Optimizing Distribution Automation in Unicad Environments

Submitted By

Rudresh Gaur

14MCEC21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Optimizing Distribution Automation in Unicad Environments

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Rudresh Gaur

(14MCEC21)

Guided By

Prof. Ajaykumar M. Patel

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Certificate

This is to certify that the major project entitled **"Optimizing Distribution Automation in Unicad Environments"** submitted by **Rudresh Gaur (Roll No: 14MCEC21)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, have not been submitted to another university or institution for award of any degree or diploma.

Prof. Ajaykumar M. Patel
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr P. N. Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Certificate

This is to certify that the major project entitled **"Optimizing Distribution Automation in Unicad Environments"** submitted by **Rudresh Gaur (Roll No: 14MCEC21)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science & Engineering (CSE) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Sanjiv Bansal
Manager,
LIPA,
STMicroelectronics
Greater Noida (U.P)

Statement of Originality

I, **Rudresh Gaur**, Roll. No. **14MCEC21**, give undertaking that the Major Project entitled "**Optimizing Distribution Automation in Unicad Environments**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Ajaykumar M. Patel

(Signature of Guide)

Acknowledgements

Firstly, I would like to express my sincere gratitude to my adviser **Prof. Ajaykumar M. Patel**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance helped me in all the time of research and writing of this thesis.

Secondly, a special gratitude I give to external supervisor **Mr. Alok Mishra**, Sr. Manager whose contribution in stimulating suggestions and encouragement helped me approaching in right direction to accomplish my work.

Thirdly, I am highly indebted to my colleagues **Mr. Sharad Gupta**, Technical Leader and **Mr. Shoikat Das**, Sr. Application Software Engineer for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

Fourthly, it gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

Fifthly, a special thank you is expressed wholeheartedly to **Dr P. N. Tekwani**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

Finally, I would like to express my deepest appreciation to my parents and siblings who always gave their moral support and maximum help.

- Rudresh Gaur

14MCEC21

Field of Training

Ongoing project completely belongs to software domain. It goes through the various faces of development life cycle from requirement gathering to design and from quality assurance to its deployment. It also follows CMM standards for development as well as its process of development.

Area of Project

Training is being done under the supervision of expertise of computer science domain veterans. It mainly focuses upon software development and automation of manual process.

Abstract

Distribution Management System is designed for the purpose of providing deliverables to its customers securely. These deliverables are called IPs of the organization. Architecture of the system is planned to make products available to its customers and deliver through various infrastructures as per the customer's interests or requirements. On daily basis hundreds of delivery requests are processed through this system in a secured manner. Distribution system has three main features namely DRT, Approval System, UolBE.

To make delivery system more stable, various optimizations were proposed and planned. This thesis covers optimizations achieved in Approval System and UolBE. Distribution Dashboard, a subsystem of Approval System was enhanced which consists of resolving few of bugs and new features like creating Ant script to automatically create project war for deployment, session management, make application IE11 compatible, introducing single sign-on and removal of application login. Later on few of the customer feedback were covered in subsequent product release.
In continuation, a new module Reporting feature was also developed for analysis purpose of requests processed in certain time interval and based upon various search parameters. This module also provides facility to export result of query in CSV format using content disposition method.

Next optimizations were achieved by enhancing delivery mail templates using HTML, CSS and process them and development of a Java mail API. Main features were multiple attachments, mail contents reading as string and reading from a file.

Final work of this project includes incorporating UptPlus size API for incremental SFT delivery.

Although distribution team has achieved a significant level of automation still due to policy or infrastructure changes there is need to focus on system stabilization for which some proactive actions are being taken. Scope of this thesis is to focus upon enhancement of Approval System and UolBE components of Distribution System.

Abbreviations

IP	Intellectual Property
CMM	Capability Maturity Model
DRT	Distribution Request Tracking
UPTPlus	Unified Project Tool+
SOL	Service On-Line
UolBEplus	Unicad Online Backend Plus
LSF	Load Sharing Facility
DR	Design Responsibility
DP	Design Package
PL	Product Line
MVC	Model view Controller
DTD, XSD	Document Type Definition, XML Schema Definition
HTTP	Hyper Text Markup Language
QA	Quality Assurance
CSS	Cascading Style Sheets
EJB	Enterprise Java Beans

Contents

Certificate iii

Certificate iv

Statement of Originality v

Acknowledgements vi

Field of Training vii

Area of Project vii

Abstract viii

Abbreviations ix

List of Figures xii

1 Introduction 1
- 1.1 Project Definition and Overview . . . 1
- 1.2 Introduction . . . 1
- 1.3 Project Scope . . . 3
- 1.4 Objective of Project . . . 3
- 1.5 Terminology . . . 4

2 Literature Survey 5
- 2.1 Existing System . . . 5
  - 2.1.1 IPs . . . 5
  - 2.1.2 DRT . . . 5
  - 2.1.3 Approval System . . . 8
  - 2.1.4 UolBEplus . . . 9
  - 2.1.5 LSF . . . 10
- 2.2 Reasons for Optimization . . . 10
- 2.3 Glassfish Server: Application Deployment . . . 12
- 2.4 LDAP . . . 12

3 Technological Environment 13

4 Distribution System optimization: Distribution Dashboard 15
4.1 Implementation Details . . . 15
4.1.1 Analysis: . . . 15
4.1.2 Design Level Optimizations . . . 18
4.1.3 Feature Enhancements : Release v1.7.3 . . . 21
4.1.4 Feature Enhancements : Release v1.7.4 . . . 21
4.1.5 Feature Development : Release v1.7.5 . . . 22
4.2 Test Cases . . . 24

5 Distribution System optimization: UolBEplus 25
5.1 Implementation Details . . . 25
5.1.1 Mail templates Enhancement . . . 25
5.1.2 JAVA Mail API . . . 27
5.1.3 Incorporate UPT+ Size API in SFT Incremental delivery . . . 29

6 Discussion and Conclusion 30

7 Planned Optimizations in Approval System for Q3-2016 32
7.1 DRT Approval Admin tool migration . . . 32
7.2 FAQ for DRT . . . 32

Bibliography 33

List of Figures

1.1 System Process 3

2.1 DRT web interface 6
2.2 DRT state presentation 7
2.3 Request Approval mail 9
2.4 Failure Pareto 11

4.1 Legacy Admin Dashboard flow diagram 17
4.2 Legacy user Dashboard flow diagram 18
4.3 Optimized Admin Dashboard flow diagram 19
4.4 Optimized user Dashboard flow diagram 19
4.5 Optimized admin Dashboard web interface 20
4.6 Optimized user Dashboard web interface 20
4.7 Form of Reporting feature 23
4.8 Result based on search parameters 24

5.1 Form of Reporting feature 26
5.2 Result based on search parameters 27

6.1 Distribution System success timeline 30

Chapter 1

Introduction

1.1 Project Definition and Overview

Data management team is expertise in distributing libraries to its customers internal or externals. Distribution system has two types of customers to deliver libraries (IPs)-

- **Internal**
- **External**

Products distribution is done through a web interface called DRT and each request is uniquely identified as DRTID containing list of deliverables which uses various infrastructures to send to its end customers. These requests are given to UolBE system in terms of xml files. these files are dumped to a area where they are processed and delivery is sent successfully to customers. Various medias are used to send delivery such as FTP, DesignSync, SOL+ etc.

Continuous efforts are put by team to make the system reliable and stable.

1.2 Introduction

Distribution System belongs to Data Management team of ST Microelectronics. Upon customer requests or feedback enhancements are exercised in automation and support to existing service is optimized. Referring to fig 1.1 that Distribution mainly has 3 features

This project comes under Data Management team of T&DP division. Distribution

System aimed to automate product delivery system with zero manual support unless it is needed.

DRT Distribution Request:

This is a web interface where requester raised the request to send products to customers. It has been written in [1]php language and for database it uses [2]sql database. Each request is assigned a unique id (numeric number). It basically is a form which has multiple entries like requester name, project name, deliverables list, media, downloaders etc. It has various facilities like search and update details. Once entries are done and saved, a unique id is generated for that request. Then it goes to Approval system to decide whether it will go in processing delivery or not.

UolBEplus:

It comes in picture when request is approved and UolBEplus takes care from downloading products from repository to end user delivery. It works on xml one for each product. Various modules are written for each type of media in [3]Perl script. A specific directory structure is maintained for all operations. Finally a delivery mail is sent to customers and to distribution system after its processing and in case some error occurs then it sends a filed mail to requester send distribution system.

Approval System:

This subsystem of distribution system has further three classifications as below

DRT Approval Admin tool: This system contains a list of approvers which belong to approval list for various DR divisions and Product Lines. One new enhancement has been done is inclusion sending approval mail Product Line based also because lot of database structure used to change due renaming of departments which blocks the automation.

Approval Algorithm: This algorithm takes care of to whom approval is send and if PMs are the requesters then request should be directly approved. If selected products belong to more than one DR divisions then mails to approve is sent to PM and notify DR divisions. Else if all products belong to only one division (IO or Standard Cell or memory) then ask for approval from respective division.

Distribution Dashboard:

This subsystem is used to track the request approval and maintains its states. It mainly has two interfaces one for approvers and other for administrator. Authorized approvers visit approval page dedicated for them which contain details of DRTs in pending approval and recent requests. For each request approvers take action either to approve or to reject the delivery. Admin page tracks the states of requests such as submitted, pending approval, approved and failed. It also contains a form to log reasons for failed requests.

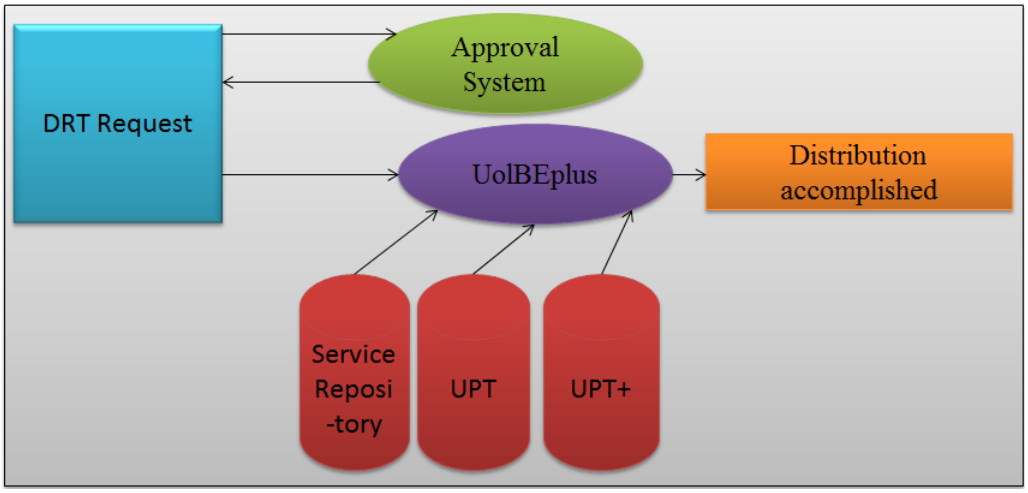

Figure 1.1: System Process

1.3 Project Scope

It automates the process of delivering the IPs to its respective customers. All the deliverables which are to be sent to its internal or external customers are listed in a DRT are delivered by distribution automation system. Various types of requests like Un-publish products, PPOD products, SFT are processed and finally end customers may ask to get delivery using various infrastructures and also can use many types of media like FTP, SOL+, DesignSync etc.

1.4 Objective of Project

Project is to be optimized in a way so that it could reduce completely avoid manual support which may block automation. Main objective of the project is to deliver the products as soon as possible, secured and complete. From development perspective it has been considered that if any change request or enhancement is required that can be

implemented easily and with least changes in other components.

It goes through a rigorous test and quality assurance for robustness and reliability. QA environment gives the system a live or real time flavor which ensures if system is good in QA then it is going to be aligned in production environment.

1.5 Terminology

Automation: Automation is basically a process which consists of various fixed number of steps to achieve a task. These are-
Build: Build is the task of converting code snippets into independent artifact that can be run on a computer by itself. It involves many distinct functions:

1. Version Control: The version control function is used for tasks such as creation of workspace where actual code files and related files can be stored, base lining and notifying. It allows build to run in a proper environment that is needed for success of the code.

2. Code Quality: It is also called code analysis that actually makes sure that a developer has not coded against rules of organization and followed all standards required for a production quality code.

3. Compilation: Compile files to create executable directly when required and then make platform independent executable.

Chapter 2

Literature Survey

2.1 Existing System

2.1.1 IPs

IP is abbreviated for Intellectual Property. These are realized products of the organization which are ready to deliver products to its customers. For a system to run its needs few essential components such as processor, RAM, ALU etc. hence to say that all are fabricated for computation purpose. Electronics fabrication and CAD design a system is realized on a single chip called IC. To design these blocks different divisions work independently. These design blocks are called IPs.

Mainly IPs are categorized in two types :
Soft IP: When customer asks for a soft IP, dedicated designs it in the form of a HDL (Hardware Description Language). In simple terms it can be understood a program code or in the form of a structured logic gates.

Hard IP: Embedding HDL in to actual logic gates and realizing into layout. Layout shows abstract level while once again layout is realized into GDS which is corresponding text format.

2.1.2 DRT

Distribution Request Tracking is a web interface to deliver libraries, which can be used for current and future projects be it internal or to external users, developed by T&DP

division of organization. To understand please refer fig 2.1 each of the requests to deliver a set of libraries is tracked by Distribution Team from the submitted state of request till its processing.

Requester just needs to put a request in the system using interface by selecting project specification, to which customers the libraries are to be delivered. Rest of the task like how to process, how to get these libraries, on what infrastructures this request is to be processed all these activities are handled by Distribution team. The distribution system still having some manual tasks to be done for few types of deliveries hence is the target is to fully automate the system and optimize its process of delivery. Human interventions may not be just because of lack of development but due to some policy and agreement reasons which are confidential to organization.

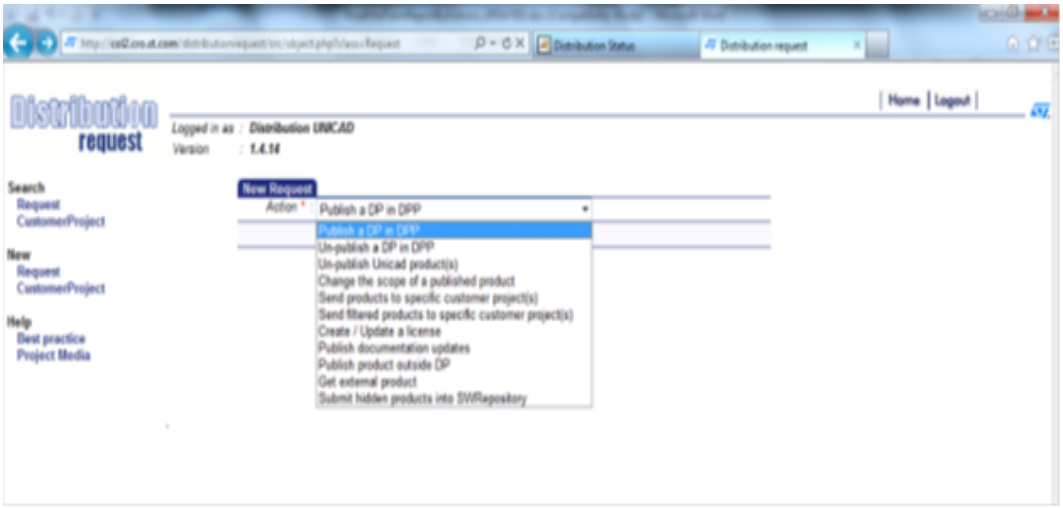

Figure 2.1: DRT web interface

Any request starts by choosing an action which infers which type of delivery and on which infrastructure it will be sent to make it available to its customer. Action is essential and basic selection before putting a request.

Refer to the fig 2.2 it has various states against state filed in DRT. Below are the states described which a request goes through in the distribution system-

Submitted: This is the very first state which any delivery request comes in. It con-

firms that the request is inserted in the system to process.

Assigned: Its a notification to requester that a new request has arrived in the system to process. If it is manual then human intervention is required and after taking action accordingly request is moved to next state or other. On the other way if it can be handled by automation then no need to move request to other manually.

Pending Approval: Before delivery each of the libraries in request has to go through the checks that if it lies under policy and agreement with our customers. In abstract, an approval mail goes to PM of organization who is authorized to approve or rejected any request. Here automation waits a request to come into approved ie; next level of processing.

Pending Support: If any issue arises or approver asks to approve request then all deliveries are moved to approve state and DRT request is moved to pending support. A mail is also attached with the request.

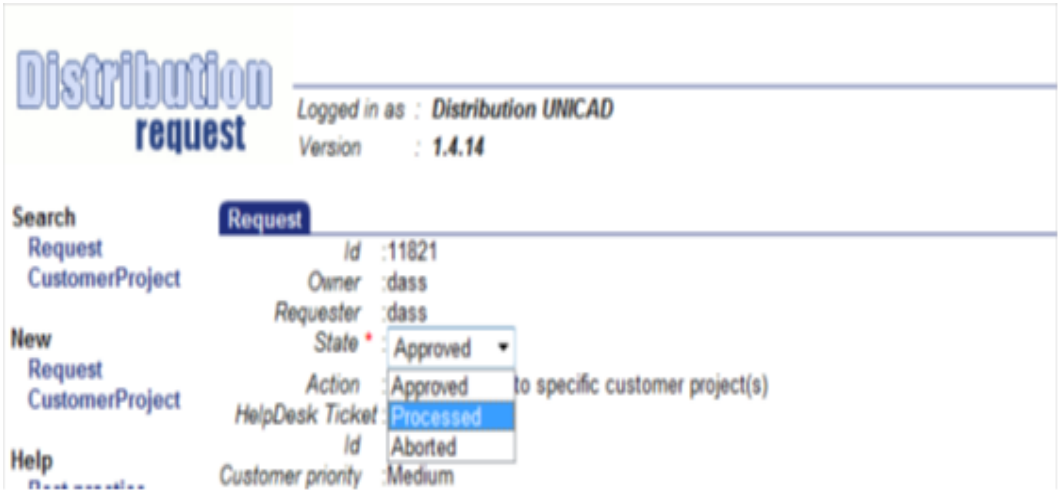

Figure 2.2: DRT state presentation

Approved: An approval mail may be sent by automation to more than one approvers. If any one of them approves request then automation starts working on delivery to process it as soon as possible.

Processed: Here each of the products in delivery is processed and ready to deliver.

This may be done through a mail having download link from servers or simply informing customers that products are available at a requested location and are ready to be used. Which media is to be used to deliver libraries depends on customers on which infrastructure it asks to deliver say FTP, DesignSync, SOL, SOLPlus etc.

Cancelled: If DRT is put wrongly and not meeting standards then it is moved to cancel state.

Aborted: there can be multiple scenarios when request has to be moved to aborted state.

Recently, these states are revised and now they have well-structured and streamed flow of request which request goes through as below diagram

In overall processing, components of distribution system play defined role to process DRT request successfully. These components are UolBE, Distribution Dashboard, Approval system, DRT are designed to give front-end and back-end support as the role defined.

2.1.3 Approval System

Approval is decisive phase for delivery of products. It solely depends on PMs of divisions and organization to decide whether this request should be processed or it is to be cancelled or aborted. To get an approval from PM or approvers an automated and well-structured mail is sent to approvers. They go through mail as shown in fig 2.3 and check if all products in delivery can be sent then they approve else request is rejected. Here Distribution Dashboard application comes into the picture.

Algorithm: Recent change is done in approval algorithm. It checks for if products from more than one Product line or DR then mail are sent to PM else send to a particular DR/PL. If requester is in approval list then request is auto approved. If requester is PM then request is auto approved. For FTP, Webgen and License requests only PMs are authorized to approve.

Hello all,

We need the confirmation of one of the group managers to proceed with treatment of this **SOL** request.
Please click the below link to Approve/Disapprove

Approve or Reject!!

If you cannot access the link directly please copy this link in your browser window
http://xxxxx.dlh.st.com:12070/DetApproval/ApprovalLink.jsp?Id=10836&Count=1

Requestor: Seema Garg
Customer Project: Accordo2

UPT deliverables Labels:

Label	UPT CURRENT	DESIGN RESPONSIBILITY	TYPE
DA24VASDHP_33_C40LP@1.0beta1@20121207.0	PreliminaryMAT10/submitted	STMicroelectronics/TR&D/CCDS/PIMDS/Mixed Signal Core IP/Configurable IP	DAC

Downloaders: Manish SINHA , manish.sharma.dlh
CCList: Rakesh MALIK, Chandrajit DEBNATH
Rationale: Preliminary package release.
To view details of the request see : http://col2.cro.st.com/distributionrequest/src/object.php?class=Request&key=10836

Thanks and Regards,
Distribution Team

Figure 2.3: Request Approval mail

Another feature is Approval Admin tool which takes care of who the approvers are going to be present to approve/reject a request. One component distribution dashboard was developed to take delivery approval and track status of requests.

2.1.4 UolBEplus

This tool provides back-end support to service a delivery of products. It comes into picture once a DRT is approved. It processes product xml and finally packages are transferred to customers. UolBEplus has reads product details then it sets environment and runs for different type of Medias discussed below-

DesignSync

Products in delivery are sent to this server and access is granted to access only to authorized customers.

SOL and SOL+

SOL is for internal customers products stored at a location so that customers can use it just by their reference. SOL+ is portal which facilitates users to downloaders to their local disk.

FTP

This is an external delivery category. It downloads data from repository and package is sent to customer in mail with product url on server.

WebGen

It is similar to FTP but only single package is sent to customer.

HDD

This is used to be manual delivery. But now it has been automated except copy to disk cant be automated is a limitation in automation.

PPOD

Products are uploaded to /sw/unicad standard area for all products.

Filter

Products are filtered keeping customer service level agreement in consideration.

2.1.5 LSF

It can be understand that you need to execute a big task but if you run it locally then it may crash or hang the server. Hence, as a solution a large computer farm is dedicated to provide this facility.

2.2 Reasons for Optimization

System is automated to process delivery well in time. There are some issues which occur accidentally and impact/block the automation

1. Filter error
2. Product download failed
3. User Input
4. DP infra
5. SFT issue
6. LDAP

7. Distrib infra

8. IT infra

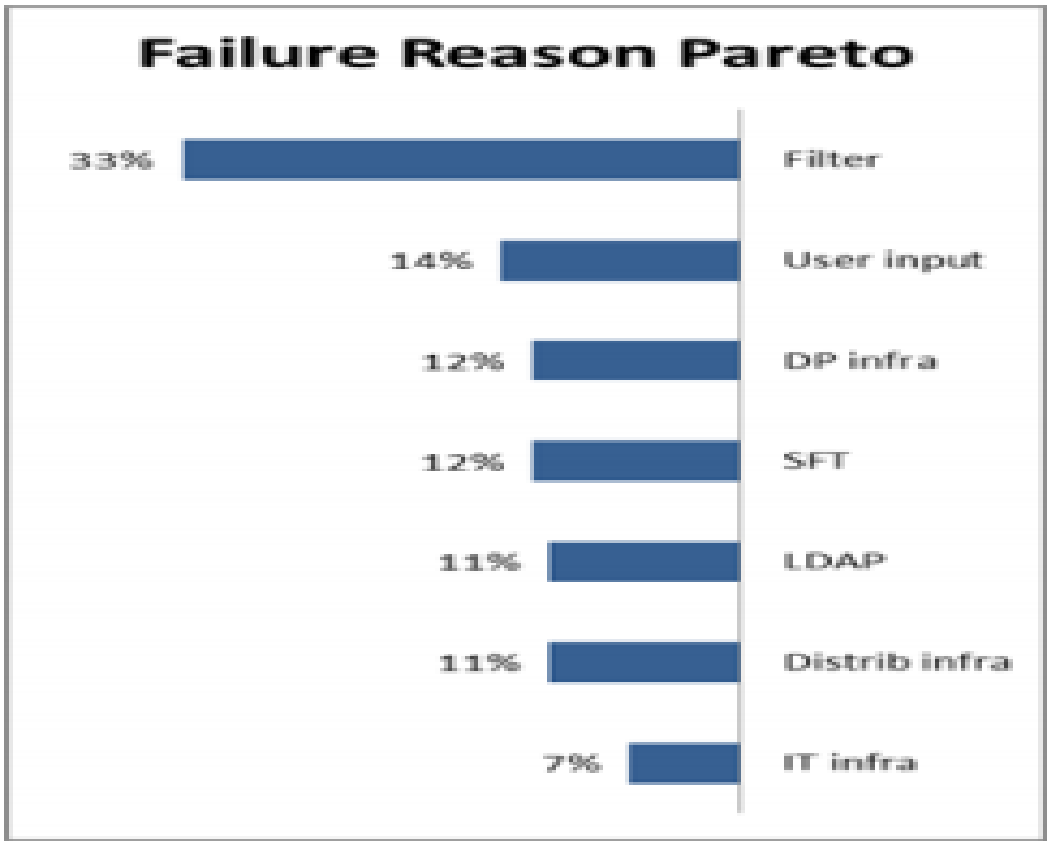

Figure 2.4: Failure Pareto

Referring to 2.4 shows reasons of obstruction in product delivery due to which a DRT requests get failed. It shows after optimizations 95% DRTs delivered successfully and only 5% of DRTs were failed.This failure percentage distribution is of those failed 5%. Some of the factors are not maintained by data management team rather outside of distribution system.

In Distribution System, due to importance of this tool scope of improvement is always possible. Hence, in this action of enhancement one new module Reporting feature was also proposed to develop to generate report of DRTs for analysis purpose.

In UolBE, complex and unstructured contents of SFT and license mails prompted to restructure the mail, change look and feel using HTML, CSS and processing the contents. Apart from that, a java mail API which was developed to send mail was also enhanced to provide multiple attachments, adding contents of mail through string or read it from

a file and option based command line arguments.
In SFT request, a incremental delivery concept was introduced in which it is decided to send a complete product or just the difference or incremental part of it. before sending it needs to be set a threshold to decide whether to send this delivery completely or incrementally.

2.3 Glassfish Server: Application Deployment

An Assembly can also be called package is prepared at development end containing various components of an application or module before deploying it on [4]Glassfish Server. Deployment is the process to install module or an application. Many of the Java EE applications can be deployed in different ways-

1. Archive Deployment- deploy as an archive.
2. Dynamic reloading- Redeploy after modifying application.
3. Automatic Deployment- deploys application archive in auto deployment directory.

A deployment descriptor decides how an EE module or application will be deployed on server as each xml file has a corresponding DTD or XSD.
To distinguish between modules and application, module is a collection of one or more Java EE components which runs in a same container type. There can be various types of modules like Web module, EJB module and others. Whereas, on the other hand and application is a logical collection of one or more modules.
Module based deployment allows shared access to a bean from web, EJB.
Application based deployment comes into picture when components are supposed to work as a unit.

2.4 LDAP

Lightweight Directory Access Protocol is a process adopted by organization for user authentication. It is also being used in every application or web interface of internal users.

Chapter 3

Technological Environment

Distribution team follows a standard for software development. Any application goes through three phases:

1. Dev environment
2. QA environment
3. Prod environment

In dev environment, application is developed and tried and realized on local machine with all installations required. Thereafter goes through QA environment to check if functionality is achieved in dev are working proper. QA and Prod environments are similar environment setup. For quality assurance purpose developers setup environment similar to Prod. Once, confirmed in QA then application is deployment on Prod to make it available to its users.

Operating System: Windows 7

Application Type: Developed application is ReST full application. No specific framework has been used. This type of application use client-server, stateless, cacheable communication protocol. ReST is an architectural style for designing web applications. Other type of web application is SOAP which provides a way to communicate between different applications running different operating systems, with different programming languages. Distribution Dashboard application is a ReST full application because these are lightweight application.

Programming Language: Java 1.7 and JRE7, HTML, Java Script, JSP, [3]Perl Script.

Database and its Design Tool: DRTDB and UPTDB; SQL Workbench 5.4

IDE: [5]Eclipse (J2EE environment)

Local host server: Apache tomcat 7.0

Application Deployment Server: [4]Glassfish Server 3.0

Chapter 4

Distribution System optimization: Distribution Dashboard

4.1 Implementation Details

Distribution System has three components UolBEplus, DRT and Approval System. This chapter is dedicated to discuss the work done in Distribution Dashboard which is sub component of Approval System. During this project, three releases were planned and deployed successfully for various enhancements and few bugs identified in the system at customer's end.

Previously the Distribution Dashboard was having few bugs, unoptimized database queries as few of the database connections or statements were not closed and exceptions were not handled properly. Further after close review it was seen that redundant code was written which could increase the efforts and cost of maintenance.

Underwritten are some of the majorly present flaws in the system which could increase the cost and efforts required for the application.

4.1.1 Analysis:

Flaws in legacy system-

1. Redundant jsp files.
2. MVC was not properly maintained.
3. Lack of code reusability and presence of hard coding.

4. Session variables were maintained using Http Session request object.
5. Maintenance of two source codes for QA and Production.
6. Separate application level login along with server login.
7. Separate code maintained for development and quality assurance hence required multiple merging efforts.

Design and Flow Diagram

Distribution Dashboard is a dynamic web development project. It is a ReST full application. This subsystem mainly contains two parts, one for admin and other for user.

Admin Dashboard: This interface keeps track of DRT states such as Submitted, failed, approved etc., one table of DRT details for each state. Other features are like highlighting the row of DRT details which are to be processed manually. Maintaining logs for failed DRTs. Comparing 4.1 is legacy flow diagram, 4.3 optimized flow diagram, in which latter overcomes lot of complexities. Previously, it had a application login page and due to security purpose server authentication was also asked by the user hence multiple login was making application less user-friendly. Hence it was decided to drop application level login and avoid server level login and introduce a central authentication login throughout all the applications for a session. If any application is already opened then other applications will not be asked for login credentials. It was very good implementation for both developer and users. Hence, it avoids development of LDAP check to be performed at application level.
Next step to access in optimized process is inclusion of a java class which takes care of fetching details of DRTs in each states and return it back to the jsp page to display. Also a ParameterReader java class was also introduced to read all the configuration parameters from a property file. this file contains database server credentials hence ParameterReader class handles it. After reading details the admin dashboard page gets opened. This process incorporates MVC model and use of property file which makes it more usable and stable.

User dashboard: This displays the details of requests which were recent and pending for approval at user end. Comparing 4.2 is legacy flow diagram, 4.4 optimized flow

diagram. To display details of approval and recent requests three pages were developed which could be possible in two pages only. To display details of approvals one page was maintained for each user using different sessions. It had three entry points. One of them was through the approval mail which comes from the DRT once approval system notifies to it that a new approval mail has arrived in system. The other page gets opened if user directly tries to access these details and third would open if user has taken decision on DRT either to approve or reject.

In new process only two entry points are there one from directly opening and other one if try to login from approval mail url. It does the same CAS login as in admin page.

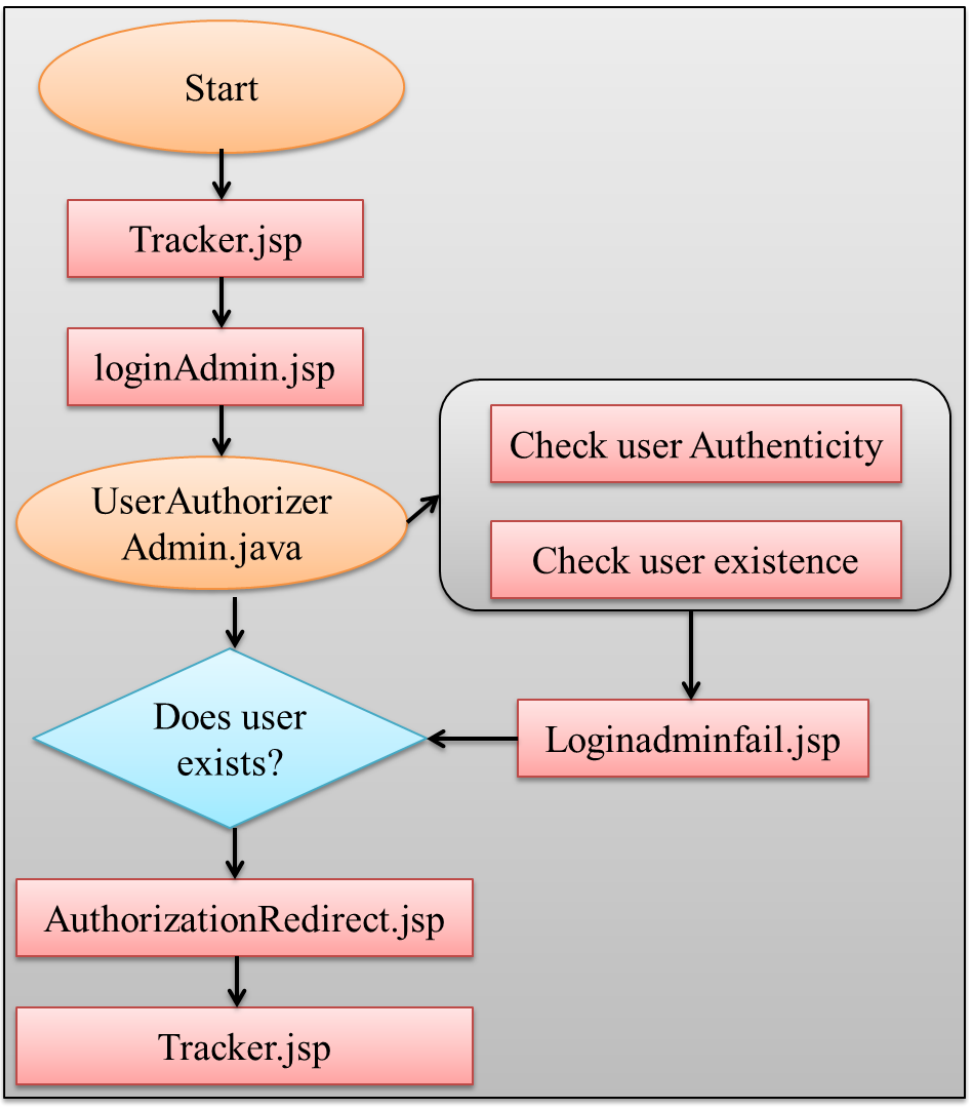

Figure 4.1: Legacy Admin Dashboard flow diagram

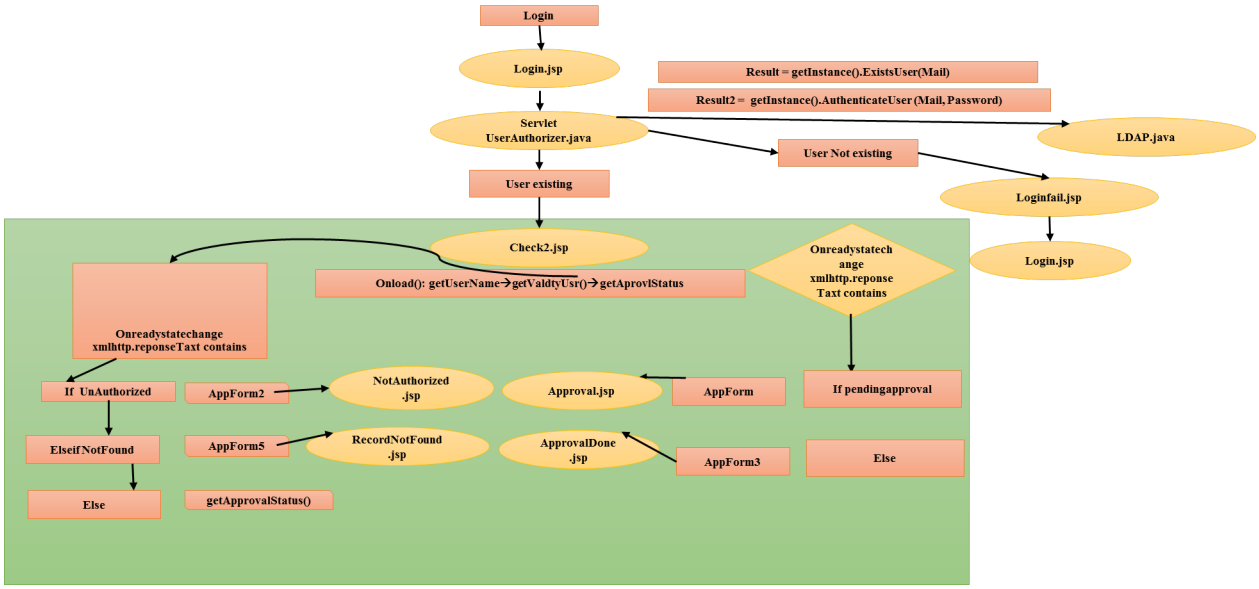

Figure 4.2: Legacy user Dashboard flow diagram

4.1.2 Design Level Optimizations

Once analysis was done for legacy system it was decided to changes at design, implementation as well as deployment level.

1. Application level login was dropped
2. CAS login was integrated
3. Merging of jsp files
4. Various intermediate jsp files were discarded
5. Http session request object was dropped to maintain session variables
6. Removal of useless servlets

Flow diagram of optimized system is as below-

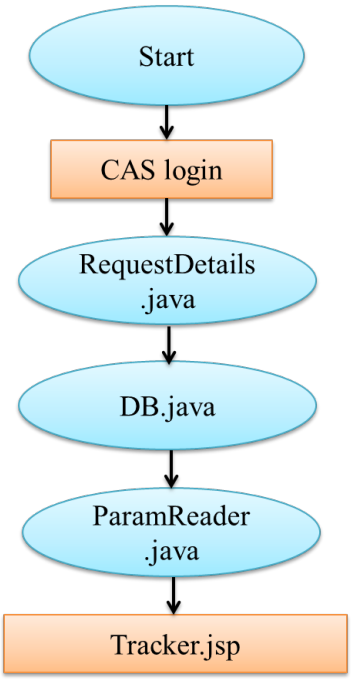

Figure 4.3: Optimized Admin Dashboard flow diagram

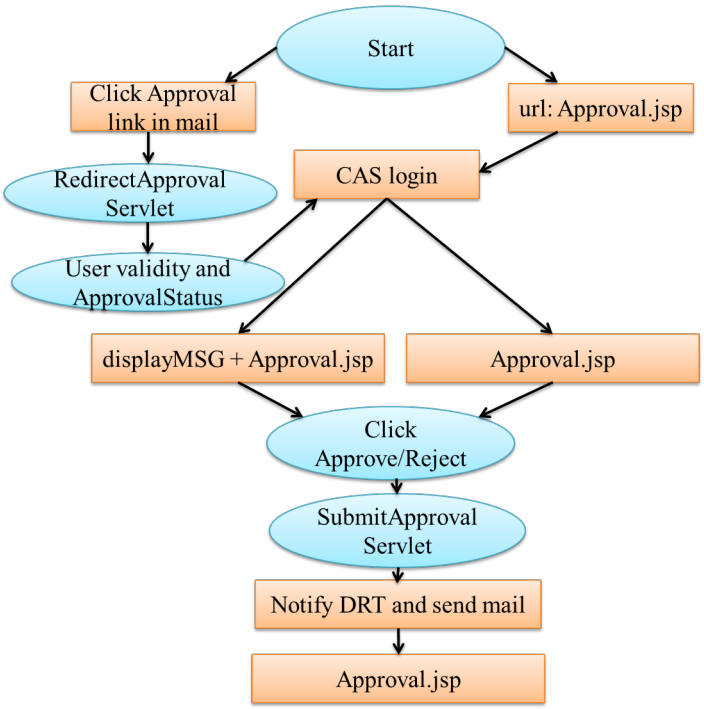

Figure 4.4: Optimized user Dashboard flow diagram

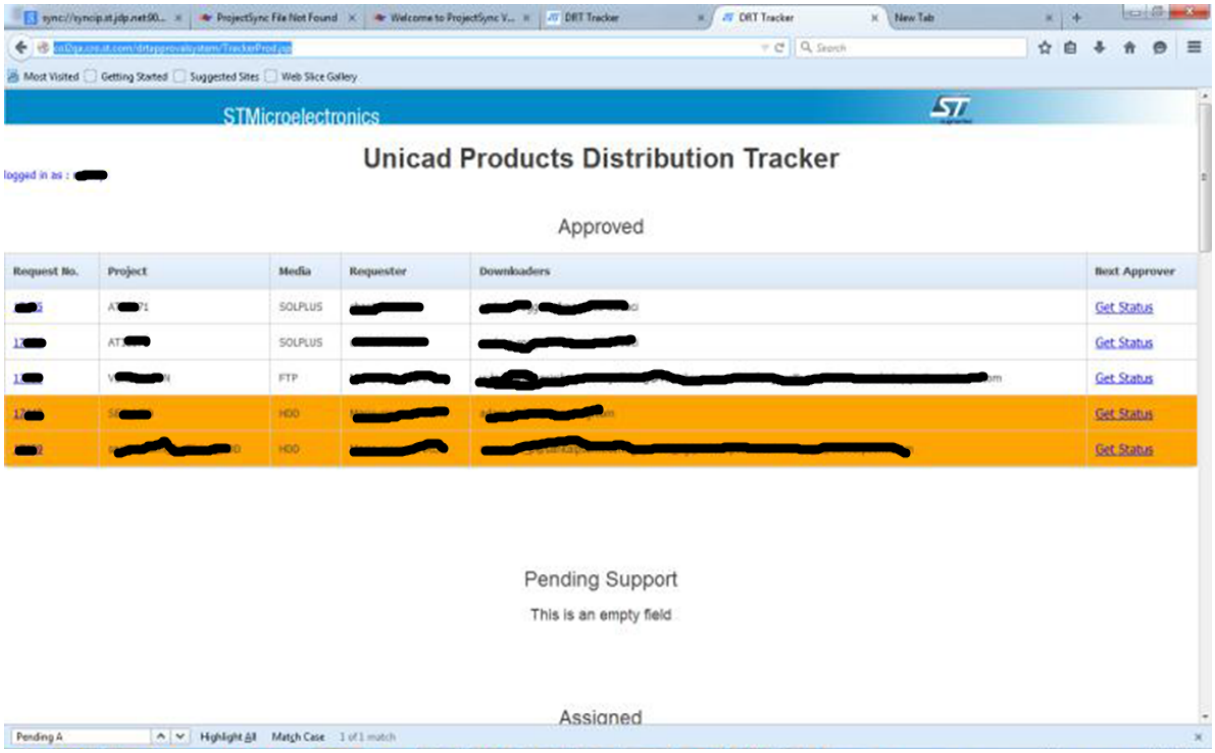

Figure 4.5: Optimized admin Dashboard web interface

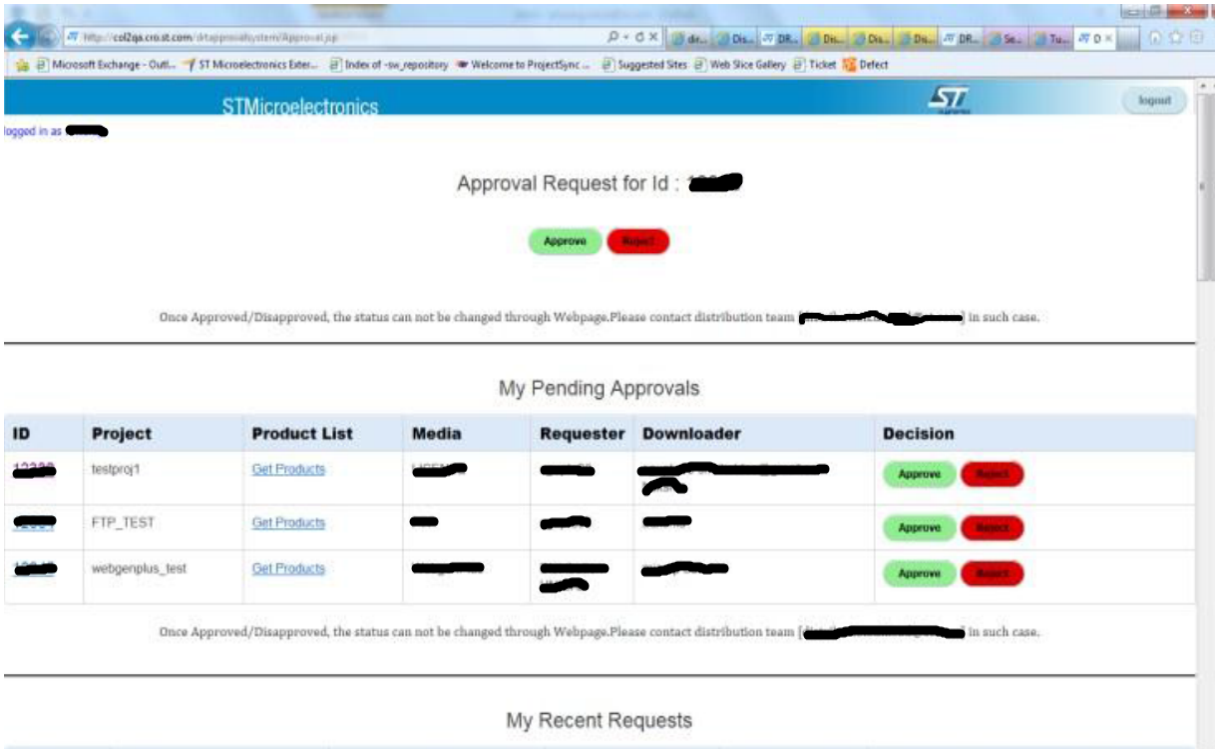

Figure 4.6: Optimized user Dashboard web interface

4.1.3 Feature Enhancements : Release v1.7.3

- **Renaming a jsp source file (release v1.7.3)** Renaming a jsp file does not affect deployment and deployment descriptor file.
- **Highlight all manual deliveries on admin interface (release v1.7.3)** All requests which are to be processed manually should be highlighted to distinguish from those are to be taken care by automation.
- **Select Action as ProjectName and Media if absent release v1.7.3** For few actions in DRT, delivery Medias and project name not defined hence Action is used to display for these details.
- **GetProduct link pop-up with product details populated (release v1.7.3)** Populate a pop-up window displaying products corresponding to request.
- **"GetDetails" about approvers DR as well as PL (release v1.7.3)** Populate a pop-up window displaying DR/PL details corresponding to request.
- **Downloaders name column should not be empty (release v1.7.3)** For few actions in DRT, downloaders not defined.
- **Configure database connectivity (release v1.7.3)** Remove all hard link codes for database access replacing with local variables. These variables are initialized with parameters of db connection reading from a property file.

4.1.4 Feature Enhancements : Release v1.7.4

- **Build script for QA and Prod (release v1.7.4)** Use Ant script for QA and Prod code separately to create application war files.
- **IE11 document compatibility mode issue (release v1.7.4)** Previously AJAX was being used for user validity purpose and http responseText. It was not compatible with IE11 as all systems were upgraded to use IE11 inside organization. Hence, problem was severe and immediate actions were taken and use of AJAX was dropped and as an alternative all this handling was done through a single servlet.

- **SSO Integration (release v1.7.4)** To avoid multiple logins (application level and server authentication level) Distribution Dashboard was designed to use SSO service which checks authenticity of user from LDAP and it works for all applications for current session.

- **GetStatus of DRT from DRTDB but not from UPTDB (release v1.7.4)** To avoid one more extra query that to from separate database only single query was used to display SRT status.

- **Use displayMessage instead of separate jsp page (release v1.7.4)** In legacy for separate errors or notifications multiple jsp files were being maintained. From optimization perspective it was decided to display message on user dashboard itself. This functionality was achieved by setting and getting session attributes.

4.1.5 Feature Development : Release v1.7.5

Reporting Feature module Reporting Feature was aimed to develop to facilitate its users to fetch details of DRTs and get an analysis report in the form of csv file. Day to day users of system do the DRT to deliver products to the customers. To have a better understanding refer to 4.7 each DRT contains certain details such as owner of request, date of approval, list of products, type of media to send delivery to its customer etc. If a user wants to see the details of these in future then she will have to go through each and every DRT.

Hence, in the development of this module a search page was designed which has a form in which she can give some parameters like project, from date and till date, media, action and others. On clicking search button a these details are parsed and sent to a [6]Java API which generates a dynamic [2]SQL query having search criteria and joins.

Once a call is made to search form to fetch drt details a form gets opened which allows user to insert entries in various fields. Search criteria are project name, requester name, product name all these fields have keyword search facility. Other fields like media, action all have a list which gets populated while calling a function which reads these entries from an enum class hence it is dynamic list. Date picker of jquery has been implemented to the get from and to date getting the calendar. Along with this handling of dates like

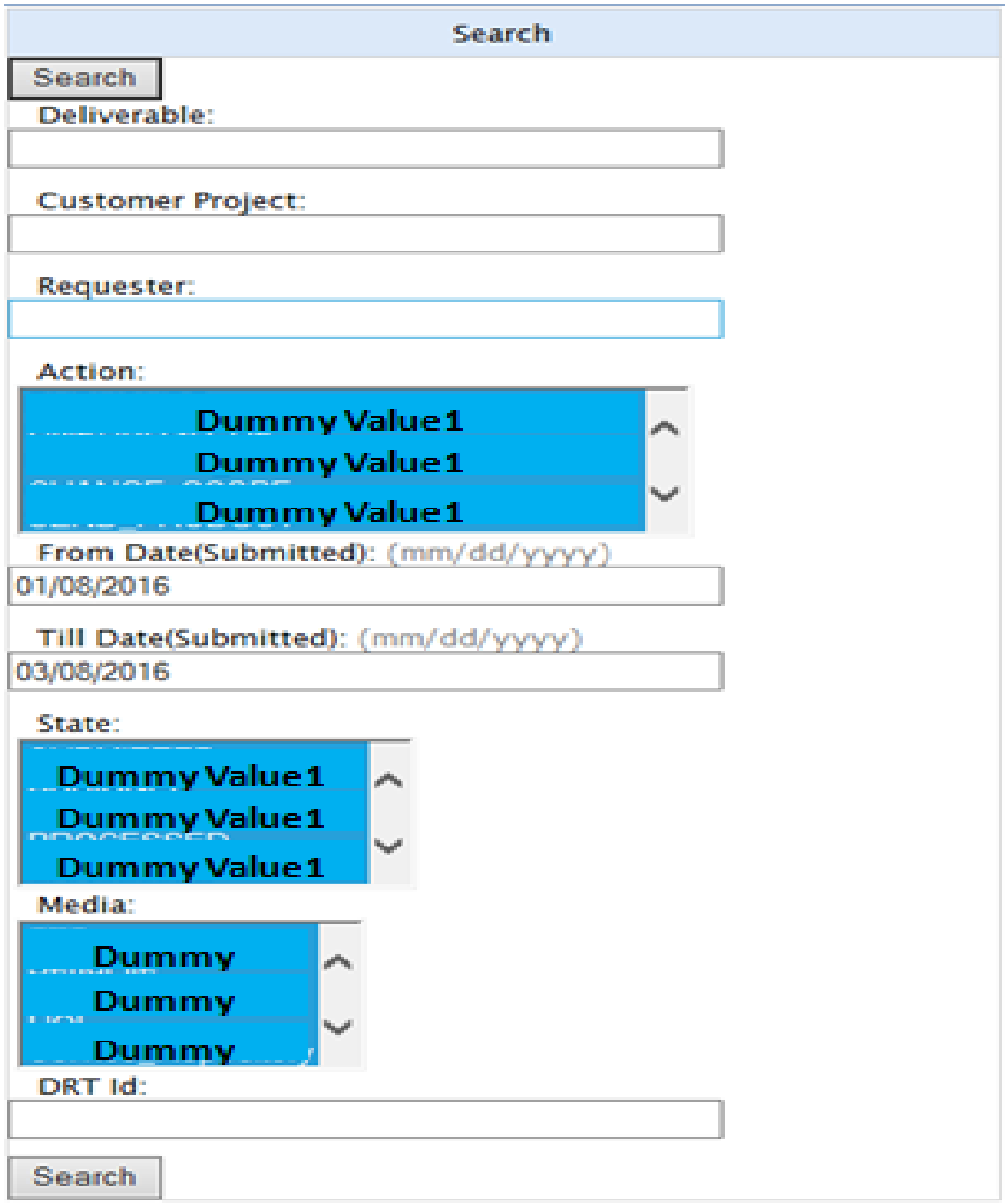

Figure 4.7: Form of Reporting feature

from the date should not be later than end date and validation of whether day of month exists or not. If date is not valid then it display a message box with appropriate warning.

Populated result on a page can be seen in 4.8 where rows of table are filled one row per deliverable. Dynamic query which is created at run time using a java API fetches result and sends back for rendering on result page.

In this module we have two data structure signatures, firstly input is a HashMap<String,

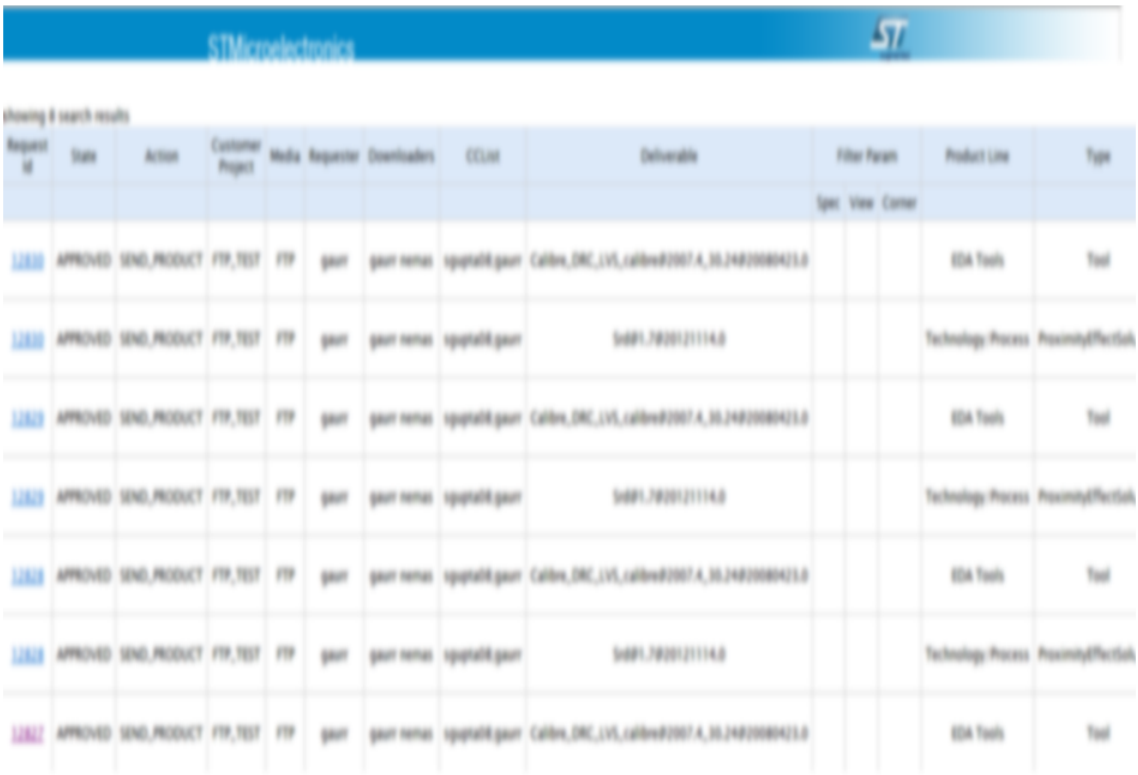

Figure 4.8: Result based on search parameters

ArrayList<Object>>and secondly output signature is ArrayList<Hashmap <String, Object >>.

On the bottom of the result page there is a button provided which on click lets you download the contents as csv file to your local storage. The csv export mechanism has been implemented using content disposition method which parses data from session variable and gets it data from it and save as csv also if user has downloaded first time and button gets disabled until you again send the query to the server by pressing refresh button.

4.2 Test Cases

Test cases were prepared for all releases and thorough Unit testing was done. Latest release v1.7.5 is in production, deployed on glassfish server.

Chapter 5

Distribution System optimization: UolBEplus

5.1 Implementation Details

UolBEplus is another feature of distribution system in Data Management team. As explained earlier this comes into the scenario once a product delivery request comes into the system and xml files are generated for each of the product listed in the request. Hence, from the start of processing those [1]xml files till the delivery is done to customer it plays a big role all around.

This chapter mainly focuses upon work done in the direction of improving overall mailing system form the format of mail till internal processing of sending e-mail to customers.

To provide a better look and feel after taking feedback from customers mail template was improved by including HTML and CSS which was plain text earlier.

5.1.1 Mail templates Enhancement

Mail contains certain instructions to install products at their end, thanking note and delivery sender's note can be seen in fig 5.1 and fig 5.2. Hence, rendering in plain text on a simple background these instructions were not very user-friendly. hence, decision was taken to incorporate HTML and having sections with different colours for each type of information like sender's note and install instructions.

During its development two main problems were faced-

1. To find out how to implement HTML in mail template and validate whether it is possible or not? It was found that there is an attribute HTMLBodyText=0 in upload.tsk file which means enabling it will work for any type of HTML content. Later on contents of mail were written in HTML format and used to send the mail.

2. Parse that HTML formatted mail template file as user comment section could be displayed(or hide) when sender's note is present(or absent) in DRT. To achieve this a tag was included that would be used to delete those lines in HTMl which are dedicated to render sender's note while parsing in Perl.

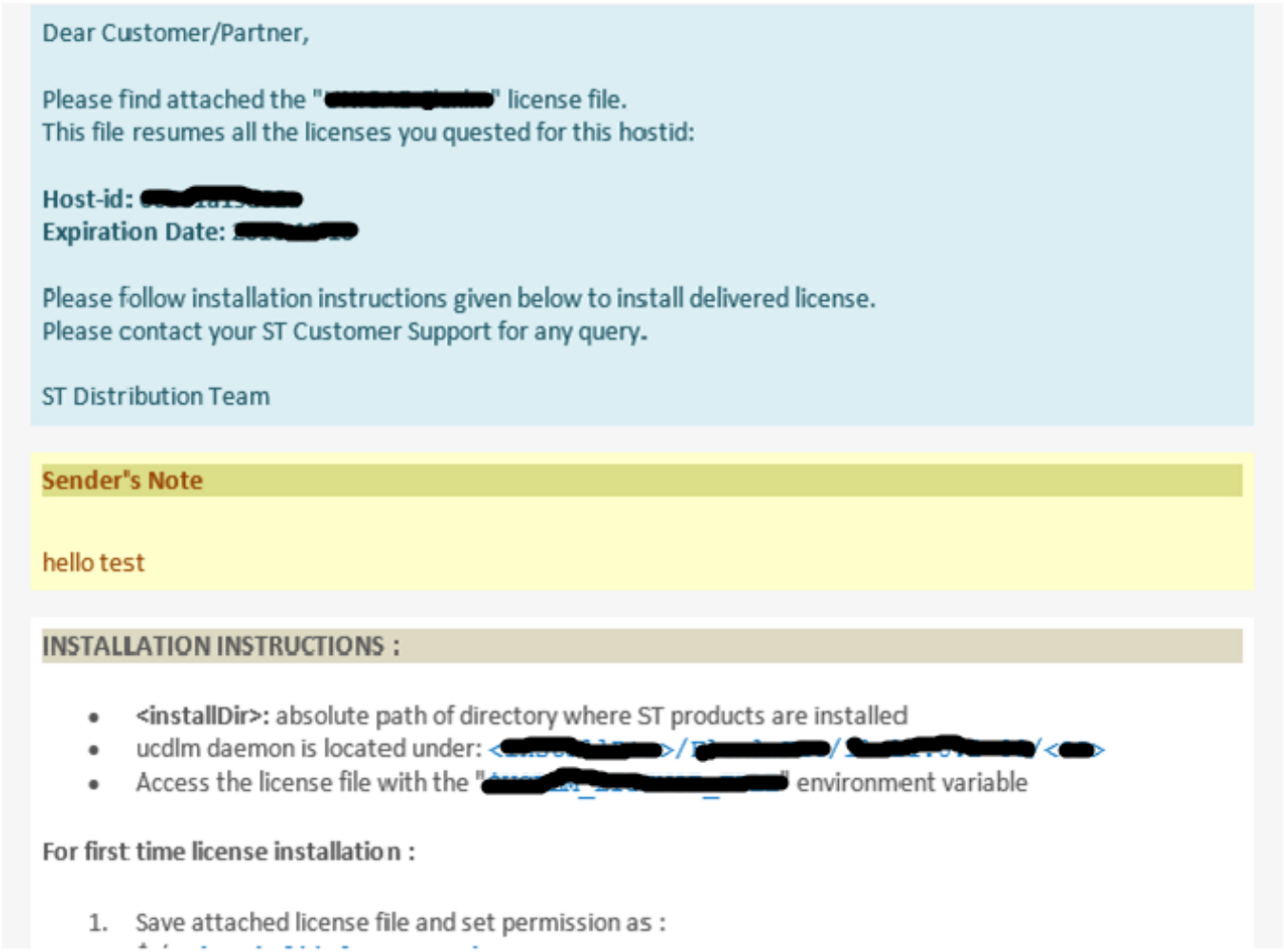

Figure 5.1: Form of Reporting feature

Here, in both fig 5.1 and fig 5.2 it can be seen that a mail contains three section. First section is delivery note, second is sender's note and last is installation instructions

hence to achieve this level of clarity it was taken on priority. Also, customer was asking to attach a separate pdf or an instruction's file which could be used to product installation.

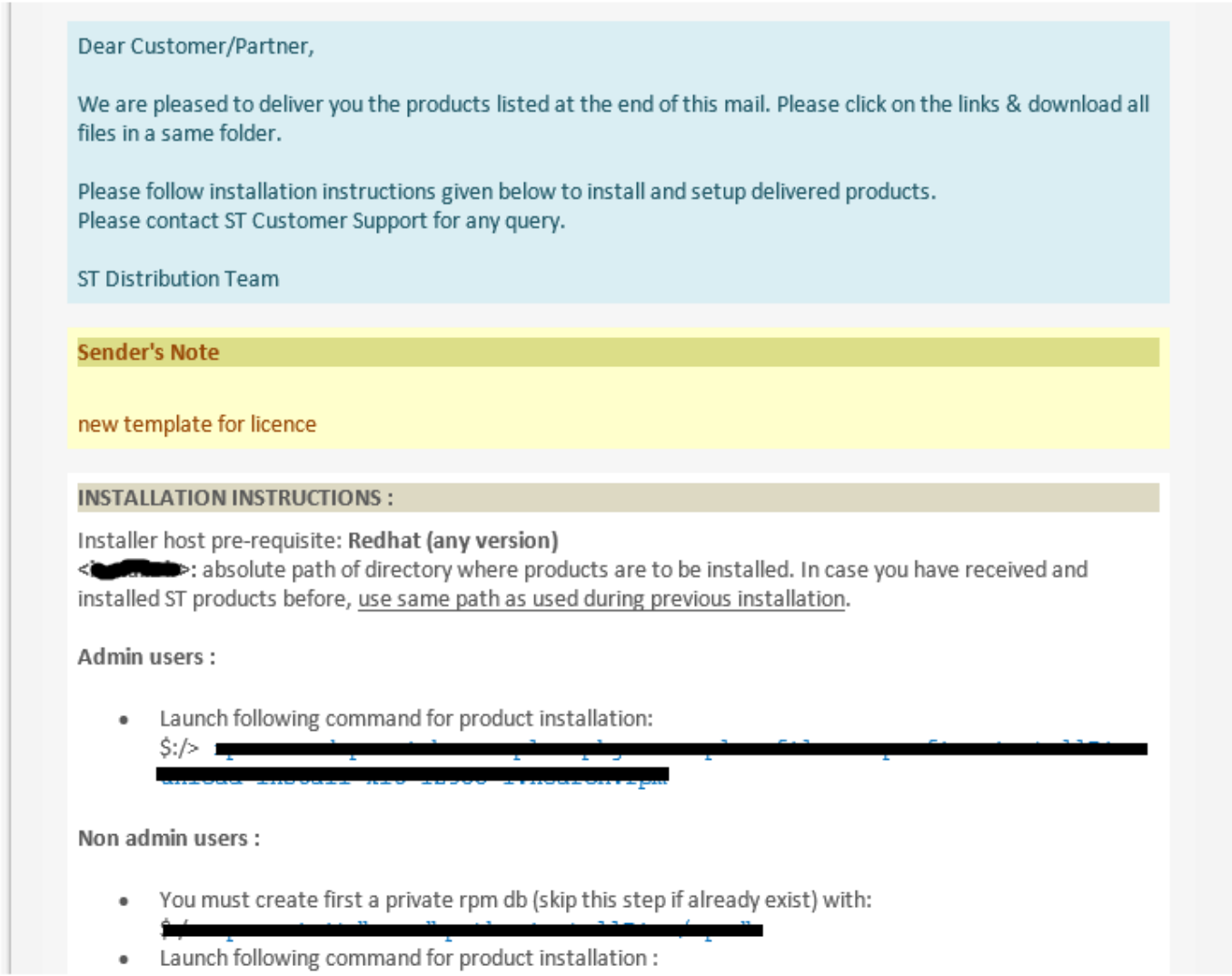

Figure 5.2: Result based on search parameters

5.1.2 JAVA Mail API

In continuation of mailing system improvement process to send delivery mails a java mail API was developed in UolBEplus (written in Perl) which the only java class exists. In [7]shell we have send mail API which can't have mail attachment and mutt command option which could not send mail to outside customers.

This mail API is called from a Perl module (one for each type of delivery) in which there is a mail to be sent. To call this java class it requires few arguments to process the call.
Below are discussed features covered in development of java class to send delivery mails-

- **Option based command-line arguments** Option based command-line is always gives better understanding to its user because option name gives an idea that what this option is for? if user has to give certain number of values in command-line then she may not be aware of what a particular value is being used for also the order of input arguments may have to be maintained.

- **Email Validation** Previously no email validation was done. While parsing email, regular expression has been used to validate it.

- **exitStatus() from Java to Unix shell** This feature was very crucial. if Java class throws an error or Perl will be notified how and why the execution failed. Hence, exitStatus() is a function which sets a variable and returns a status to Unix shell through which helps in easy debugging the real cause of failure.

- **-help Option** This option was implemented for user to get herself acquainted with the command-line options can be given at the time of execution. Here user has the flexibility to user ("-h", "–h", "-help" or "–help") to get the help.

- **Multiple occurrence of any Option** In command-line user can give one option any number of times and API will take care of all of them. Apart from that if user wants to give multiple values corresponding to an option then she can give multiple values in single option using ';' as delimiter.

- **Multiple File Attachments** Previously there was no handling or facility to send multiple files in mail as attachments. But with this development it allows you to do so. Also, user do not need to check if there is any attachment or not to be sent with

this mail which was being done earlier based on passing an argument "Attachment" or "No Attachment".

- **Read mail content from File or String** To compose mail content it has two arguments "-mailContent" and "-mailContentFile". At the time of sending delivery all those contents be it as string or from a file are collected in one string sent by rendering it to content section of mail.

- **User defined Function for throwing Appropriate Error** Instead of relying on Java to handle errors, a procedure was developed to display the error like "SUBJECT_EMPTY_WARNING", "INVALID_FILEPATH_ERROR" or "MULTIPLE_SENDER_ERROR".

5.1.3 Incorporate UPT+ Size API in SFT Incremental delivery

Incremental deliveries can be sent only if the base or previous version of the product has already been sent. Hence, various checks apart from this like if the product is a UPT+ product? and others. One more check is to be added to decide if the delivery will be incremental or not. For that we need to set two thresholds -

- **Go for incremental only if the product is of size bigger than a threshold.**
- **If the new version has below than significant difference so that to send incremental.**

Product size and difference of product and base product is calculated by using UPT query to get the size of product. On running a query product size and diff size is resulted. Based on these values we decide whether to send the particular product incrementally or not.

Chapter 6

Discussion and Conclusion

Distribution team has achieved 80-90 percentage automation in products delivery. Drastic change has been observed from after overcoming filter error and user input error two main reasons for failure DRT and automation breakdown. Along with ease of use for its users optimizations are achieved to maintain more secure and robust system. Ease of getting installation, transparency in process. Fig 6.1 reveals below success story of distribution system automation.

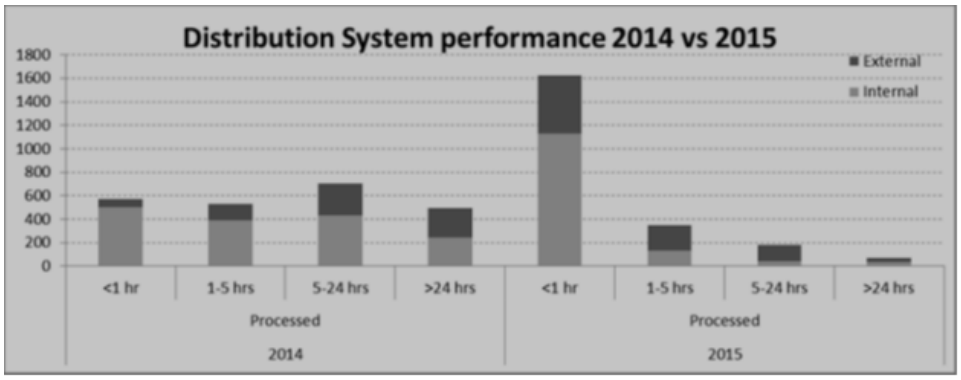

Figure 6.1: Distribution System success timeline

This work has a major contribution towards adding value to the service provided to customers. Distribution Dashboard is very much crucial component as user directly interacts with it to approve or reject request hence it becomes that it should be robust. On one side implementing CAS is an ease to user and development of Ant Script and code re-usability reduces maintenance cost.

reporting Feature feature was one of the long awaited requirement from the user hence

developed on priority. This reduces manually fetching the DRT details.

Improvement in mailing API and look and feel in templates resulted in customer satisfaction and reduced the support.

Finally, incorporating UPT+ size API has reduced the chances of unnecessary sending incremental delivery by adding two checks based upon product sizes.

Chapter 7

Planned Optimizations in Approval System for Q3-2016

7.1 DRT Approval Admin tool migration

This tool is used to maintain the list of approver to approve or reject any request comes in system. Only PMs can modify this detail in database. This tool has been planned to migrate on jsp servlets so to remove dependency of maintaining the VBA code.

7.2 FAQ for DRT

User while doing DRT through a web interface may face some issues because of less understanding of the system. Hence, it has been planned to prepare a FAQ dedicated to DRT related queries. This application has been planned but whether we will use jsp web development or through Sharepoint we will try to achieve it.

Bibliography

[1] "Xml parser in perl."
http://www.perl.com/pub/2000/10/begperl1.html.

[2] *MySQL WorkBeanch Manual, STMicroElectronics.*

[3] "Perl introduction documentation."
http://www.perl.com/pub/2000/10/begperl1.html.

[4] *Working with Glass Fish Server, STMicroElectronics.*

[5] *Working with Eclipse, STMicroElectronics.*

[6] *Coding guidelines in JAVA programming, STMicroElectronics.*

[7] "Basic of shell scripting."
http://linuxcommand.org/writingshellscripts.php.