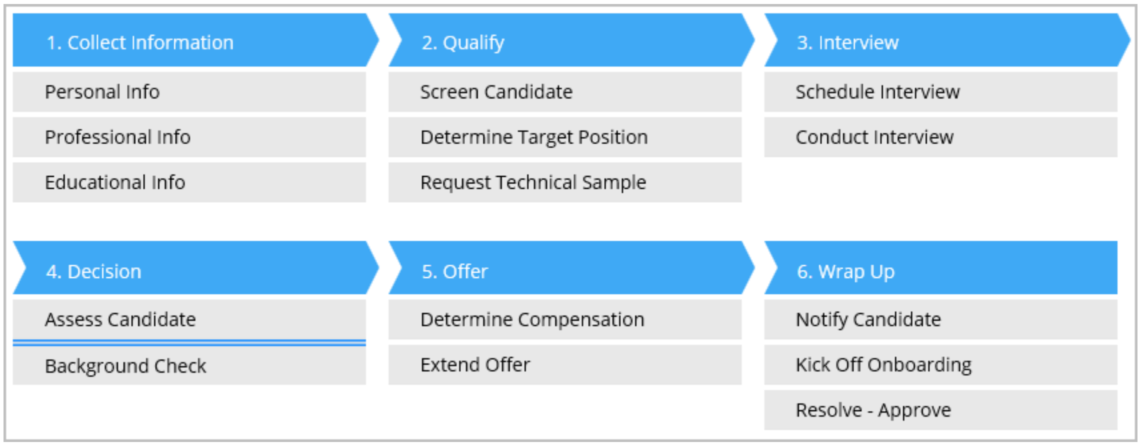


Figure 2: Case Stages of Application

Adding Steps to a Case Stage. Once a case is divided into stages, the next step is to add steps to each stage. Each step represents a specific action or item to complete in order to process–and ultimately resolve–a case.

Figure 3 shows that each step represents a specific action or item to complete in order to process and ultimately resolve–a case.



1. Collect Information
- Personal Info
- Professional Info
- Educational Info

2. Qualify
- Screen Candidate
- Determine Target Position
- Request Technical Sample

3. Interview
- Schedule Interview
- Conduct Interview

4. Decision
- Assess Candidate
- Background Check

5. Offer
- Determine Compensation
- Extend Offer

6. Wrap Up
- Notify Candidate
- Kick Off Onboarding
- Resolve - Approve

Figure 3: Steps to Stages of Application

Data Collection: It describes the process and elements involved in data Collection.

Defining the Process.

Configured Assignments with the Process Outline. Created flow actions for the following steps in the Candidate case.

Table 2 displays list of flow actions created for the assignments with instructions.

Table 2: Flow Actions Steps in Candidate Case

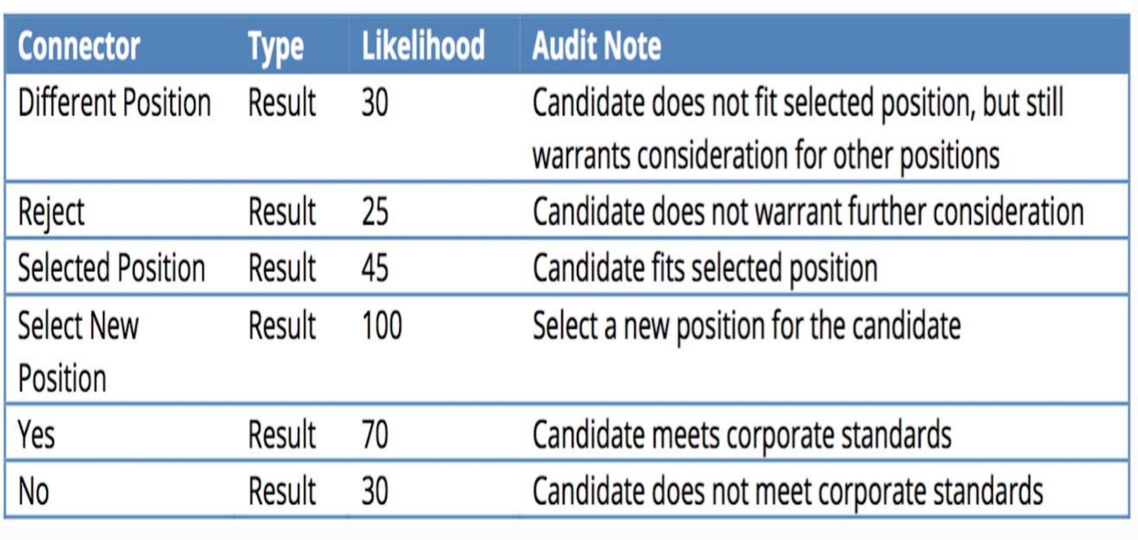
Assignment	Flow Action	Instructions
Personal Info	Collect Personal Info	Enter personal information
Professional Info	Collect Professional Info	Enter work history information
Educational Info	Collect Educational Info	Enter educational background
Screen Candidate	Screen Candidate	Conduct telephone interview
Request Technical Sample	Request Technical Sample	Obtain sample of prior work
Schedule Interview	Schedule Interview	Schedule hiring manager interview
Conduct Interview	Conduct Interview	Conduct interview with candidate

Model Processes with the Process Outline. Created process diagrams for the Determine Target Position, Assess Candidate and Resolve-Approve steps in the Candidate case.

Configured the flow shapes according to the following information.

Table 3 displays details of names of connectors with type, likelihood and audit note.

Table 3: Flow Connectors Likelihoods and Audit Note



Connector	Type	Likelihood	Audit Note
Different Position	Result	30	Candidate does not fit selected position, but still warrants consideration for other positions
Reject	Result	25	Candidate does not warrant further consideration
Selected Position	Result	45	Candidate fits selected position
Select New Position	Result	100	Select a new position for the candidate
Yes	Result	70	Candidate meets corporate standards
No	Result	30	Candidate does not meet corporate standards

Figure 4 shows process diagrams for the Determine Target Position, Assess Candidate and Resolve-Approve steps in the Candidate case.

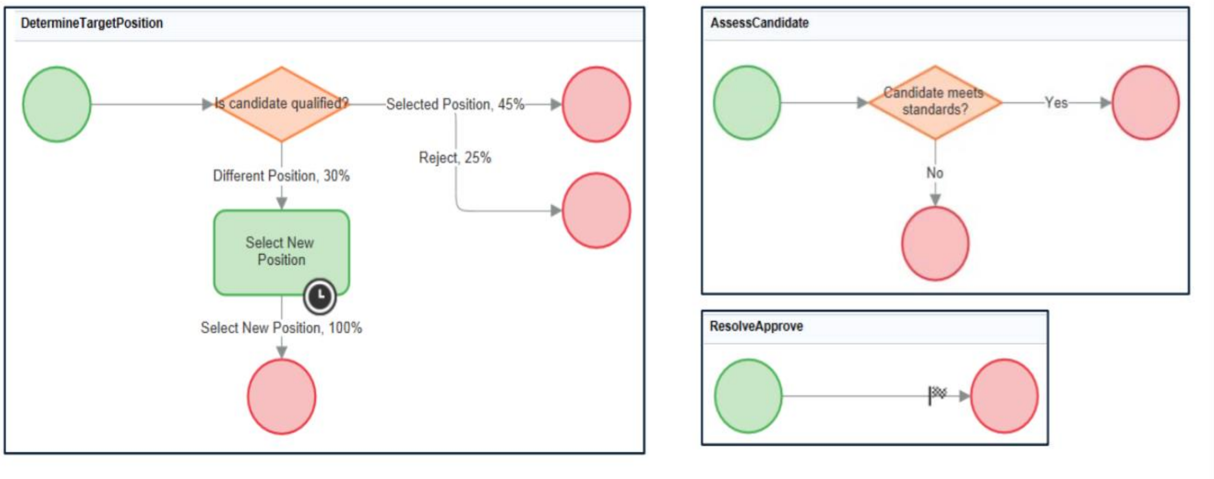


Figure 4: Process Flow Diagrams

Added Specifications to a Process. CollectPersonalInfo flow action: Collect the following information from the candidate:

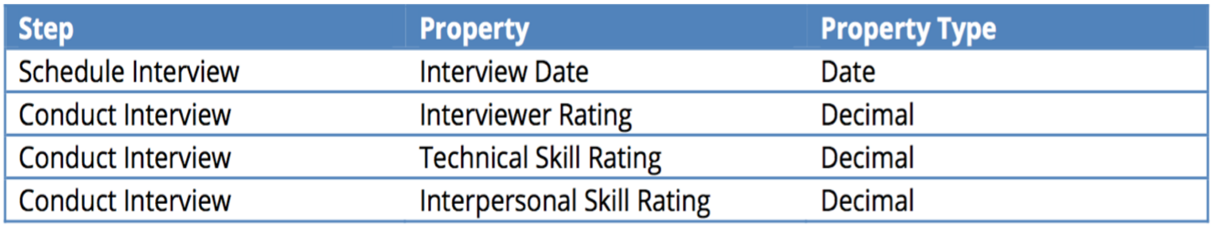
- FirstName : text; required
- LastName : text, required
- DateofBirth: date
- Email: text
- PhoneNumber: text
- Gender: radio button(Male/Female)
- PositionAppliedFor: Drop-down list populated from data table

Defining the Data Elements:

Data Elements. Single-Value Properties: Used the properties panel for the Schedule Interview and Conduct Interview.

Table 4 displays names of assignments that are added to the following properties to the data model.

Table 4: Data Elements

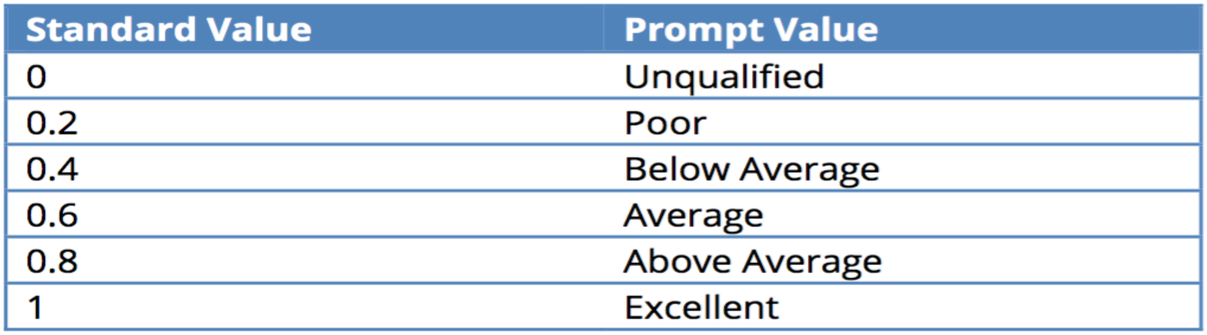


Step	Property	Property Type
Schedule Interview	Interview Date	Date
Conduct Interview	Interviewer Rating	Decimal
Conduct Interview	Technical Skill Rating	Decimal
Conduct Interview	Interpersonal Skill Rating	Decimal

Configured the property form for each of the Interviewer Rating, Technical Skill Rating, and Interpersonal Rating properties so that they include a prompt list, using the values in the following table. Also, configured the property to display as a set of radio buttons by using the pxRadioButtons control.

Interviewer Rating. Table 5 displays Interviewer Rating which has standard values ranging from 0 to 1 with respective prompt values.

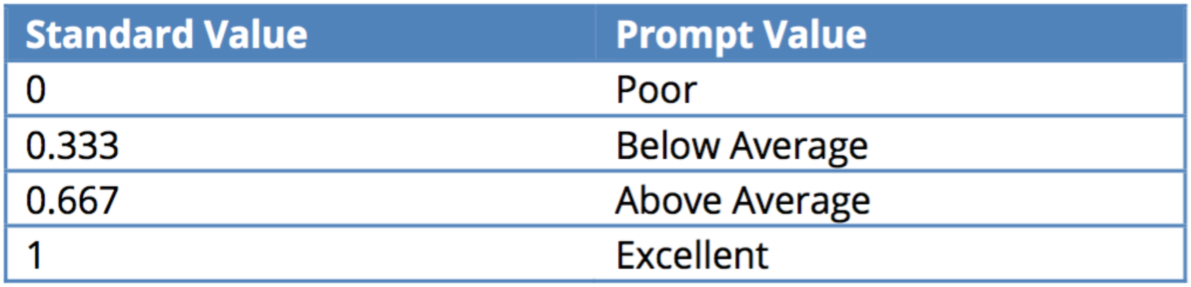
Table 5: Interviewer Rating



Standard Value	Prompt Value
0	Unqualified
0.2	Poor
0.4	Below Average
0.6	Average
0.8	Above Average
1	Excellent

Technical Skill Rating and Interpersonal Skill Rating. Table 6 displays Technical and Interpersonal Skill Ratings which has standard value ranging from 0 to 1 and prompt value from poor to excellent.

Table 6: Technical and Interpersonal Skill Ratings



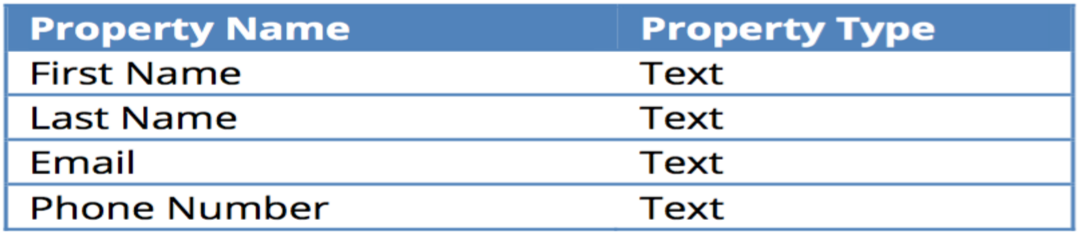
Standard Value	Prompt Value
0	Poor
0.333	Below Average
0.667	Above Average
1	Excellent

Data Modeling.

Property Pages. In the Personal Information step of the Collect Information stage, created a page property named Candidate, which contains the following single-value properties.

Table 7 displays property pages with property names and property type.

Table 7: Property Pages with Property Names



Property Name	Property Type
First Name	Text
Last Name	Text
Email	Text
Phone Number	Text

Setting Values of Data Elements. Added a read-only field for the property Full Name to the Schedule Interview step.

Created a data transform, named SetFullName, to set the value of the property. FullName according to the formula. FirstName + ““+. LastName. All properties are defined on the Candidate page.

Added the data transform to the flow action CollectPersonalInfo as a post-processing action.

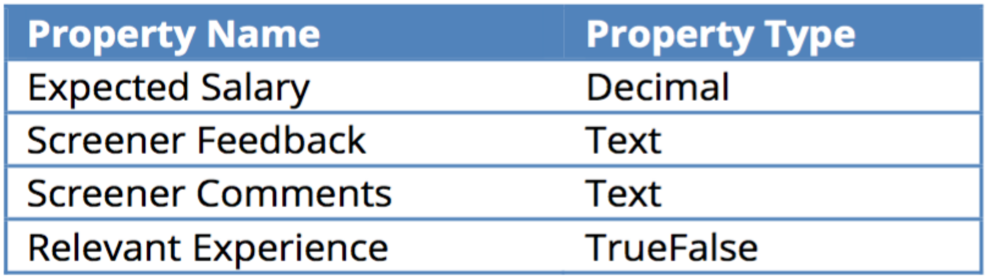
User Interface.

Designing Layouts.

Adding Layouts. Used Stage Designer to define the properties:

Table 8 displays properties with property type which are added to layouts.

Table 8: Properties to Layouts



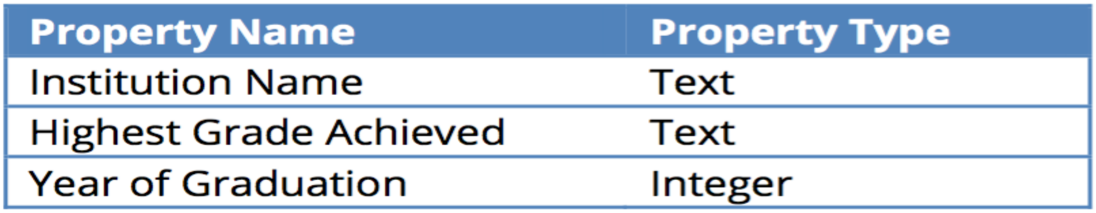
Property Name	Property Type
Expected Salary	Decimal
Screener Feedback	Text
Screener Comments	Text
Relevant Experience	TrueFalse

Adding a Repeating Layout. Used the stage designer approach to create new repeating layouts in the Collect professional info and Collect educational info steps. For the Collect Professional Information action, created a page list named Work History with properties from the table below.

For the Collect Educational Information action, created a page list named Educational Background with properties from the table below.

Table 9 displays properties with property type which are added to repeating layouts.

Table 9: Properties to Repeating Layouts



Property Name	Property Type
Institution Name	Text
Highest Grade Achieved	Text
Year of Graduation	Integer

Formatting Data Elements. Adding a New Control on an Existing Field.

In the professional information step of the collect information stage, added a new property named Skills using the Stage Designer. This property can be defined as text (which will use pxTextinput). Modifed the control to use the Rich Text Editor (RTE).

In the Screen Candidate step of the Qualify stage, modifed the control in Screener Comments to use a Text Area.

In the Screen Candidate step of the Qualify stage, modifed the control for the Screener Feedback field to use a dropdown menu with the following selections:

- Fits the Role
- Requires Additional Training
- Is Not a Fit for the Role

Data Analysis

It describes the elements used in developing reports for data analysis

Accessing Data in the Application:

Data Tables. Used the Data Table wizard, available from the Designer Studio Data Model menu, to create a data table that will contain open positions.

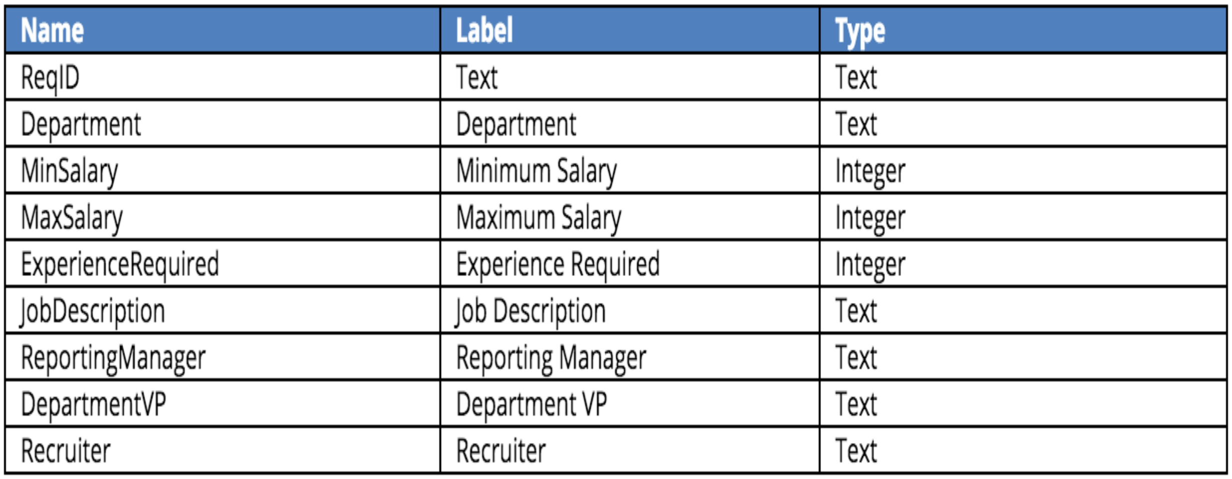
Described the Position table using the following information:

- The rules created in the wizard will be stored in the SAE-HRServices-Data-Position class
- The derived class will be SAE-HRServices-Data
- Accept the defaults for RuleSet information

The results of the data table must be accessible in the application using the two data pages- D_Position and D_PositionList.

Add columns using fields in the following table.

Table 10: Data Tables of Application



Name	Label	Type
ReqID	Text	Text
Department	Department	Text
MinSalary	Minimum Salary	Integer
MaxSalary	Maximum Salary	Integer
ExperienceRequired	Experience Required	Integer
JobDescription	Job Description	Text
ReportingManager	Reporting Manager	Text
DepartmentVP	Department VP	Text
Recruiter	Recruiter	Text

Table 10 displays Data Tables which shows the name, label and type of the property used in application.

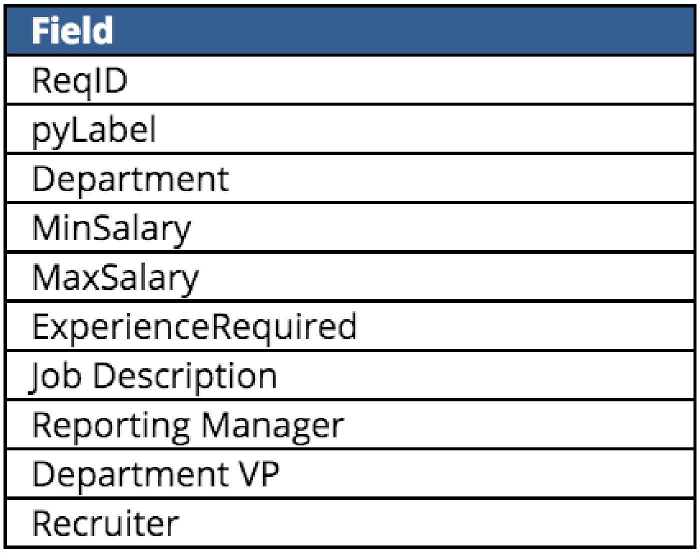
Assessing Data in the User Interface. Configured the PositionAppliedFor field in the Collect Personal Info section to use a drop down control. Use the data page that is created by the Data Table wizard to display the list of all positions. The available positions should have an ID stored internally but should display the label to help the candidate to easily select the position.

Configured the ScreenCandidate section to use a new embedded section. This embedded section should use the data page D_Position to provide page context. The parameter used for lookup should be the property that the candidate has used for selecting the position.

Created a new section within the Screen Candidate section and add the following fields in a dynamic layout–use Inline Grid Triple or Inline Grid Double as the format.

Table 11, Data table shows the names of properties used in Inline Grid.

Table 11: Data Table with Inline Grid



Field
ReqID
pyLabel
Department
MinSalary
MaxSalary
ExperienceRequired
Job Description
Reporting Manager
Department VP
Recruiter

Auto Populate a Page Property. Performed the tasks listed below.

- Added a new page property to the Schedule Interview section and named it Position.
- Configured the Position property to auto-populate position data from the existing data page D_Position, using the property Position Applied For as the value used for lookup. Opened the Position property and selected Refer to a data page in the Data Access Area.
- Used the data page D_Position to source this property and Position set to Applied For Property as the value of its parameter.

Dynamic User Interface. Configured solution to modify the Select Position screen to perform the functionality described in the scenario.

From the process outline for the Qualify stage, expand the Determine Target Position step and then selected the Select New Position Assignment. Added fields for the PositionAppliedFor text mode property and Position page property.

Opened the section and changed the control used for PositionAppliedFor to Dropdown. Configured the listing source to use D_PositionList. Displayed the pyLabel as display field and JobReqId for storing the value for PositionAppliedFor field.

For the included section that displays position details, switched the page context to use a data page instead of the embedded page. This allows values for the position details to be accessed from the data page.

Applied the refresh condition so that the section refreshes when the value in the dropdown changes.

Assessing External Systems. Reusing Data Transform for Background Check.

Completed the Background Check in the Decision stage by accessing existing data objects and a User Interface Section.

The Candidate’s Social Security Number (SSN) is used to perform a background check, so an SSN field needs to be added to the Collect Personal Info step. Also added a flow name and fields to the Background Check step. Fields that will be added to the flow are those returned from the background check transform that is imported.

Once properties that will contain the return values from the background check have been added to the flow, reuse the User Interface Section that was included in the import.

Automating Business Process.

Declarative Processing.

Determine Offered Salary. Configured solution to use two Declare Expressions. For average calculation, used the average function or computed the sum of all three factors and divide by three.

Created the flow action and section for Extend Offer the screen. Used Candidate, Position and Offered Salary as fields to display. Marked all of them except OfferedSalary as read-only.

Skip Screening Process. Configured solution to skip the “Qualify” stage when referred by an employee.

Requested Work Sample for Technical Positions

Added the following fields to the Request Technical Sample User Interface:

- URL
- Sample Code

Automating Business Processes Using Decision Rules.

Directed Flow Processing with a Decision Table

Updated the allowed values for the Screener Feedback property to “Matches the position”, “Matches a different position”, and “Does not match any position”. Created a decision table to direct the Candidate case through the Determine Target

Position process. This table should evaluate the value of the Screener Feedback property, and direct the case according to the following logic:

- If the recruiter selects “Matches the position”, return the result “Selected Position”.
- If the recruiter selects “Matches a different position”, return the result “Different Position”.
- If the recruiter selects “Does not match any position”, return the result “Reject”.

Directed Flow Processing with a Decision Tree.

Created the Candidate Scoring decision tree to evaluate the interview for each candidate, and automatically return No for a candidate who meets any of the following criteria:

- The Assessment Rating is .4 or less.
- The Assessment Rating is greater than .4 and less than or equal to .6, and the Interviewer Rating is .4 or less.
- The overall Assessment Rating is greater than .4 and less than or equal to .6, and either the Technical Skill Rating or Interpersonal Skill Rating is .333 or less.

Advanced Case processing.

Routing Cases.

Routing to Get Approval for Offer

Configured solution to add routing to the Screen candidate step of the Qualify stage. The assignment must be routed to a group-wide workbasket.

Case Stage Configuration.

Identifying the Resolution Stages

Performed the following tasks:

- Created a new alternate stage named Post-Rejection.
- Configured your solution to make sure both Wrap-up and Post-Rejection stages are resolution stages.

Creating an Optional Process.

Scheduled Company Tour

Configured your solution to reference an optional process named Schedule Company Tour in the Interview stage.

Task Based Smart Shapes.

Converted the Notify Candidate step to a process that contains a Send Email smart shape configured to send an email to the candidate with the following information.

- Subject: Offer Letter
- Message: SAE Corp is pleased to offer you a position, effective December 1, 2014.

Updated the Determine Target Position flow with a Change Stage smart shape added to the Rejected connector, to direct the case to the Post Rejection stage.

Creating a Case Level SLA.

Set Goal and Deadline Intervals for a Case

Established a goal interval of 21 days and a deadline interval of 35 days for the Candidate case type, using the Details tab of the Case Designer.

Sprint Cycle Outline.

Iteration Structure:

- Sprints will run for duration of one week.
- Backlog Refinement will be every Tuesday and Friday at 10:30 a.m.
- Retro and iteration planning will be every Wednesday at 2:30 p.m.
- Tasking will immediately follow iteration planning at 3:00 p.m.
- A public demo will be held at the end of each sprint at 11:30 a.m. every Wednesday.
- Stories will be considered “Done” when they have been approved by the project owner and ready for demo in a staging environment.

Deployment Locations.

- All development work will be done on local developer environments and test.
- All development work will be deployed to QA when ready for certification.
- Staging will be used for all feature/update reviews and presentation to Stakeholders.

- No feature/update branches will go to production without project owner review on staging.

Budget

The entire project is completed within the cost provided by the industry. No extra costs were incurred during the development of this project.

Timeline

The total time taken by the project was 3 months. Requirements Specification and Analysis and Design were completed in 1 month. Development and Integration Testing also took 1 month to complete the project in time. Finally, in 1 month, users were trained and took over the maintenance support.

Table 12: Timeline of the Project

Activity	Timeline	Comments
Problem Identification Proposal	September 2015	
Planning	21th September to 26th September	
Analysis	28th September to 2nd October	
Design	5th October to 9th October	
Execution	12th October to 23rd October	
Testing	26th October to 30th Oct	
Closure	2nd November to 6th November	
Final Defense presentation	January 2016	

Chapter 4: Data Presentation and Analysis

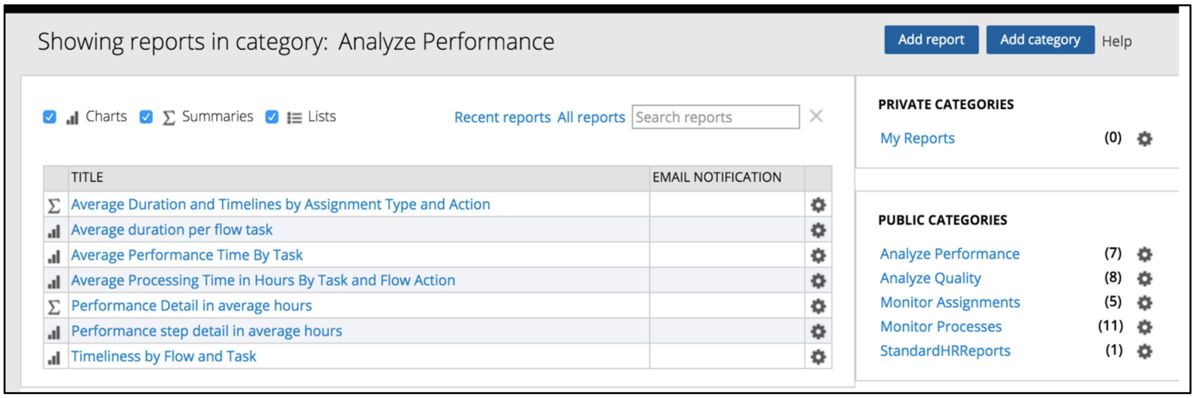
Introduction

This chapter will primarily focus on data representation strategies, and data analysis. Data representation uses a strategy of displaying the data in terms of rows and columns. Data analysis will implement the analyzing the data for validity.

Data Presentation

Out of the box custom reports are generated by pega to analyze performance, quality, monitor assignments, processes, etc.

Figure 5 shows that Pega by default generates reports that can be readily used by users which are in turn divided into several public categories like performance, quality etc. One can easily develop a custom report which are under private categories.



Showing reports in category: Analyze Performance

☑ 📊 Charts ☑ Σ Summaries ☑ ☰ Lists

	TITLE	EMAIL NOTIFICATION	
Σ	Average Duration and Timelines by Assignment Type and Action		⚙
📊	Average duration per flow task		⚙
📊	Average Performance Time By Task		⚙
📊	Average Processing Time in Hours By Task and Flow Action		⚙
Σ	Performance Detail in average hours		⚙
📊	Performance step detail in average hours		⚙
📊	Timeliness by Flow and Task		⚙

PRIVATE CATEGORIES

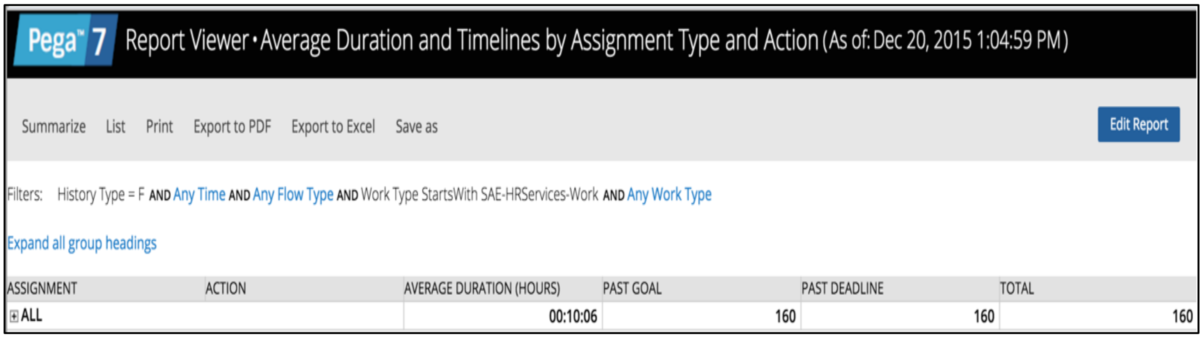
My Reports	(0)	⚙

PUBLIC CATEGORIES

Analyze Performance	(7)	⚙
Analyze Quality	(8)	⚙
Monitor Assignments	(5)	⚙
Monitor Processes	(11)	⚙
StandardHRReports	(1)	⚙

Figure 5: Different Types of Report Generated by Application

Below is one of the report to show the Average duration and timelines by Assignment Type and Action of all cases.



Report Viewer • Average Duration and Timelines by Assignment Type and Action (As of: Dec 20, 2015 1:04:59 PM)

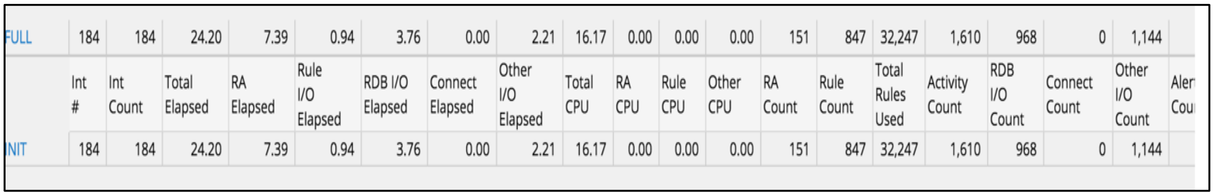
Filters: History Type = F AND Any Time AND Any Flow Type AND Work Type StartsWith SAE-HRServices-Work AND Any Work Type

ASSIGNMENT	ACTION	AVERAGE DURATION (HOURS)	PAST GOAL	PAST DEADLINE	TOTAL
⊞ ALL		00:10:06	160	160	160

Figure 6: Individual Report Generated by Application

Figure 6 Report viewer shows average duration and timelines by assignment type and action.

Performance Analysis. Below results shows the total performance of the application which is out of the box tool.

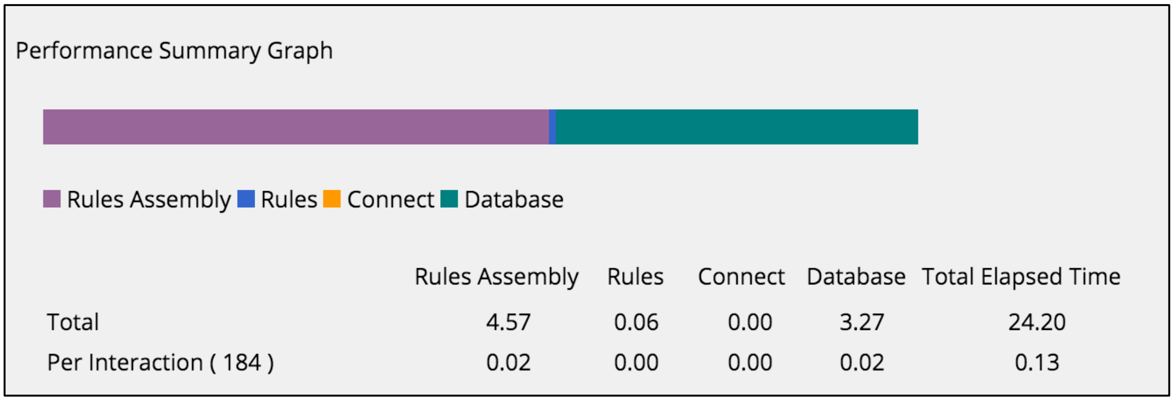


FULL	184	184	24.20	7.39	0.94	3.76	0.00	2.21	16.17	0.00	0.00	0.00	151	847	32,247	1,610	968	0	1,144	
	Int #	Int Count	Total Elapsed	RA Elapsed	Rule I/O Elapsed	RDB I/O Elapsed	Connect Elapsed	Other I/O Elapsed	Total CPU	RA CPU	Rule CPU	Other CPU	RA Count	Rule Count	Total Rules Used	Activity Count	RDB I/O Count	Connect Count	Other I/O Count	Aler Cou
INIT	184	184	24.20	7.39	0.94	3.76	0.00	2.21	16.17	0.00	0.00	0.00	151	847	32,247	1,610	968	0	1,144	

Figure 7: PAL Tool Readings

Figure 7 PAL Tool Readings analyses the performance of the application and it is out of the box tool.

The graph below, the performance summary graph, shows the results in interaction/sec and total time.

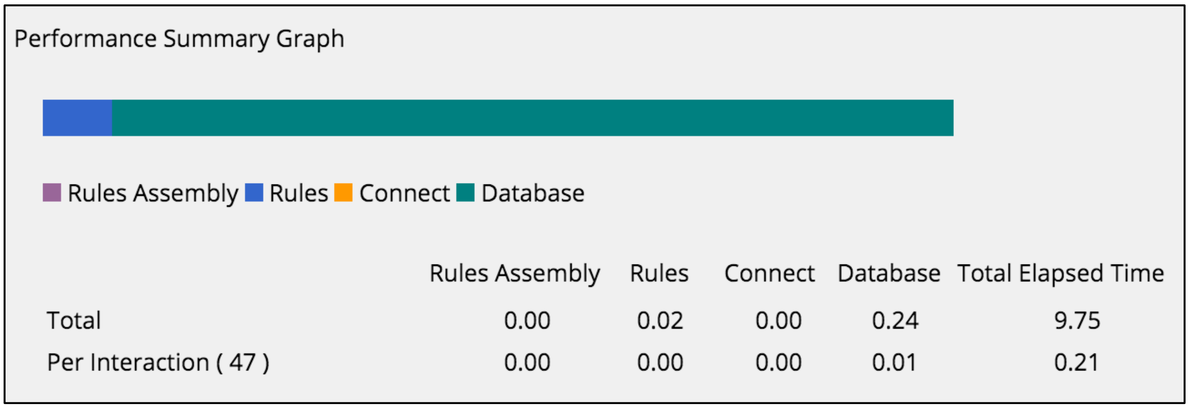


Performance Summary Graph

	Rules Assembly	Rules	Connect	Database	Total Elapsed Time
Total	4.57	0.06	0.00	3.27	24.20
Per Interaction (184)	0.02	0.00	0.00	0.02	0.13

Figure 8: Performance Summary Graph of Legacy Application

Figure 8, Performance Summary Graph of legacy application has a total elapsed time of 24.20 sec.



Performance Summary Graph

	Rules Assembly	Rules	Connect	Database	Total Elapsed Time
Total	0.00	0.02	0.00	0.24	9.75
Per Interaction (47)	0.00	0.00	0.00	0.01	0.21

Figure 9: Performance Summary Graph of New Application

Figure 9, Performance Summary Graph of New applications has a total elapsed time of 9.75 sec.

The average of the total lapsed time 1 = 24.20 sec

The average of the total lapsed time 2 = 9.75 sec

The total elapsed time difference = 14.45 sec

Percentage of time difference = 60% (approx.)

This was because of the manual process involved with the transfer of cases. Now that the process has been automated, let us look at the new readings below:

The total number of interactions 1= 184

The total number of interactions 2= 47

We can observe the drastic fall in all the interactions after implementing the project using Pega.

Data Analysis

Based on the above performance analysis time as been decreased by almost 60% because of automation process.

There has been drastic reduction in number of interaction which are reduced by 75%, which in turn results in increase of performance.

Chapter 5: Results, Conclusion, and Recommendations

Introduction

This chapter will focus on the project results that are obtained after the implementation of the project, conclusion of the project and finally the recommendations that are considered during the implementation of the project.

Results

Implemented the Agile methodology for the automation of hiring process. The results obtained are accurate and precisely define.

Project Questions

1. *What are the time savings after implementing the automation?*

Manual Program: The average of the total lapsed time = 24.20 sec

Automation Program: The average of the total lapsed time = 9.75 sec

Time savings: 14.45 sec

2. What is the cost savings after implementing the Automation?

Cost is one of the important factors that need to be considered in any industry. By implementing this automation, company saved a considerable amount of cost.

Manual Program: $2,400 per month (1.5 hours per day (3 directors) x 20 days x $80 per hour).

Automation Program: $0 per month

Cost savings: $2,400 per month.

3. What is the maximum limit to which the performance can be improved?

The target was to increase the performance by at least 20%.

Manual Program: Total elapsed time which is used as a parameter to test the performance = 24.20 sec.

Automation Program: Total elapsed time which is used as a parameter to test the performance = 9.75 sec.

Improvement in performance = 60%.

5. *What needs to be done to implement the cross browser compatible feature?*

Pega 7 has Inbuilt cross browser compatibility.

Conclusion

Customer was satisfied with the results and they could see the potential of expanding these automations to other departments as well. They were also impressed by Agile methodology where business decisions were changing fast and agile process helped in managing those changes and providing what customers needed with in short span of time. Sales directors were extremely pleased by the amount of time this automation has save.

This study was about automating hiring process using Agile methodology. The company tried to accomplish automatic decision making, send correspondence, and improving the reports. This project was mainly implemented to overcome the issues of time, cost, scalability, and flexibility in changing assignment logic.

Recommendations

Despite the fact that the task was effectively actualized, there were a few hazy areas that required consideration. Following are a few of them:

- Any logic changes in application needed code change and maintenance.
- As it licensed product one should be very careful in planning and implementation.

References

Agile methodology. (2008). Understanding agile methodology. Retrieved from http://agilemethodology.org/.

Agile software development. (n.d). Retrieved from https://en.wikipedia.org/wiki/Agile_software_development.

Amazon Web Service. (2010). *Amazon RDS*. Cloud Relational Database Service: MySQl, Oracle, SQL Server.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . & Thomas, D. (2001). *Manifesto for agile software development.* Retrieved from http://www.agilemanifesto.org/.

Highsmith, J. (2001). *History: The agile manifesto*. Retrieved from http://agilemanifesto.org/history.html.

Highsmith, J. A. (2000). *A collaborative approach to managing complex systems*. New York: Dorset House Publishing.

Manifesto for agile software development. Retrieved from http://www.agilemanifesto.org/.

Pegasystems. (n.d.). Retrieved from http://www.pega.com/.

Pressman, R. S. (2008). Software engineering. *A Practitioner's Approach, 6*(2), 20-24.

Scrum. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Scrum_(software_development).