

## St. Cloud State University theRepository at St. Cloud State

---

Culminating Projects in Mechanical and  
Manufacturing Engineering

Department of Mechanical and Manufacturing  
Engineering

---

12-2015

# Improving Performance of Customer Success Centre using PRPC

Jaya Chandra Reddy Yeddula  
*St. Cloud State University*

Follow this and additional works at: [https://repository.stcloudstate.edu/mme\\_etds](https://repository.stcloudstate.edu/mme_etds)

---

### Recommended Citation

Yeddula, Jaya Chandra Reddy, "Improving Performance of Customer Success Centre using PRPC" (2015). *Culminating Projects in Mechanical and Manufacturing Engineering*. 25.  
[https://repository.stcloudstate.edu/mme\\_etds/25](https://repository.stcloudstate.edu/mme_etds/25)

This Starred Paper is brought to you for free and open access by the Department of Mechanical and Manufacturing Engineering at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Mechanical and Manufacturing Engineering by an authorized administrator of theRepository at St. Cloud State. For more information, please contact [rswexelbaum@stcloudstate.edu](mailto:rswexelbaum@stcloudstate.edu).

**Improving Performance of Customer Success Centre using PRPC**

by

Jaya Chandra Reddy Yeddula

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

December, 2015

Starred Paper Committee:

Ben Baliga, Chairperson

Hiral Shah

Balasubramanian Kasi

### **Abstract**

This capstone project proposal will focus on implementation of improving the performance of customer success center where various applications are integrated, tested, debugged and modified accordingly in order to meet several user, business needs and provide outstanding end user experience.

### **Acknowledgments**

I am introducing you to my capstone project; this project document wouldn't have been possible without the valuable guidance and support from many individuals and organizations.

It gives me an immense pleasure to thank Dr. Ben Baliga, Professor and Graduate Director for Engineering Management Program, at St. Cloud State University for his support and guidance.

I would like to take this opportunity to thank Dr. Hiral Shah, Associate Professor, for Engineering Management Program, at St. Cloud State University without whose support, encouragement, and guidance this project would not have been a reality.

Also, I would like to thank Prof. Kasi Balasubrahmanian for serving on the committee and support throughout the project.

## Table of Contents

	Page
List of Tables .....	6
List of Figures .....	7
Chapter	
1. Introduction .....	8
Introduction .....	8
Problem Statement .....	8
Nature and Significance of the Problem .....	9
Objective of the Project .....	11
Project Questions/Hypotheses .....	11
Limitation of the Project.....	12
Summary .....	13
2. Background and Review of Literature .....	14
Introduction .....	14
Background Related to the Problem .....	14
Literature of the Methodology .....	15
Outline to Build an Application .....	17
Summary .....	18
3. Methodology .....	19
Introduction .....	19
Design of the Study .....	21

	5
Chapter	Page
Data Collection .....	24
Timeline .....	25
Summary .....	25
4. Data Presentations and Analysis .....	27
Introduction .....	27
Data Analysis .....	27
Data Presentation .....	32
Summary .....	42
5. Results, Conclusion, and Recommendations .....	43
Introduction .....	43
Results .....	43
Conclusion .....	46
Recommendations .....	47
References .....	48

**List of Tables**

Table	Page
1. Phases to Build an Application .....	17
2. Timelines of Project .....	25

## List of Figures

Figure	Page
1. Customer Success Center Build Cycle .....	16
2. Agile Methodology .....	22
3. Report Analysis .....	27
4. Open Tickets by Severity .....	28
5. Each Account Ticket Creation .....	28
6. Open Tickets by Status .....	29
7. Ticketing System .....	30
8. Custom Success Centre.....	32
9. Pending Response .....	34
10. Event Analysis .....	35
11. Instances .....	36
12. Selected Instance Screen .....	38
13. Instant Initiation .....	39
14. Instant Submission .....	40
15. Customer Feedback 1 .....	40
16. Customer Feedback 2 .....	41
17. Customer Feedback 3 .....	42



## Chapter 1: Introduction

### Introduction

Pegasystems Inc. was founded in 1983 by CEO and Chairman Alan Trefler, based in Cambridge, Massachusetts. It has several products that focus on customer service and predictive analytics. They are consistently providing high-quality software products in a cost-effective manner to the clients by meeting the deadlines. They also provide the right tools, technologies and skill sets to execute most comprehensive and complex projects with considerable ease. Pegasystems customers are in various sectors such as financial services, insurance, healthcare, communications and media, life sciences, and government (Pega PDN, Home page, n.d.).

This capstone project proposal will focus on implementation of customer success center where various applications are integrated, tested, debugged and modified accordingly in order to meet several user, business needs and provide outstanding end user experience. The applications mainly consists of Dashboard, Reports, Tickets and Instances. This application will be the center to which all customers visit to manage their product and service requests. All customers, partners, employees are all intended use the Customer Success Center. In addition, the proposal also includes the problem statement, objective and nature and significance of the problem.

### Problem Statement

Now Users, partners and employees are raising service requests or tickets (service issues of product) without using customer service Center. They are raising

tickets manually through e-mail or phone about the product issues. This is long and time taking process. They did not even know whether the ticket is assigned to particular engineer to resolve. In addition to that most of the results were not so effective in achieving the desired response times as expected by the customers or users. Customer success Center has become a self-process where user can create service request and check the status of them. Each user will have their own credentials, upon logging it will be presented with dashboard for that user and account. The user account can generate the reports and can manage the instances also.

### **Nature and Significance of the Problem**

Pega customer success center is an application responsible for capturing the any type of requests which are raised by customers/partners/users across their application development life cycle. Generally the issues in a software product, increases in missing the project deadlines. Issues often dependent on other teams within the same organization and sometimes requires the approval of immediate manager. In order to eliminate the time delays in resolving an issue and any miscommunication, Pegasystems has started a global access center wherein the customer can directly raise a request for process of approval to any skilled resource who can resolve the request. Each request is differentiated by unique ID and links with the customer, the same unique ID can be used as a reference for future interactions with the customer. Here the application is accessible to all Pega customers throughout the world and they all can raise a request through the same

URL. Once the request is raised, the request will be routed to Pega customer service agent who will help the customer to resolve the issue. The thing we need to understand here the process of routing to any skilled resource is automated. The scope of project is capture the customer requests and help them to resolve the request in best possible way.

Each customer representative has an access to dashboards for all requests raised by them. In addition customer can generate reports of requests and can check the status of request also.

As end users perform their work—completing assignments and resolving cases—they often need to achieve certain performance milestones. For example, they may be required to respond to a customer complaint within 24 hours. Or they may need to resolve a support issue within 5 business days.

These obligations are called service-level agreements (SLAs), and they outline an agreed-to expectation of service to be provided by a business to their customers. They stipulate the amount of time in which the business intends to respond to the issue. As an item of work—either an individual assignment or an entire case—approaches its established deadline, it becomes increasingly urgent that it be completed. To help communicate this increasing urgency—as a way of enforcing the deadline—we can create a service level.

A service level consists of a goal interval—how much time we prefer an end user uses to complete the work—and a deadline interval—the maximum amount of time the end user can use. Ideally, the business wants its users to complete the work

within the goal interval, and expects them to complete the work within the deadline interval. For example, an online retailer might set a goal of 4 hours to respond to a customer complaint, and a deadline of 8 hours.

### **Objective of the Project**

The objectives of the project are:

- The application promotes self-service use which save time of users/ partners/customers.
- Customers don't have to repeat information they have provided to the agents, everyone can check it in their Dashboard.
- Add perceived value to entering data in the self-service by providing product issue information, which increases in productivity.
- Customers or users or partners fetch, search, can check the status of the tickets, and support requests.
- Every user have their own dashboard where they can see the reports, manage instances and tickets to Increase customer satisfaction.

### **Project Questions/Hypotheses**

- What is the need for Customer Success Center?
- How customer success Center is going to enhance the services?
- What is the need of making customer success Center as self-service?
- What is the feasibility of using other methodologies compared to Agile in this project?
- How it is Implemented and Tested?

## Limitations of the Project

The Awaiting Response page is for tickets that have been responded to by the Pega Cloud Operations support team and are now holding for your response or confirmation. This has to be implemented yet to fulfill the product and know the satisfaction of the work.

- The Awaiting Response page lists all Tickets flagged as a status of "Pending-Customer Response"

Once a business defines a flow for a certain process there are inevitably business exceptions that will need to be handled in real time. Flows should be designed to account for predictable exceptions, but emergency “jumps” should also be strategically placed. For example, say a purchase request has been created and approved. Unfortunately, the approval was rushed and no one noticed that instead of 10 items, 100 items were ordered accidentally. This extenuating circumstance completely interrupts the flow of a standard purchase request, and ultimately how funds are passed around in the system. A special post-approval withdrawal will need to be made and the funds will need to be properly re-allocated as a result of the mistake. Instead of pre-designing a process that exhaustively accounts for every possible scenario that breaks the standard flow, wouldn't it make more sense to have an elegantly triggered exception that allow users to jump outside of the standard flow constraints?

## **Summary**

Tickets are a powerful feature that helps to handle business exceptions easily and efficiently, when alternate stages or Wait shape and dependencies cannot be used. They are also a great tool to build flexibility into our applications. Users raise the ticket where the business exception is happening and reference the ticket where exactly the flow of a case instance to resume when the ticket is resolved.

## **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 Pega Systems and the issues related to the products and manage their Service Requests.

### **Background Related to the Problem**

Pega Customer Service Center puts the customer in the center by helping in: Anticipate customer needs, with intelligent guidance, to simplify the steps for personalized service. Customer success center will connect all the people and systems required to resolve an issue of a product. Focus on customers not their systems, with unified platform, to keep the product always up to date.

Earlier there are many different sources to reach to the product owners about the issues they are facing about the different products. Once they raise any request about the product malfunction then they have to wait till they get response from the engineer. There is a gap between the request and response. To reduce these gap mainly we are proposing this solution.

That way, no matter how customers contact, and which employee they interact with, always deliver a satisfying, efficient and accurate customer service experience by using this customer success center.

## **Literature Related to the Methodology**

Pega RULES Process Commander (PRPC) enhances both developer and end-user experience. Bring straightforward case management design, flexible data reuse, security, mobile and social capabilities, and connectivity enhancements to your applications to drive customer satisfaction.

As shown in the diagram below, application runs on top of two main platform sub-systems:

1. A database system to store the rules and work objects used and generated by Process Commander.
2. An application server that supports the Java EE specification. The application server provides a run-time environment and other services, such as database connections, Java Messaging Services (JMS) support, and connector and services interfaces to other external systems.

Both developers working on Process Commander applications and users working with those applications access Process Commander through Web browser interfaces.



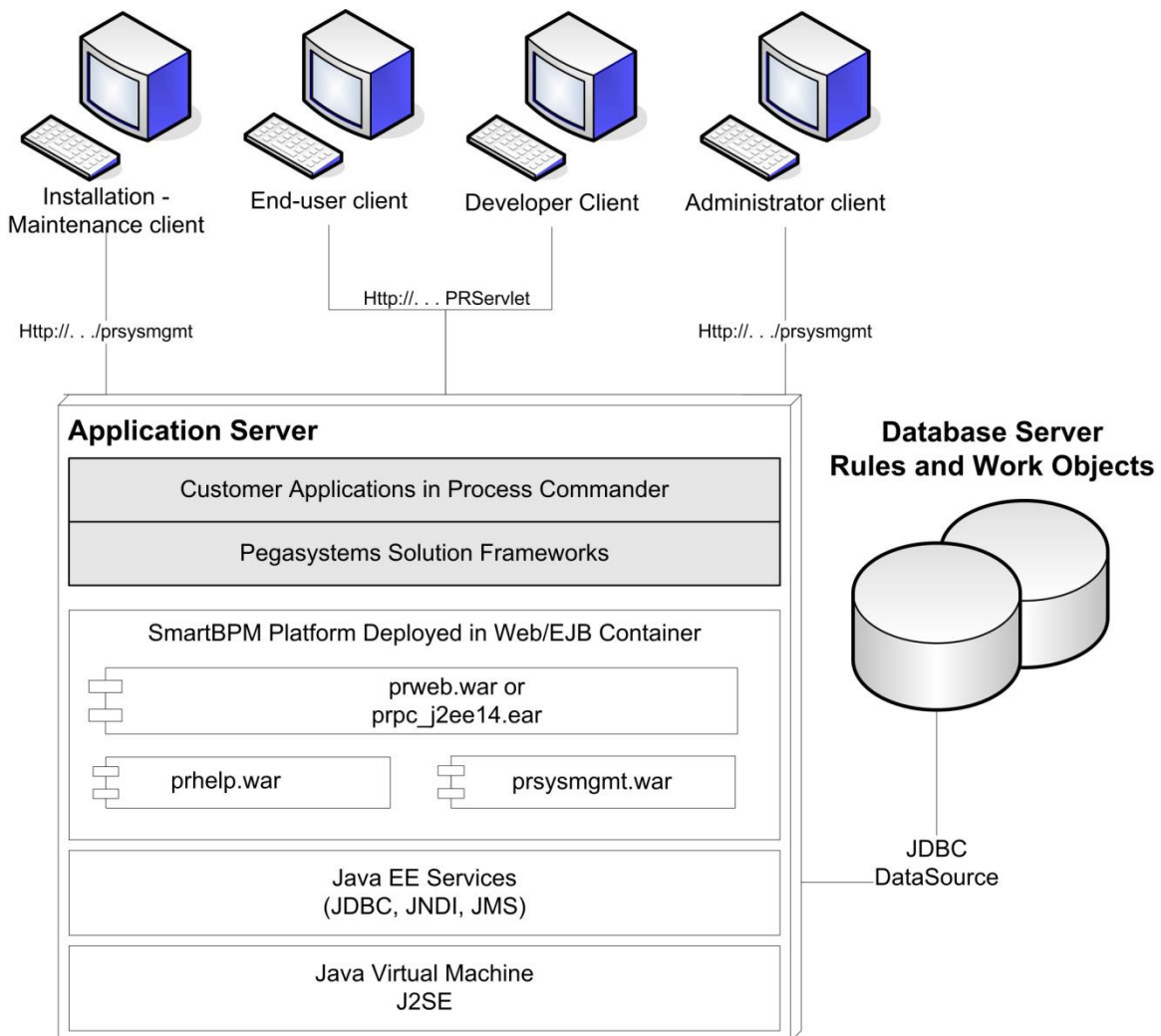


Figure 1: Customer Success Centre Build Cycle

Both developers working on Process Commander applications and users working with those applications access Process Commander through Web browser interfaces.

## Outline to Build an Application

Table 1: Phases to Build an Application

Preparation Phase	Step 1: Environment Considerations
	Step 2: Database Preparation
	Step 3: Application Server Preparation
Rulebase Setup Phase	Step 4: Database setup
	Step 5: Temporary Directory
	Step 6: Rulebase Loading
PRPC Configuration Phase	Step 7: Final Launch

### Preparation Phase:

Step 1: Environment Considerations.

Make sure that that environment meets the requirements:

Hardware, OS, JVM, JDBC Driver, Application Server, Database Server, PRPC, and their respective settings.

Step 2: Database Preparation.

Create and configure a blank database for Process Commander to use.

Step 3: Application Server Preparation

1. Configure the right Database Driver for Database.
2. Configure the PRPC Data Source.
3. Configure other Application Server settings.
4. Deploy the Process Commander application (WAR or EAR).

5. Deploy the online help application and System Management Application.

**Rule base Setup Phase:** Database configuration and Connection includes in the Database setup.

Temporary directory to store the history and cache of the application. The atlast Load the rules into the Database.

**PRPC Configuration Phase:** The Process Commander core application is packaged in to versions to support deployment into Java EE environments.

### **Summary**

The concentration of this chapter has been focused towards making the readers understand more about the background of the problem, literature related to the problem. Background literature review towards the methodology of the project has been explained in a detailed manner.

## Chapter 3. Methodology

### Introduction

The first software development methodologies were hardly methodologies at all, but a free-for-all as organizations struggled to profit from new computer-related technologies. As the industry learned more about developing software, certain techniques for managing and predicting the cost of software development projects came into use.

The methodology that has dominated software development projects for decades is called waterfall. Adoption of waterfall has helped drive down the failure rate of software development projects, but even with rigorous project management and processes, a full 70 percent of software projects using this methodology fail to meet their objectives. To put this in perspective, waterfall software projects have less than half the success rate. Organizations tried to cut the failure rate by insisting on more detail in the requirements and design phases.

Agile scrum methodology is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

One of the most important differences between the agile and waterfall approaches is that waterfall features distinct phases with checkpoints and deliverables at each phase, while agile methods have iterations rather than phases. The output of each iteration is working code that can be used to evaluate and respond to changing and evolving user Waterfall assumes that it is possible to have

perfect understanding of the requirements from the start. But in software development, stakeholders often don't know what they want and can't articulate their requirements. With waterfall, development rarely delivers what the customer wants even if it is what the customer asked for.

Agile methodologies embrace iterations. Small teams work together with stakeholders to define quick prototypes, proof of concepts, or other visual means to describe the problem to be solved. The team defines the requirements for the iteration, develops the code, and defines and runs integrated test scripts, and the users verify the results. Verification occurs much earlier in the development process than it would with waterfall, allowing stakeholders to fine-tune requirements while they're still relatively easy to change (Schwaber, 2004).

Scrum is a project management framework that is applicable to any project with aggressive deadlines, complex requirements and a degree of uniqueness.

A Scrum project consists of a number of time-boxed iterative periods called sprints, which are usually 2-4 weeks long. The sprint consists of a number of components—called user stories—that are selected from a repository of requirements—called a product backlog.

The number of user stories pulled from the product backlog is just enough for completion in the sprint, and have been moved by the Scrum team into a sprint backlog. The team delivers a production-quality configuration at the end of each sprint.

After the sprint is complete, the team demonstrates the result to the product owner, and any other interested stakeholders, in a session called a Sprint Review. Before we get off the elevator, let's have a quick chat about the scrum team.

The scrum team is self-organizing and self-managing, and should be completely autonomous. The recommended team size is between 5-9 resources.

The members of a scrum team cover many different disciplines - analysis, development, testing, documentation, and so on. Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint.

### **Design of the Study**

Agile methodology break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

**Planning:** Planning for the quality assurance requirements and identification of the risks involved in the project are done in the planning stage.

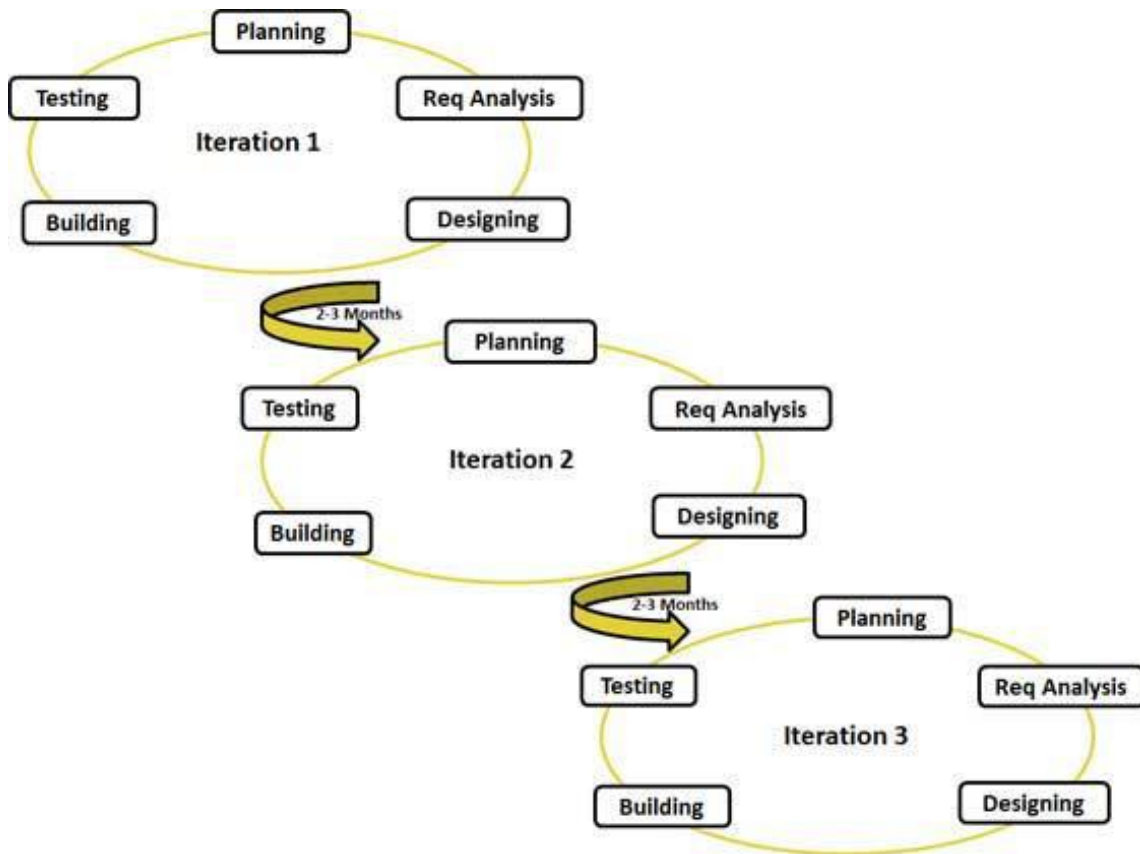


Figure 2: Agile Methodology

**Requirement analysis:** Requirement analysis is the most important stage in the product development lifecycle. It is performed by members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas. The outcome of the technical feasibility study by the business analytics is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

The Supported Server Platform section lists all the supported Server platforms for application:

- Server Hardware
- Load Balancers
- Operating Systems
- Application Servers
- Relational Databases
- JDBC Drivers
- Java Versions

The Client requirements section lists all the supported client configurations for this application.

- Client Operating Systems
- End-User Browsers
- Developer Browsers

**Designing:** The researchers in the Design phase came up with best solution in terms of cost that describes such plan to acquire any hardware and system needed to build or operate the proposed system. Based on the functionality workflows for the business are designed which are best in terms of usage, performance, cost benefit and risk.

**Building:** The actual development of the product is built here. If the design is performed in a detailed and organized manner then there won't be much difficult in building a product.



**Testing:** Testing stage refers to the testing of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards. Once testing is done then the product is deployed.

Customer interaction is the backbone of Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

### **Data Collection**

There are a couple applications that rely on Data collection and categorize incidents, including a service ticketing system and a contact database. Both will ensure that proper parties are notified in the event of a cybersecurity incident.

It is necessary to have an incident database and a ticketing system to capture data, filter through incidents and produce a report with those classified incidents so management and other data owners can look through incidents and figure out how to fix them.

**Categorizing the Incident:** Once all the data is collected, Application should have all the information it needs to categorize the incident. A Category 1 incident typically are considered the most serious threats and let the data owners and others impacted by the threat that a fast and complex response may be needed for remediation. Every category an enterprise uses may require a different level of response, differentiated by the response time and resources needed for action.

After collecting data and categorizing the incident, next steps are handling and managing the incident data and materials in the system.

## Timeline

Table 2: Timelines for the Project

Activity	Timeline	Comments
Literature review Proposal	June 15 2015 – June 26 2015	Completed
Planning & Requirement Analysis	June 29 2015 – July 17 2015	Completed
Designing & Building	July17 2015 – October 23 2015	Completed
Testing	October 23 2015 – November 20 2015	Work In Progress
Production	November 23 2015	
Final Defense presentation	November 2015	

## Summary

The intention of this chapter is to explain the process of the project life cycle using agile methodology, its main deliverables and behavior for each stage and the project background.

Agile software development stresses rapid iterations, small and frequent releases, and evolving requirements facilitated by direct user involvement in the development process. Serena's application lifecycle management tools provide a framework to visualize scope, orchestrate mundane and repetitive development tasks, and enforce process. Unlike agile-specific products offered by agile-only vendors, Serena products are methodology neutral and can be applied equally well to agile as well as more traditional serial development processes, so they can support all the development activities within an enterprise.

Serena's application lifecycle management tools provide a framework to visualize scope, orchestrate mundane and repetitive development tasks, and enforce process.

## Chapter 4: Data Presentations and Analysis

### Introduction

This chapter will focus on the data, interpretation of number of support requests raised by customers based on different criteria for that account.

### Data Analysis

Reports link to the following screen shown in the below format.



Figure 3: Report Analysis

**Open tickets by Severity:** This graph will give an overview for the customer, the tickets opened by them based on severity. Also there is an option to view the graph in 2D as well, this can be obtained from the Chart Actions option present on the top of each graph.



Figure 4: Open Tickets by Severity

**Tickets created for Account:** This graph gives an overview of number of support requests(tickets) raised by customer for last 3 months.

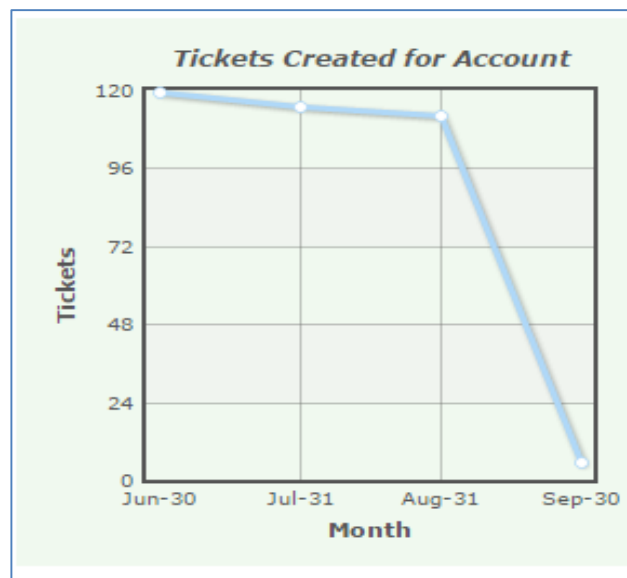


Figure 5: Each Account Ticket Creation

**Open tickets by status:** It is the ticket status by which the customer comes to know the work done or pending for the request raised. This graph gives an overview of the number of tickets present in different status.

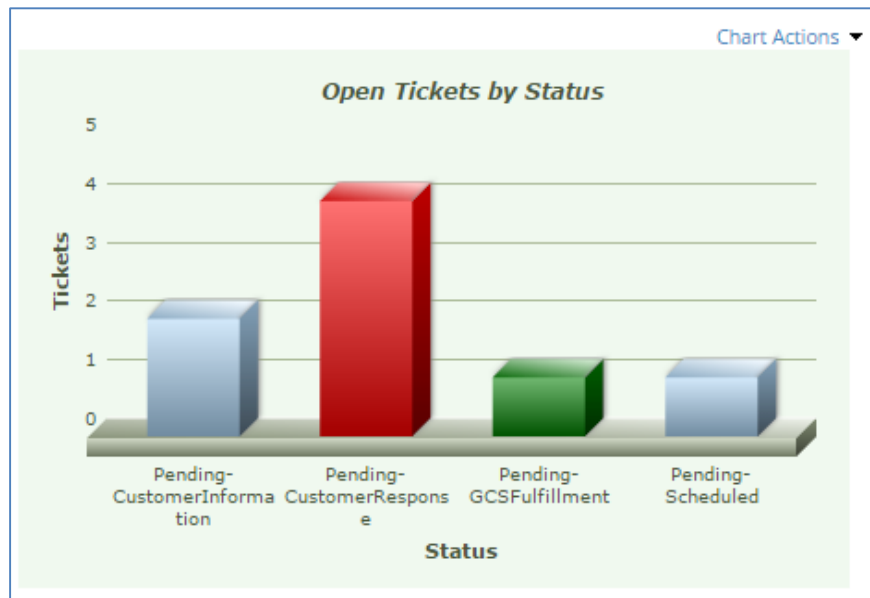


Figure 6: Open Tickets by Status

**Tickets:** In the flows for a case, we also need to define the starting point of the processing that needs to occur once a ticket is raised. We do this by referencing the ticket.

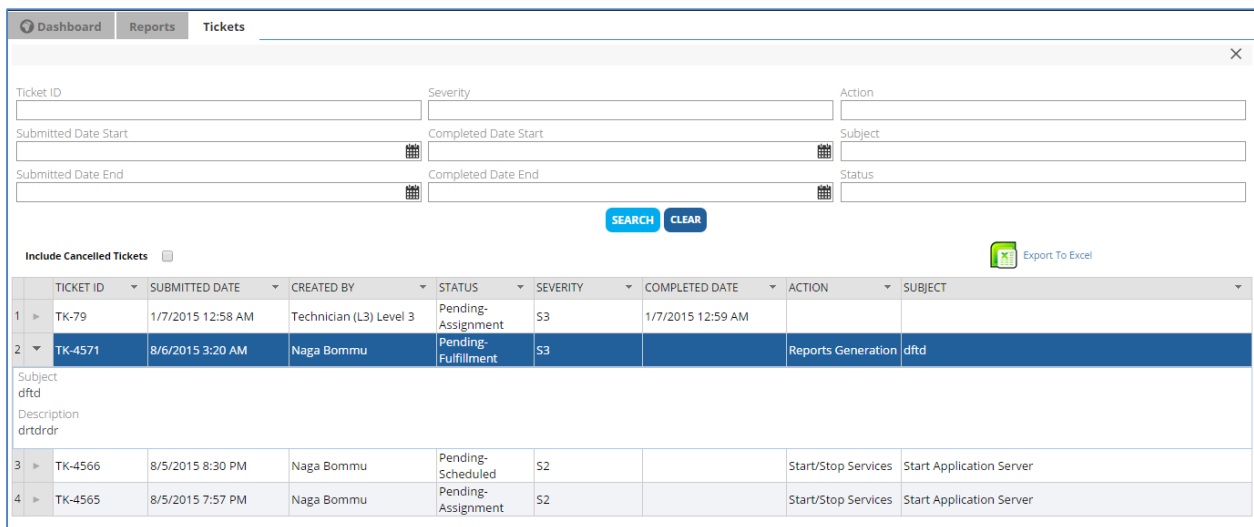
Once a ticket is raised, flow processing stops and resumes at the point in the flow where the ticket is referenced.

In the process modeler, specific shapes have the ticket functionality built into the Properties panel. Use the “Tickets” tab of the Properties panel on the shape we want the processing to resume from when the ticket is raised.

The following shapes provide a Tickets while developing

- Assignment
- Decision
- Subprocess
- Utility
- End
- Split For Each
- Split Join
- Integrator
- Assignment Service
- Most of the smart shapes

The tickets page can be reached by clicking on the 'Tickets' link located in the navigation panel on the left side of the screen. The tickets page allows you to search for, filter, and view all the tickets for your account.



The screenshot shows a web application interface for a ticketing system. At the top, there are navigation tabs for 'Dashboard', 'Reports', and 'Tickets'. Below the tabs is a search form with fields for Ticket ID, Severity, Action, Submitted Date Start, Completed Date Start, Subject, Submitted Date End, Completed Date End, and Status. There are 'SEARCH' and 'CLEAR' buttons. Below the search form is a checkbox for 'Include Cancelled Tickets' and an 'Export To Excel' button. The main part of the interface is a table with the following columns: TICKET ID, SUBMITTED DATE, CREATED BY, STATUS, SEVERITY, COMPLETED DATE, ACTION, and SUBJECT. The table contains four rows of ticket data.

	TICKET ID	SUBMITTED DATE	CREATED BY	STATUS	SEVERITY	COMPLETED DATE	ACTION	SUBJECT
1	TK-79	1/7/2015 12:58 AM	Technician (L3) Level 3	Pending-Assignment	S3	1/7/2015 12:59 AM		
2	TK-4571	8/6/2015 3:20 AM	Naga Bommuru	Pending-Fulfillment	S3		Reports Generation	dftd
Subject dftd Description drtddr								
3	TK-4566	8/5/2015 8:30 PM	Naga Bommuru	Pending-Scheduled	S2		Start/Stop Services	Start Application Server
4	TK-4565	8/5/2015 7:57 PM	Naga Bommuru	Pending-Assignment	S2		Start/Stop Services	Start Application Server

Figure 7: Ticketing System

By default, all your tickets will be displayed. To narrow the list, you can use the search section at the top. Enter one or more values and click the button labeled 'Search'. The list will refresh with only the tickets matching the search criteria. To clear the selections, press the button labeled 'Clear'. You can sort the items by clicking on the header of the column you wish to sort by. Choose the 'Export to Excel' link to export the current visible list of tickets into an excel file. Selecting a row in the list will open that ticket's review screen on a new tab. Selecting the toggle icon to the left of the row will show the subject and description for the ticket.

There are two basic types of routing: Push routing and Pull routing.

Push routing routes the assignment to a single user's worklist, and as such there is immediate ownership as soon as the assignment is created. PRPC can use multiple criteria to select the owner, including availability (if an operator is available or on vacation), the operator's work group, his/her skills, or current workload.

Depending on how the routing is configured, it is possible to route to a substitute operator if the chosen operator is not available. The worklist shows all the assignments for that operator, which can be filtered to show only specific case type instances and can be sorted using any column title. Users can work on any assignment from the worklist.

Pull routing, on the other hand, routes to a workbasket that is shared across multiple users. Ownership does not occur until an operator takes action on it. Most applications use "GetNextWork" logic to select the most urgent assignment from multiple workbaskets. When this happens, the assignment is removed from the



workbasket and is assigned to the operator's worklist. GetNextWork logic is covered in more detail in another lesson. In most of the standard portals, we can pull the next assignment to work with "Get Next Work" logic by clicking on "Next Assignment" button at the top of the portal.

## Data Presentation

Upon logging into the application, you will be presented with the dashboard for that user and account. The sections that appear depend on the access level of the user, and the current effort state of the account and tickets. The actual view may differ from the screenshots included:

The screenshot displays the Pega Customer Success Center dashboard for the user 'John Smith' in the 'Pegasystems - Development' environment. The dashboard is divided into several sections:

- Announcements (1):** A red banner titled "Mailbox Decommissioning" dated 10/8/2014 12:55 PM. The text states: "As part of our ongoing effort of streamlining the communication and improving the turnaround time for support requests, we will be decommissioning support@pegacloud.com mail box effective 31st October, 2014. For clarifications or immediate Pega Cloud support requests, please reach out to Cloud Support and contact details are available at <https://pdn.pega.com/pega-cloud/pega-cloud-support>".
- Recent Cloud Support Tickets for Pegasystems - Development:** A table with columns: ID, DATE OPENED, CREATED BY, STATUS, and LAST UPDATE. It lists 10 tickets with various statuses like Pending-Fulfillment, Resolved-NoResponse, Resolved-Completed, and Resolved-Cancelled.
- Recent Support Requests for Pegasystems - Development:** A table with columns: SUPPORT REQUEST, STATUS, SEVERITY, CREATE DATE, LAST UPDATE DATE, and LAST UPDATE BY. It lists 10 support requests with statuses such as Resolved-Environment, Resolved-Patch, Resolved-Explanation, Resolved-MediaRequest, Resolved-CloudReferral, Resolved-Explanation, Resolved-Withdrawn, Resolved-LocalChange, Resolved-MaintLevel, and Resolved-ServicesReferral.
- Instances for Pegasystems - Development:** A table with columns: INSTANCE, TYPE, ACCESS URL, EVENT STATUS, and ACTIONS. This section is currently empty.

Figure 8: Custom Success Centre

**Announcements:** If there are any announcements, they will appear across the top of the dashboard. Urgent announcements will appear with a red background while informational announcements will appear with a yellow background. This section is collapsible and will only appear if there is a current announcement to be displayed

**Attachments:** Many cases require associated assets ranging from files, screenshots, scanned documents and more. These assets all have different methods of uploading an attachment. By using out-of-the-box tools to easily attach these files to a case, a robust case can be assembled and fleshed out for further processing. In addition, internal security precautions ensure that only those who are allowed access to that case file can edit the attachments.

For example, a legal secretary can attach a scanned legal document, making it into an editable PDF. When another member of the legal team needs to review the case documents, they can easily access the secured case file and find the required document. Later, when the team needs to look for a specific type of attachment within the case, they can do so based on filter criteria, such as all file attachments.

**Tickets Pending Customer Response:** Below the announcements section will be a section to specifically call out the tickets that are waiting on some action from you. Like the announcements section, this section will only appear if tickets exist with a status that is pending action from the customer. Clicking on a row will open the ticket review screen on a new tab. Clicking on the toggle icon to the left will expand the row to show the subject and description of the ticket.

▼ Tickets Pending Customer Response				
ID	DATE OPENED	CREATED BY	STATUS	LAST UPDATE
TK-2927	9/16/2014 3:22 PM	John Smith	Pending-CustomerResponse	Kevin Walsh

Figure 9: Pending Response

**Recent Cloud Support Tickets:** The most recent tickets opened for the account are displayed in this section. Clicking on a row will open the ticket review screen on a new tab. clicking on the toggle icon to the left will expand the row to show the subject and description of the ticket.

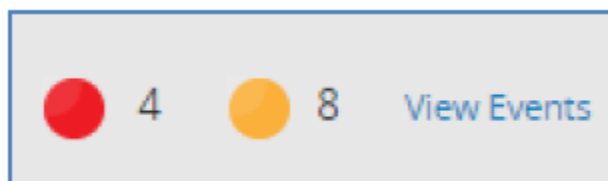
**Recent Pega Support Requests:** The most recent Pega Support Requests for the account will be displayed here. Note that these are support requests opened with Pega Global Customer Support (GCS) and differ from Pega Cloud tickets. GCS requests are prefixed with SR- whereas cloud tickets are prefixed with TK-. Clicking on the row or the toggle icon will expand the row to show more details about the support request. This is a read-only section (Pega PDN, Home, contents, index, glossary, n.d.).

**Instances:** This section includes a brief summary of the instances associated with this account. Clicking the row or the toggle icon will expand the row and show additional details about the instance. The 'Event Status' column will indicate if any events exist for this instance. The potential colors are green (no events), yellow (warning), and red (error). The 'Actions' column contains a button that will allow you to perform self-service functions on your instance.

**Refresh:** The refresh button, located in the top right corner of the dashboard, will refresh the screen for an on demand update.

**Account:** For the majority of users, this section will simply display the name of the account. Some users, however, have access to multiple accounts. In this case the section will be clickable and will show a menu displaying the other valid accounts. Selecting one of these accounts will change the context of the application to that new account.

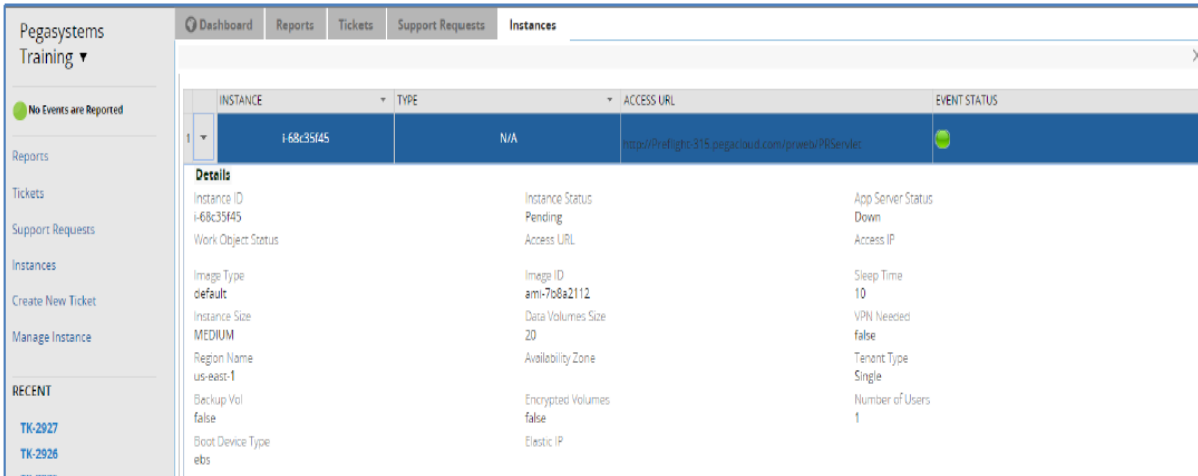
**Events:** Below the account is an icon that will give the status of the instances for this account. If all is well, the indicator will appear green with the text 'No Events are reported'. If one or more events have been reported for any of your instances, the section will change to display red and yellow icons. Beside the icons will be a number that represents the number of total events for each type - error and warning. A link will appear that will allow you to view more details of these events.



INSTANCE NAME	SEVERITY	EVENT STATE	LAST SEEN	EVENT SUMMARY
1 appdb-29-1	Error	New	2014-09-18 20:20:37	Error Connecting to Server ; Failed to connect to service:jmx:rmi:///jndi/rmi://.../jmxrmi [Nested Exception: java.rmi.ConnectException: Connection refused to host: ...; nested exception is: java.net.ConnectException: Connectio
2 appdb-36-1	Error	New	2014-09-18 20:19:46	Error Connecting to Server ; Failed to connect to service:jmx:rmi:///jndi/rmi://.../jmxrmi [Nested Exception: java.rmi.ConnectException: Connection refused to host: ...; nested exception is: java.net.ConnectException: Connection

Figure 10: Event Analysis

**Instances:** The instances page can be reached by clicking on the 'Instances' link located in the navigation panel on the left side of the screen. The instances page gives you a larger view of the Instances associated with this account. Additional details of the instance by clicking on a row or clicking on the toggle icon to the left of that row. As with the instance list on the dashboard, each row will include an event indicator. The indicator will be green for no events, yellow for warning events, and red for error events.



INSTANCE	TYPE	ACCESS URL	EVENT STATUS
i-68c35f45	N/A	https://preprod.315.pegasystems.com/preprod/PServlet	🟢

Details			
Instance ID	i-68c35f45	Instance Status	Pending
Work Object Status		App Server Status	Down
Image Type	default	Access IP	
Image ID	ami-768a2112	Sleep Time	10
Instance Size	MEDIUM	VPN Needed	false
Region Name	us-east-1	Tenant Type	Single
Availability Zone		Number of Users	1
Encrypted Volumes	false		
Elastic IP			

Figure 11: Instances

**Managing Your Instance:** Normally in order to restart their application servers either the customer should login to application server or reach out to Operations in order to get it restarted. In order to avoid them we provided an option in the portal itself to manage their application server. The following actions can be performed

- Start
- Stop

- Restart
- Restart with clear cache

Also there is also an option to do the action selected immediately or in a future date time.

It can be initiated by selecting the 'Manage Instance' link in the navigation panel on the left side of the screen. It can also be initiated by selecting the 'Manage' button on a row of the instance list.

**Select Instance:** The first step is to select the instance you want to manage. This step will be skipped if you make your selection from the instance list as opposed to the navigation panel. On the screen that is presented, choose the instance from the list, enter any special instructions and press Submit. If the instance is not present in the list choose the 'Instance is Not Listed' checkbox. Fill in the details of the app server type and URL. Note that if you choose this option the action will not be completed automatically but instead routed to an engineer to be performed manually.

Source	Instance	Access URL	Elastic IP	Instance Status
PCM	I-0518637D	http://DevTech-30683.pegacloud.com:9080/prweb/PRServlet		Open
PCM	I-6EEAD842	http://DevTech-30685.pegacloud.com/prweb/PRServlet		Pending-Command
PCM	I-6F7F7943	http://DevTech-30690.pegacloud.com/prweb/PRServlet		Open

Instance is not listed

Special Instructions

[SUBMIT](#)

Figure 12: Selected Instance Screen

**Select Action:** Select the action required (i.e. restart application server). Then select when the action to occur. You can choose 'Immediately' or 'At a future date/time'. If the latter is chosen, a date selection box will appear. Check that your selections are correct and press submit.

Ticket TK-2944		Creation Date 9/19/14 2:22 PM	
<b>Instance Details</b>			
Instance ID i-0518637d	Status	Region Name us-east-1	
Availability Zone us-east-1d	Access URL http://75.101.219.155:9080/prweb/PRServlet	Access IP 75.101.219.155	
Vanity URL http://DevTech-30683.pegacloud.com:9080/prweb/PRServlet			
Action Restart		Execute <input type="radio"/> Immediately <input checked="" type="radio"/> At a future date/time	
		Date/Time Requested 9/20/2014 2:28 PM	
Reason Code deployment requires a restart			
<b>SUBMIT</b>			

Figure 13: Instance Initiation

**Confirmation:** On the confirmation screen, verify that the selections are correct. If correct, press 'Submit'. If you need to make a change to one of your selections, you can return to the previous screen by selecting 'Other Actions' and 'Previous Screen'.



Please confirm that you wish to complete the following action:

Ticket TK-2944	Creation Date 9/19/14 2:22 PM
-------------------	----------------------------------

Instance Details		
Instance ID i-0518637d	Status	Region Name us-east-1
Availability Zone us-east-1d	Access URL http://75.101.219.155:9080/prweb/PRServlet	Access IP 75.101.219.155
Vanity URL http://DevTech-30683.pegacloud.com:9080/prweb/PRServlet		

Action Restart	Execute At a future date/time Date/Time Requested 9/20/14 2:28 PM
-------------------	--


Reason  
Code deployment requires a restart

Description  
Please perform the following action:  
on http://DevTech-30683.pegacloud.com:9080/prweb/PRServlet  
This action is requested to be completed Sep 20, 2014 2:28:00 PM

**SUBMIT**

Figure 14: Instance Submission

**Net Promoter Score (NPS):** In order to know the customer satisfaction , after resolving each ticket an email will be sent to the primary contact of the ticket which is shown in the below screenshot.



We value your feedback!

Please [click here](#) to take brief,one minute survey based on your completed ticket TK-4937 related to 'test'.

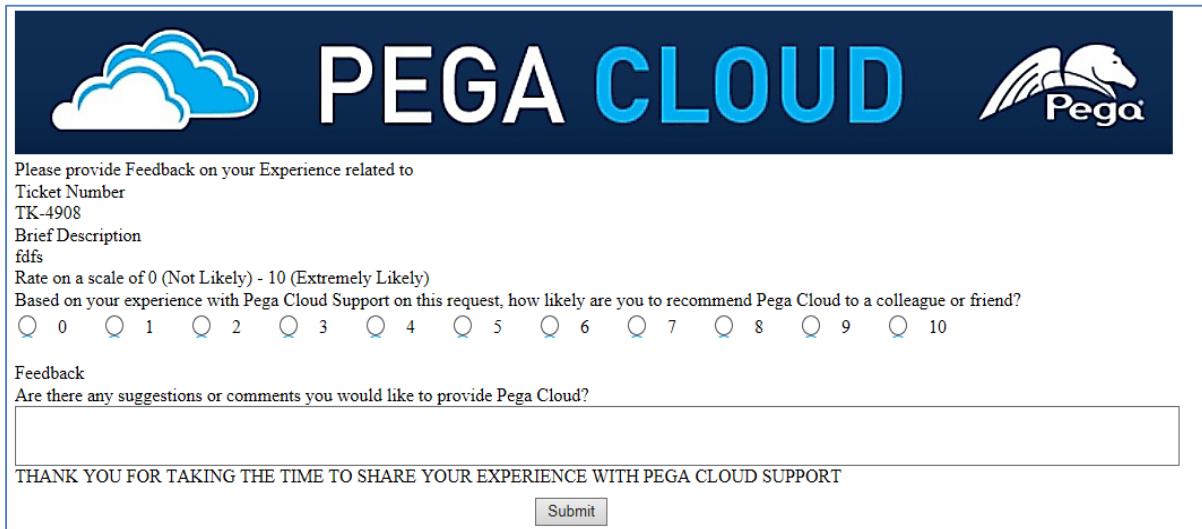
We strive to provide an exceptional customer experience.Your input will help us enhance our service.

Best Regards,

**Pegasystems, Inc.**  
 PEGA CLOUD TICKET PORTAL : <https://success.pega.com/prweb/CSC>  
 PEGA CLOUD TOLL FREE : <https://pdn.pega.com/support/pega-cloud>

Figure 15: Customer Feedback 1

Clicking on “click here” link will open the feedback in new tab of your default browser. Customer can provide their rating from 1 to 10 based on their experience.




The screenshot shows a feedback form for Pega Cloud. At the top, there is a dark blue header with the Pega Cloud logo (a cloud icon) on the left, the text "PEGA CLOUD" in white and blue in the center, and the Pega logo (a winged horse) on the right. Below the header, the form contains the following text: "Please provide Feedback on your Experience related to", "Ticket Number", "TK-4908", "Brief Description", "fffs", "Rate on a scale of 0 (Not Likely) - 10 (Extremely Likely)", and "Based on your experience with Pega Cloud Support on this request, how likely are you to recommend Pega Cloud to a colleague or friend?". Below this text is a horizontal row of 11 radio buttons labeled 0 through 10. Underneath the radio buttons is the text "Feedback" and "Are there any suggestions or comments you would like to provide Pega Cloud?". Below this is a large empty text input box. At the bottom of the form, there is a message: "THANK YOU FOR TAKING THE TIME TO SHARE YOUR EXPERIENCE WITH PEGA CLOUD SUPPORT" and a "Submit" button.

Figure 16: Customer Feedback 2


Based on the rating that customer will select, feedback questions changes. We will track all the ratings and feedback provided by the customer and categorize them into three categories:

- Promoters (9-10)
- Detractors (1-4)
- Passive (5-8)

Based on this we can reach out to the customers who are Detractors and sort out the problems they have.



# PEGA CLOUD



Please provide Feedback on your Experience related to

Ticket Number  
TK-4908

Brief Description  
fdfs

Rate on a scale of 0 (Not Likely) - 10 (Extremely Likely)  
Based on your experience with Pega Cloud Support on this request, how likely are you to recommend Pega Cloud to a colleague or friend?

0  1  2  3  4  5  6  7  8  9  10

Feedback  
What improvements should Pega Cloud make so you'd be more likely to recommend us in the future?

Are there any suggestions or comments you would like to provide Pega Cloud?

THANK YOU FOR TAKING THE TIME TO SHARE YOUR EXPERIENCE WITH PEGA CLOUD SUPPORT

Figure 17: Customer Feedback 3

## Summary

Data presentation and analysis explains about the tickets that are raised by the customer, the instances and how to manage them, support requests raised and also how to achieve customer satisfaction.

## **Chapter 5: Results, Conclusion, and Recommendations**

### **Introduction**

This chapter focuses on providing the final result of the project. Subsequently, the project questions posed before conducting this study are answered briefly. Possible recommendations are made based on the result and conclusion for further possible improvement opportunities.

### **Results**

The Methodology used in the project is Agile. This Methodology is followed in various organization and the percentage of success following this methodology is very high. The present project focuses on customer satisfaction and hence agile methodology was the best option.

The Project was divided into several Iterations and each the results of each iterations where studied and analyzed. Through this analysis, it was highlighted that the Customer success center is a huge success and deployment of this process should be done immediately for effective results.

#### 1) What is the need for Customer Success Center?

Customer service is critical for a company success. The Interaction with the customers in most of the organization is through Email or other Tools, but in order to be successful in business, one should streamline customer support operations.

Prioritizing customer request and resolving them in timely manner is most needed for a product success. And hence the role of customer success center comes into play.

A Responsive UI is often discussed in the context of websites. If we are able to build a responsive UI for a web application we get the benefit that you can access the application from any device, including smartphones and tablets etc. without developing a second (or third) version of the application.

It is clear that Responsive Web Design brings a lot of benefits such as:

Flexibility—the ability to use the same application for all devices and browsers

Cost Effective—because we do not have to invest additional development time to build a responsive UI for multiple devices. We only have to build one application.

Usability—increases because we have same UI for all browsers and devices

Easy to maintain—by making changes on place affects all places through responsive UI.

2) How customer success Center is going to enhance the services?

Customer success center is user centric, wherein each customer's case is assigned to a particular account and a particular engineer. This helps the customer have direct interaction with the specified people who are handling the case and understand where the case stands and the time taken to resolve it.

Application get consistent, modern, and fluid end-user interaction with less time and effort. This radically improves productivity for everyone working on the user interface. This increases our ability to quickly build mobile apps within this model. Designers will be able to add single click process such as sending an email, incorporate cascading approvals, and attach content and so on. There is a many-fold

reduction in the time to create processes involving these components, because they are correctly configured the first time.

Reusable components allow greater regulatory compliance for key processes.

3) What is the need of making customer success Center as self-service?

To Increase customer satisfaction, Better consistency, and improved efficiency, the customer should also be involved and hence the CSC is a self-service process. It is faster and easier for IT and the business to work together to define the high-level process of a case. Use metadata to prioritize and route appropriately, and enabled to respond indirectly or directly to social media posts.

4) What is the feasibility of using Agile Methodology than other methodologies in this project?

There were several recommendation like waterfall and other methodologies to be used in the project, but Agile happened to be best among all the methodology. Since we are dealing with customer satisfaction, utmost care had to be taken in delivering the expectations. Agile methodology helps to concentrate on particular context at one single time, thereby helping the project to have a strong hold on the desired target.

The stakeholders are the ones who have desires, wants, and needs, and are the reason the Scrum team is developing the product in the first place.

Using Scrum on a Pega Project–The Big Picture

Due to its flexible and inherent agility, the Pega development platform has historically aligned well with agile-based development methodologies such as Scrum.

The primary focus of Scrum is on the iterative cycles of refining requirements and delivering functionality.

We have found that the bigger picture of Scrum—the product roadmap and the high-level product backlog—and the retrospectives often fade into the background, obscuring the long-range vision and strategic goals.

5) How it is implemented and tested?

Finally, be sure to draft and amend test strategies. A key differentiator of Scrum is the integration of iterative build and test cycles within a Sprint.

PRPC is flexible enough to support parallel development and test cycles, multiple releases, and multiple versioning.

The process was implemented on several modes and conditions and it turned out to be positive and it yielded good results.

## **Conclusion**

- The customer success center help to increase customer satisfaction rate.
- It improves time-to-resolution and reducing costs.
- Meeting key growth metrics by increasing new customers and reducing customer churn.

Based on the final results of the project, it has been a user friendly, innovative, and a flexible outcome. Also, the project has been appealing to the customers while the company started to establish relations with new additional users.

**Recommendations**

- The customer success center can be built in Mobiles and can be optimized for mobile device browsers. This would help customer have more effective interaction in knowing their case status.
- Multichannel Integration can be created to ease the case solving time.



## References

Pega PDN. (n.d.). *Home, contents, index, glossary*. Retrieved October 24, 2015, from [https://pdn.pega.com/sites/pdn.pega.com/files/help\\_v719/procomhelpmain.htm](https://pdn.pega.com/sites/pdn.pega.com/files/help_v719/procomhelpmain.htm).

Pega PDN. (n.d.). *Home page*. Retrieved June 24, 2015, from <https://pdn.pega.com>.

Schwaber, K. (2004). *Agile project management with Scrum*. Redmond, Washington: Microsoft.