Design and Implementation of an Automated Tool for Issue Tracking and Report Generation

Prepared By

Kishan Varshney

12MCEI33

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2014

Design and Implementation of an Automated Tool for Issue Tracking and Report Generation

Major Project

Submitted in the partial fulfillment of the requirements

for the degree of

Master of Technology in Information and Network Security

Prepared By

Kishan Varshney

12MCEI33

Guided By

Dr. Madhuri Bhavsar

and

Malavika JM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

AHMEDABAD-382481

May 2014

Certificate

This is to certify that the major project titled **"Design and Implementation of an Automated Tool for Issue Tracking and Report Generation"** submitted by Kishan R. Varshney (Roll No : 12MCEI33) towards the partial fulfillment of the requirements for the degree of Master of Technology in Information and Network Security 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 the examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Madhuri Bhavar
Internal Guide & Professor,
Department of C.S.E.,
Institute of Technology,
Nirma University, Ahmedabad.

Malavika J.M
External Guide & Graphics Software Engineer,
PC Client Group (PCCG),
Intel Technologies, Bangalore

Dr. Sanjay Garg
Professor and Head,
Department of C.S.E,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Sharada Valiveti
Associate Prof. & PG INS-Coordinator,
Department of C.S.E,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K. Kotecha,
Director, Institute of Technology, Nirma University, Ahmedabad

Undertaking for Originality of the Work

I, **Kishan Varshney (12MCEI33)**, give undertaking that the Major Project titled **"Design and Implementation of an Automated Tool for Issue Tracking and Report Generation"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Information and Network Security** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism 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.

Kishan Varshney (12MCEI33)

Date:

Place:

Endorsed by

Dr. Madhuri Bhavsar — Internal Guide

Malavika JM — External Guide

Acknowledgements

It gives me great pleasure in expressing thanks and profound gratitude to my project mentor **Malavika J.M**, Team Lead, Intel Technology India Pvt. Ltd, Bangalore for her valuable guidance and continual encouragement throughout the Major project. I heartily thank her for her time to time suggestions and the clarity of the concepts of the topic that helped me a lot during the learning and ramping up period in the industry.

I would like to extend my gratitude to my internal guide **Dr. Madhuri Bhavsar**, Professor, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for fruitful discussions and valuable suggestions during meetings and for their encouragement and motivation.

I would like to extend my gratitude to **Prof. Sharada Valiveti**, Professor & PG INS Coordinator, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for their encouragement and motivation.

I would like to thank **Dr.Ketan Kotecha**, Honorable Director,Institute of Technology, Nirma University, Ahmedabad and **Dr. Sanjay Garg**,Professor and Head of Computer Science and Engineering Department for providing me the opportunity to go and pursue my internship at Intel Technology India Pvt. Ltd.

I would like to thank everybody at Intel who helped me and made me feel as a part of the team right from the very first day. This includes my manager **Bhavesh Shroff**, my mentor **Malavika J.M** and all my teammates.

- Kishan Varshney

12MCEI33

Abstract

At Intel, Platform Application Engieering Team works on Intel Graphics Driver Validation. Every Employee needs to track all the Graphics Issues of their respective customers on all the platforms like Ivy Bridge, Sandy Bridge, Haswell and also on up-coming platforms like Broadwell, Sky Lake, etc. Hence, Once an graphics issue is logged in internal database and the severity of the issue is determined, relative priority has to be assigned, followed by tracking the issue across various engineering teams to identify the root cause and solution. The developed web based tool provides a fast, user friendly and one stop point interface to employees to gauge the status of the customer issues, to follow up on next steps, and generate indicators and reports. This web based tool improves the productivity of employees by further automating customer support related activities. Due to very high confidentiality of the Database (HSD), All the security concerns are taken into consideration by creating the HSD API to securely login into the database and only allow authorised person within Intel to access the tool. Also, Relevant tables are copied into local database and normalised accordingly so as to increase the speed to access the database to increase the overall efficiency of the Tool.

Contents

List of Figures

Chapter 1

Introduction

When any customer files graphics related issues on Intel Platform, engineers in the team need to track those issues that get logged in internal database till closure. This involves various activities such as replicating the issue, determining relative priority of the issue based on severity, following up with engineering teams to identify the root cause of the issue and then finally the solution.[1]

With multiple issues open, across multiple designs it often becomes time consuming for the employees to track various issues, their associated need dates and various stages of progress.

As an example, assume OEM 1 submits 15 issues over a period of 1 week across 4 designs which have different need date for solution. These 15 issues are then logged into internal database. They need to be simultaneously tracked for replication, debug, fix and release as per the process. During the course of tracking, issue 1 of design 1 would need root cause analysis while issue 3 of design 4 needs to be prioritized and issue 4 of design 2 needs priority bumped up. Each issue depending on the current state elicits a different follow up activity. Currently these activities are being done independently.

The tool provides fast and user friendly interface that captures issue status in a snap shot, prompts actions depending on current status, automates request and helps generate reports and indicators.

In a nut shell, the tool is a one stop point for issue tracking and follow up the activities.

1.1 Objective

To design and develop a customizable Web based Tool as a One Stop Point to consolidate and automate multiple activities involved in Customer issue tracking and support.

1.2 Area of Work

The group consists of Application Engineers (AEs), who replicate, debug and diagnose the issues that OEMs report and follow up with engineering teams to identify, debug and fix the issue.

1.3 Present Scenario

One of the primary responsibilities of an AE includes filing OEM bugs in internal database and tracking them through the entire lifecyle of the bug till closure.[1]

AE typically builds SQL queries from Query Builder to pull out details needed to track various issues and progress made on each. For activities like issue prioritization, status update report, etc., AE pulls out the relevant information from the database to fill in request forms. This is time consuming. This tool automates the bug support activities which will reduce the manual work involved in pulling out relevant data for an issue.

1.4 Organization of this Report

This section gives brief information of all the chapters. The first chapter, introduces to the reader an overview of the project. The second chapter gives the background theory and literature review that was required to develop the tool. It details about concept such as Issue Tracking Database, the Issue Tracking Database API's, the HSD API, Active Directory and LDAP Server. The third chapter details the methodology adopted in the development of the tool that were built in this project. It also shows how each of the components detailed in chapter two are used in the development. It shows working of the tool, and what tool were used to implement their functionalities. The fourth chapter describes the results that were obtained during the project. The fifth Chapter contains the detailed analysis of the results. The last chapter explains the conclusions and future work that were drawn as a result of completion of this project.

Chapter 2

Literature Survey

2.1 Introduction to HSD

HSD is an abbreviation of High Speed Database that is meant for tracking the bugs. There is an UI available to display the issue details.[**?**]

When a bug is filed, it's in a "New" state. As there is progress in the debugging and as details are obtained about the bug, the HSD data is updated. Also, Bug moves across various stages such as New, Debug, Development, Resolved and Closed.[**?**]

On the background, the HSD is a query-able database, which can be queried for data using SQL queries. A front-end is built around this database, using which users can create bug queries, view and update individual bug details and do tasks such as exporting lists of bugs to an Excel workbook.

2.2 Introduction to FreeTDS

FreeTDS is re-implementation of C libraries originally marketed by Sybase and Microsoft SQL Server. It allows many open source applications such as Perl and PHP (or your own C or C++ program) to connect to Sybase or Microsoft SQL Server.[4]

FreeTDS provides drop-in replacements for: [4]

- Sybase's DB-Library and CT-Library
- Microsoft's DB-Library (which differs in small details from Sybase's)
- The ODBC drivers from both vendors
- Interactive SQL and BCP utilities

FreeTDS is a source code library, not a program in and of itself. Users generally compile the library from source and link another program to the library to allow the other program to use the FreeTDS API. However, recent releases of FreeTDS do include some client programs, such as fisql (a replacement for the isql utility programs distributed by Sybase and Microsoft). FreeTDS is licensed under terms of the GNU Lesser General Public License.[4]

2.3 Introduction to HSD API

The HSD API is an internal API for developing application programs that need to interact with databases. The Issue Tracking Database API allows a .Net/ C developer to interact with the database and simplifies certain tasks such as login, querying, saving queries under a user's profile and other basic tasks. These allow a programmer to accomplish tasks without knowing about finer details about the HSD working.[3]

HSD API is the Application Programming Interface for accessing the HSD database, in a sense, the automation engine for HSD. Using the HSD API with one of the programming languages mentioned below, one may create programs which may query data from the usual HSD databases.[3]

Note that there is no support in the current HSD API version for accessing the HSD Data Warehouse.

HSD API common use cases :

- Reading data from a HSD Focus
  - Use the HDataReader object
- Inserting new records into a HSD Focus
  - Use the HApiObj object
- Updating existing records in a HSD Focus
  - Use the HApiObj object
- Misc
  - Upload/Download attachments
  - Query MetaData
  - Save/Execute shared/private queries

2.4 Introduction to Active Directory

Active Directory (AD) is a directory service implemented by Microsoft for Windows domain networks. It is included in most Windows Server operating systems.

An AD domain controller authenticates and authorizes all users and computers in a Windows domain type networkassigning and enforcing security policies for all computers and installing or updating software. For example, when a user logs into a computer that is part of a Windows domain, Active Directory checks the submitted

password and determines whether the user is a system administrator or normal user.[9]

Active Directory makes use of Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Microsoft's version of Kerberos, and DNS.

Active Directory is a special-purpose database it is not a registry replacement. The directory is designed to handle a large number of read and search operations and a significantly smaller number of changes and updates. Active Directory data is hierarchical, replicated, and extensible. Because it is replicated, you do not want to store dynamic data, such as corporate stock prices or CPU performance. If your data is machine-specific, store the data in the registry. Typical examples of data stored in the directory include printer queue data, user contact data, and network/computer configuration data. The Active Directory database consists of objects and attributes. Objects and attribute definitions are stored in the Active Directory schema.[10]

Active Directory features include:

- Support for the X.500 standard for global directories
- The capability for secure extension of network operations to the Web itemA hierarchical organization that provides a single point of access for system administration (management of user accounts, clients, servers, and applications, for example) to reduce redundancy and errors
- An object-oriented storage organization, which allows easier access to information Support for the Lightweight Directory Access Protocol (LDAP) to enable inter-directory operability
- Designed to be both backward compatible and forward compatible

2.5 Introduction to LDAP Server

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network.[11]

Directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.[11]

LDAP is specified in a series of Internet Engineering Task Force (IETF) Standard Track publications called Request for Comments (RFCs), using the description language ASN. The latest specification is Version 3, published as RFC 4511. For example, here is an LDAP search translated into plain English: "Search in the company email directory for all people located in Boston whose name contains 'Jesse' that have an email address. Please return their full name, email, title, and description."[11]

A common usage of LDAP is to provide a "single sign-on" where one password for a user is shared between many services, such as applying a company login code to web pages (so that staff log in only once to company computers, and then are automatically logged into the company intranet)[11]

LDAP directory service is based on a client-server model. One or more LDAP servers contain the data making up the LDAP directory tree or LDAP backend database. An LDAP client connects to an LDAP server and asks it a question. The server responds with the answer, or with a pointer to where the client can get more information (typically, another LDAP server).[11] No matter what LDAP server a client connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server. This is an

important feature of a global directory service, like LDAP.

LDAP is not just for user validation, any task that has the following properties might be a good use case for LDAP:

- You need to locate ONE piece of data many times and you want it fast
- You dont care about the logic and relations between different data
- You dont update, add, or delete the data very often
- The size of each data entry is small
- You dont mind having all these small pieces of data at a centralized place

2.6 Sql Connections and Data Tables in .NET

Connecting to an SQL Database is a simplified task in .NET. A SqlConnection object represents a unique session to a SQL Server data source. With a client/server database system, it is equivalent to a network connection to the server. SqlConnection is used together with SqlDataAdapter and SqlCommand to increase performance when connecting to a Microsoft SQL Server database. For non-Microsoft SQL Server products, and other OLE DB-supported data sources, OleDbConnection can be used.

Once a connection is established with the database server, for which we need to specify credentials (SQL user name and password) and the initial catalogue, i.e. the database object that the connection should work with, we can perform operations on the database using the SqlDataAdapter and SqlCommand object.

Chapter 3

Implementation Methodology

3.1 Introduction

This chapter intends to discuss tool developed in this suite, their architecture and algorithms involved in building them. The tool kits/ APIs used are also discussed along with block diagrams representing overall functioning of the tool.

3.2 Methodology

3.2.1 Flow of the Tool

The Issue Tracking Tool provides user with an intutive UI, that helps them to select criteria to pull out the relevant data from HSD Database(eg: Plaform on which the issue is filed, name of the customer, priority of the issue,etc). The data retrived then is re-structured and presented in various categories depending on the issue status (New/Debug/Resolved etc). Additionally the tool automates the next set of activities that is needed to take the issue forward till closure.

AE's can sort the issues according to their requirement and paging is also applied in the gridview. Number of issues in one page can be handled by AE's in the Page

Size module. There is one search field in every state Tab in which AE's can search any issue.

The Flow of the Tool will be like :

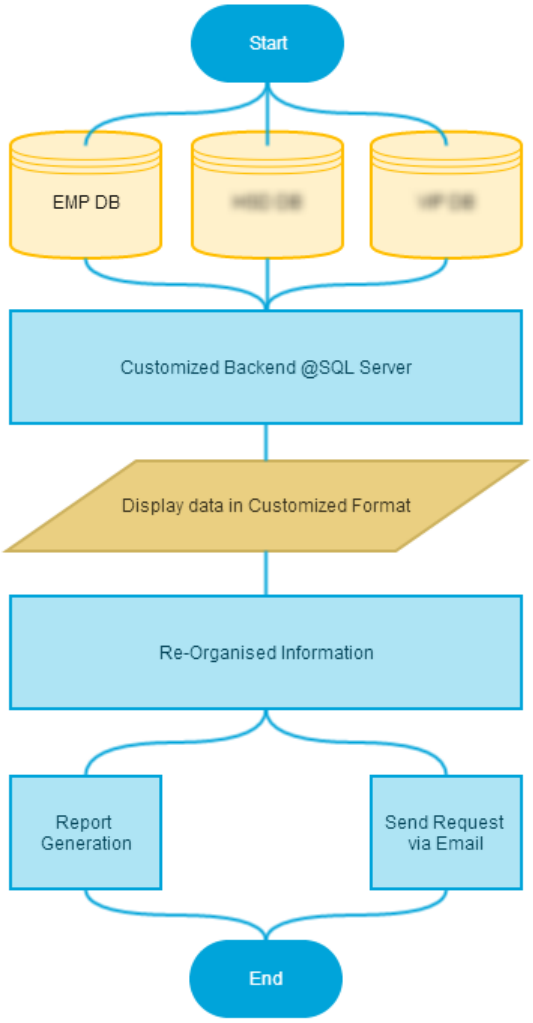

Figure 3.1: ITT Flow

On selecting an issue, A new Tab view will be generated in which there will be various fields like Debug Prioritization for sending the issues to get Debug Priori-

tized, FIX Prioritization for sending the issues to get FIX Prioritized, Critical Issue Fix Template to send a template to high level manager about the Critical Issue Fix of a particular Issue, Escalation Heads up for sending higher level manager a heads up regarding the future escalation which may come from OEM's or ODM's, Replication Request for sending the Issue details to employees for replicating the issues.

The Data will be fetched from the database and displayed into the various fields of the tabs described above. Then AE's can send those via Microsoft Outlook to concerned person to get the work done faster and easier.

HSD Database is highly confidential. It contains the information of all the designs which are going to launch in future and are still dis-closed in the market. It also contains the information about the up-coming platform on which Intel is working to launch in future. Hence, in order to access the database, a very secured medium is required to fetch the details. Hence HSD API is created in order to provide a secured medium to access the HSD database. It is explained in brief in the Features sub-section.

3.3 Features of the Tool

3.3.1 Secured Access to Highly Confidential HSD Database

The biggest challenge was to make this tool live. There were thousands of hurdles in between to make this tool live. We were not getting the permission to access the database due to security reasons. We tried all the possibility to access the HSD Database. Then the owner of the Database recommended us to make an API which can securely take the credentials of an authorised person to access the database from our tool. Hence I build this HSD API which is used to access the HSD Database securely.[3]

The Flow Chart of working of HSD API is : **How to read Data from HSD Focus:**

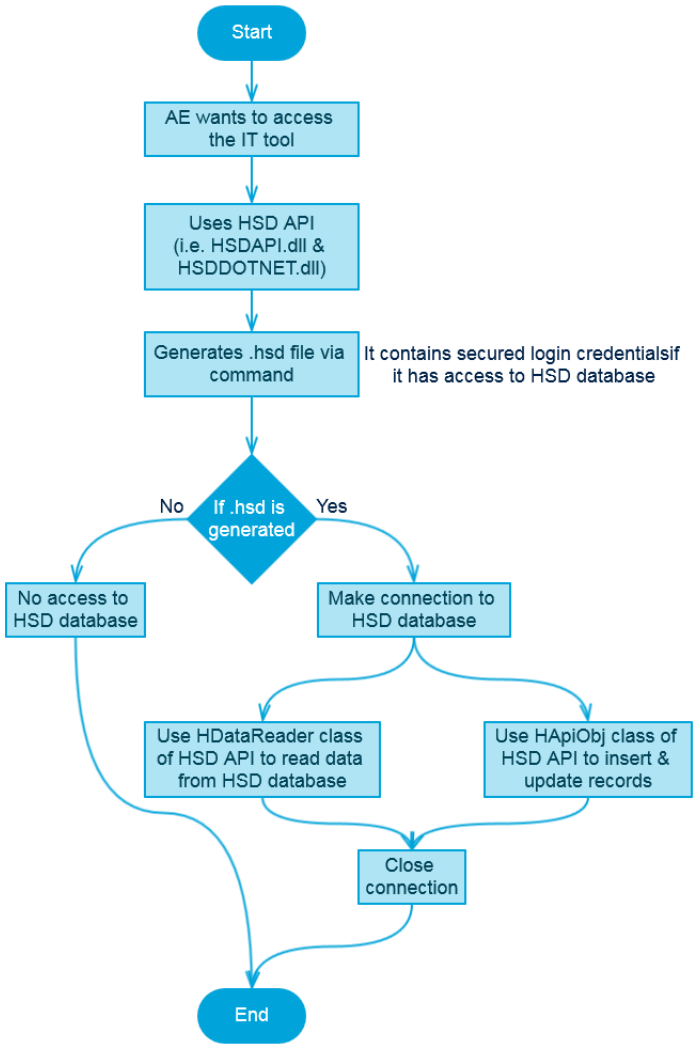

Figure 3.2: How HSD API Works

The HSD API is an internal API for developing application programs that need to interact with databases. The HSD API allows a .Net/ C developer to interact with the database and simplifies certain tasks such as login, querying, saving queries under

a user's profile and other basic tasks. These allow a programmer to accomplish tasks without knowing about finer details about the HSD Database working.[3]

Its an Application Programming Interface for accessing the HSD database, in a sense, the automation engine for HSD. Using the HSD API with one of the programming languages, one may create programs which may query data from the usual HSD databases. [3]

HSD API Availablity :

- Available on Windows + Unix platforms
  - Unix: C++ and Perl
  - Windows: C++ and .Net

Connecting to a HSD database:

- HSD Api connects to a particular HSD database using its dsn name.
- DSN name are of this format:
  - hsd_project_[pre]
  - project: refers to Project name. (skylake/haswell/nehalem etc)
    - TIP: If you do not know project name; look at the hsd web url. It will be listed as part of the url: vthsd.intel.com/hsd/projname
  - Mode: lets the API know which database you want to connect to: staging or production. Staging dsn would end with _pre.
- Examples:
  - hsd_skylake: SKL production hsd db
  - hsd_skylake_pre: SKL staging db

Authentication is done by:

- Each HSD API object takes in an optional username/ password.
  - This could be your hsd username/password or Windows domain login/password
- If you do not wish to provide your login credentials in code (most common use case); then set up your .hsd file
  - This file contains your hsd username password
  - Located in your unix home directory (or %appdata% HSD_HOME on windows)
  - Generate new .hsd file
    - /p/vt/tools/hsd/tools/account/release/common/bin/hsdsetup -p proj -m mode
      - proj: your project name from hsd dsn. For example; if dsn is hsd_haswell; then proj = haswell. hsd_skylake: skylake
      - mode: mode of the database. pre (staging) or prod (production)

How to read Data from HSD Focus:

Smart Query support :

- Allows you to query related focii without knowing the join relations between them.[3]
- For example; you can say
  - select turnin.id, bugeco.title where turnin.submitted_by = nsshergi
- The API will translate this to
  - select turnin.id, bugeco.title FROM [bugeco_turnin_rl] with(nolock),[turnin] with(nolock),[bugeco] with(nolock) WHERE (([bugeco].id=[bugeco_turnin_rl].[bugeco_i AND ([turnin].id=[bugeco_turnin_rl].[turnin_id])) AND turnin.submitted_by = 'nsshergi';

```
#!/usr/intel/pkgs/perl/5.8.5/bin/perl
use lib '/nfs/site/proj/vt/tools/hsd/api/2.0/release/lib';
use strict;
use HSDApi;

my $dsn="hsd_demo_pre";
my $reader = new HDataReader($dsn);

$reader->execQuery("select top 5 id,title,status from bug");

my $fieldCount = $reader->getFieldCount();

while ($reader->read())
{
        for (my $i = 0; $i < $fieldCount; ++$i)
        {
                print "[" . $reader->getFieldName($i) . "]  -> " . $reader->getValByIndex($i) . "\n";
        }
        print "\n\n";
}
```

Figure 3.3: How to read data from HSD Focus

- Note: no need to specify the from clause

Insert data into HSD Focus:

Update existing data in a HSD Focus :

The Complete Steps are :

- Get access permissions to the db you'll be writing applications for
- Create your $HOME/.hsd file
- Start with a simple (Perl) script
- General SQL select command
- Add exception handling

Get access permissions to the db you'll be writing applications for:

- Simple, obvious, but can be forgotten if you're a DA developing an application for a customer project. Ask that project's HSD DA for access. For your first few coding iterations, better to only have read access - you don't want to accidentally modify out records. (If you do, they can be restored of course).[3]

```
#!/usr/intel/pkgs/perl/5.8.5/bin/perl
use lib '/nfs/site/proj/vt/tools/hsd/api/2.0/release/lib';
use strict;

use HSDApi;

my $dsn = "hsd_demo_pre";
my $apiObj = new HApiObj();

$apiObj->init($dsn);

# Create a group for the records you are going to
# be working with. You can provide an optional
# group name
my $group = $apiObj->createGroup("bug","New bugs");

# To insert a new record, get a new rec object
my $rec = $apiObj->newRec();

$rec->setVal("title","New bug for testing");
$rec->setVal("description","testing insert from api");
$rec->setVal("status","New Bug");
$rec->setVal("owner","nsshergi");
$rec->setVal("validator","nsshergi");
$rec->setVal("filed_by","Other");
$rec->setVal("instruction_to_reproduce","Test run by new API customer");
$rec->setVal("assigned","BUS.BBL");
$rec->setVal("send_mail","no");

if ($rec->commit())
{
        print "Inserted Bug ID: " . $rec->getVal("id") . "\n";
}
else
{
        print $rec->getError() . "\n";
}
```

Figure 3.4: Insert data into HSD Focus

- Create your $HOME/.hsd file
  - When your application connects to the database, HSD server will want to verify that you are who you say you are and that you have permissions. To avoid hard-coding your credentials in the script, create a hidden file $HOME/.hsd. You do so by running:
  - /nfs/site/proj/vt/tools/hsd/tools/account/release/common/bin/hsdsetup -p projname -m mode
  - projname can be obtained from the URL to your hsd: e.g. http://vthsd.intel.com/hsd/n means your project is nehalem. mode is either 'prod' or 'pre'.

HSD API Authentication:

- HSD API connects to the database using your hsd username and password. This information is stored in a .hsd file. On unix; this file is located inside your home directory. For Windows; the file is saved in the %APPDATA%/HSD_HOME folder. For example; C://Documents and Settings//your_username//Application Data//HSD_HOME. If you do not have a .hsd file present; then you will need to manually provide your hsd username and password in your code.[3]

```
#!/usr/intel/pkgs/perl/5.8.5/bin/perl
use lib '/nfs/site/proj/vt/tools/hsd/api/2.0/release/lib';
use strict;

use HSDApi;

my $dsn = "hsd_demo_pre";
my $apiObj = new HApiObj();

$apiObj->init($dsn);

# Create a group for the records you are going to
# be working with. You can provide an optional name
my $group = $apiObj->createGroup("bug","My bugs");

# Load some data into this group
# 1st parameter - name of group
# 2nd parameter - Where clause (criteria)
# 3rd parameter - fields to load (optional)

# Sample query: Load all bugs that are open
# apiObj->loadGroup(group,"status = 'Open'");

# For this sample; I will load 10  bugs
# that are assigned to nsshergi
$apiObj->loadGroup($group,"owner = 'nsshergi'");

my $numRecs = $apiObj->numRecords();
print "Loaded: " . $numRecs . " Records\n";

# Lets modify these 10 records
for (my $i = 0; $i < $numRecs; ++$i)
{
        my $rec = $apiObj->getRec($i);
        print "About to modify bug id: " . $rec->getVal("id") . "\n";

        $rec->setVal("title","testing - please ignore");
        $rec->setVal("send_mail","no");

        # We can commit the changes back to the database
        # one rec at a time; or just commit the whole group
        # at the end of the loop. For this sample; we will
        # Commit changes 1 at a time
        if ($rec->commit())
                print "Updated ID: " . $rec->getVal("id") . " successfully.\n";
        else
                print $rec->getError() . "\n";

        print "\n";
}
```

Figure 3.5: Update existing data in a HSD Focus

- IF No HSD File present
  - HDataReader hr = new HDataReader(”hsd_demo_pre”,”my hsd username”,”my hsd pasword”)
- If I have a .hsd file
  - HDataReader hr = new HDataRader(”hsd_demo_pre”);

How to get .hsd file:

- **On Unix; please run this utility to get a .hsd file.**
  - /nfs/site/proj/vt/tools/hsd/tools/account/release/common/bin/hsdsetup -p projname -m mode
  - The project is your hsd project. For example; it could be nehalem/penryn or anything. The easist way to figure out your project code is to take a look at the hsd url for your project. If you are connecting to vthsd.intel.com/hsd/nehalem; then your project is nehalem. If you are connecting to vthsd.intel.com/hsd/test_123; then your project is test_123.[3]

  - Mode refers to the mode of the project. Possible values are prod (production) or pre (staging).
  - For example; if you wanted to get access to Haswell; call it like this. (Note: use haswell instead of hsw)
    * /nfs/site/proj/vt/tools/hsd/tools/account/release/common/bin/hsdsetup -p haswell -m prod
  - If you are unable to get access using hsdsetup; then you need to contact your project owner so they can grant you access. HSD team does not manage access to database. This resposiblity is delegated to project owners. After you get access; you can run hsdsetup again to get your .hsd file.[3]
  - Please note that for running the hsdsetup utility on a SLES11 machine, you need to set the environment variable LD_LIBRARY_PATH as setenv LD_LIBRARY_PATH /nfs/site/proj/vt/tools/hsd/vendor/gcc/3.1/lib

Windows :

- A .hsd file will be automatically created the first time you run any HSD API programs. If you are running into issues creating a .hsd file; then you can use the standalone HSDAccountTools.exe utility. Make sure you run the command below from Command Prompt. Do not run the exe file as it is.[3]
  - HSDAccountTools.exe -setup -proj demo -mode pre **OR** HSDAccountTools.exe -setup -proj demo -mode prod
- This will create a .hsd file for you.

Faceless accounts

- If you are using a faceless account to connect to HSD; you will not be able to use the Account utilities mentioned above to create a dot hsd file. You will need to contact the HSD team to get the HSD password for the faceless account. Once

you have that; just make a text file called .hsd in your home dir and manualy put in the hsd username/password. The format of the file is as follows[3]

- [default] username=hsdusername password=hsdpassword

3.3.2 Global HSD Database Search - HSD Google

HSD is a huge database which contains TBs of data and lacs of records. Many times, AE has some specific information from which it becomes very difficult for them to find the issue details. They usually create Query from the Query Builder where they enter their information and fetch the details. From this tool, it will be very easy for them to find the issue details.

They just need to write any value in the Search field, and the HSD Google will crawl the complete database and search the most relevant data and presents them the details.

3.3.3 Integration of DCN and CDI Tool

We have integrated two existing Intel's Internal Tools.[2]

- DCN Tool[2]
- CDI Tool[2]

The Main objective of integrating these tools is to simplify their day to day activities. These tools are frequently used by AEs in the team for tracking the Intel Graphics Issues. Hence it becomes time consuming to navigate from one site to other for different activities. Hence Integrating these tools helps them increasing the efficiency and decrease the time taken to get work done.

Introduction to DCN Tool:

- DCN Tool helps AE to find on which Driver, the Issue got Fixed.[2]
- For every Graphics Issue, there is one DCN Number attached to it and saved in the Database. Next time, when any other similar graphics issue comes up, AE first looks up in the database to find the DCN number of the previous issue which contains the details of the Driver version in which the issue got resolved.[2]
- This helps them a lot to reduce the same cycle again to resolve the issue. i.e Replicating/Debugging/Developement/Bug Fix/etc...
- Hence integrating DCN tool helps them a lot in making their day to day life better.

Introduction to CDI Tool :

- CDI Tool helps AE to download Intel Confidential Documents which contains the Intel Graphics Information[2]
- Every time, AE needs to login into the tool to download the documents due to its confidentiality[2]
- This integration helps them to acess the tool automatically and download the documents without any login which reduces the time a lot.

3.3.4 Dynamic Template Generation for Easy Issue Tracking

Apart from the Intel Graphics Issue Tracking, there are lots of day-to-day follow up activities, which an AE has to manage.

The Activities includes :

- Sending the issue for Debug Priorization. Whenever an issue comes to Intel. We need to set the priority of the issue for debugging. AE needs to make a manual

template to send the issue for Debug Prioritization. This Tool helps them to dynamically generate the template and send it to the concerned person.

- Sending the issue for TIL Prioritization. Whenever an issue got root caused by the Debug Team. We need to set the priority of the issue for sending it to the development team. AE needs to make a manual template and answer several set of questions to send the issue for TIL priorization. This Tool helps them to dynamically generate the template and upload the data in Excel sheet and send it to the concerned person.

- Sending the Must Fix Request to Manager. Whenever any new request/modifications in the Graphics Driver comes from the customer before the release.

- Sending the Potential Escalation Headsup to Manager. Many times, to get the issue prioritised due to release date Customers escalate the issue to higher manager. Hence If AE knows that this issue can get escalate in near future, he needs to give a headsup to his manager regarding the same. For that They need to make presentation regarding the Potential Escalation Headsup and the reason. This Tool helps them to dynamically generate the presentation and send it to the manager for a particular issue.

- Sending the Escalation Form to Manager. Whenever an issue gets escalated, Manager needs to be informed and include him in the follow up activity till the issue gets resolved. Hence this tool helps them to maintain the history of escalation and the follow up activities too.

- Sending the issue for Replication. All the Graphics Issues needs to be replicate in Intel Environment so that they can find the root cause of the issue. Hence everytime sending the Issue Details, Steps to Replicate, Information like GOP Version, OS, Design name, etc.. to employees is very time consuming and annoying. This Tool helps to automatically sends the information to the employee

which contains all the information which is needed to replicate any issue.This saves a lot of time.

Hence, Automation of these activities is helping a lot in increasing the overall efficiency.

3.3.5 Automatic Login using Active Directory

- This Tool is capable of identifying you automatically through your system name.
  - It uses Active Directory services to authenticate and authorize you to access the tool. Hence, no login required.
  - Fetches your Name, Alias, Email address and all related information from Microsoft Outlook.
- Prevents access to unauthorized users.
  - Those who dont have access to HSD,they will not be able to access the Tool.

3.3.6 Maintaining the History of Escalations

- Provides an easy way for AE/Management to maintain/search history of escalations per customer/platform for future reference
- Managers would need the number of escalations per customer/platform
  - It can be used as a metric to track AEs performance
  - Helps them to track the progress of an escalated issue
- A local database is maintained for storing the data

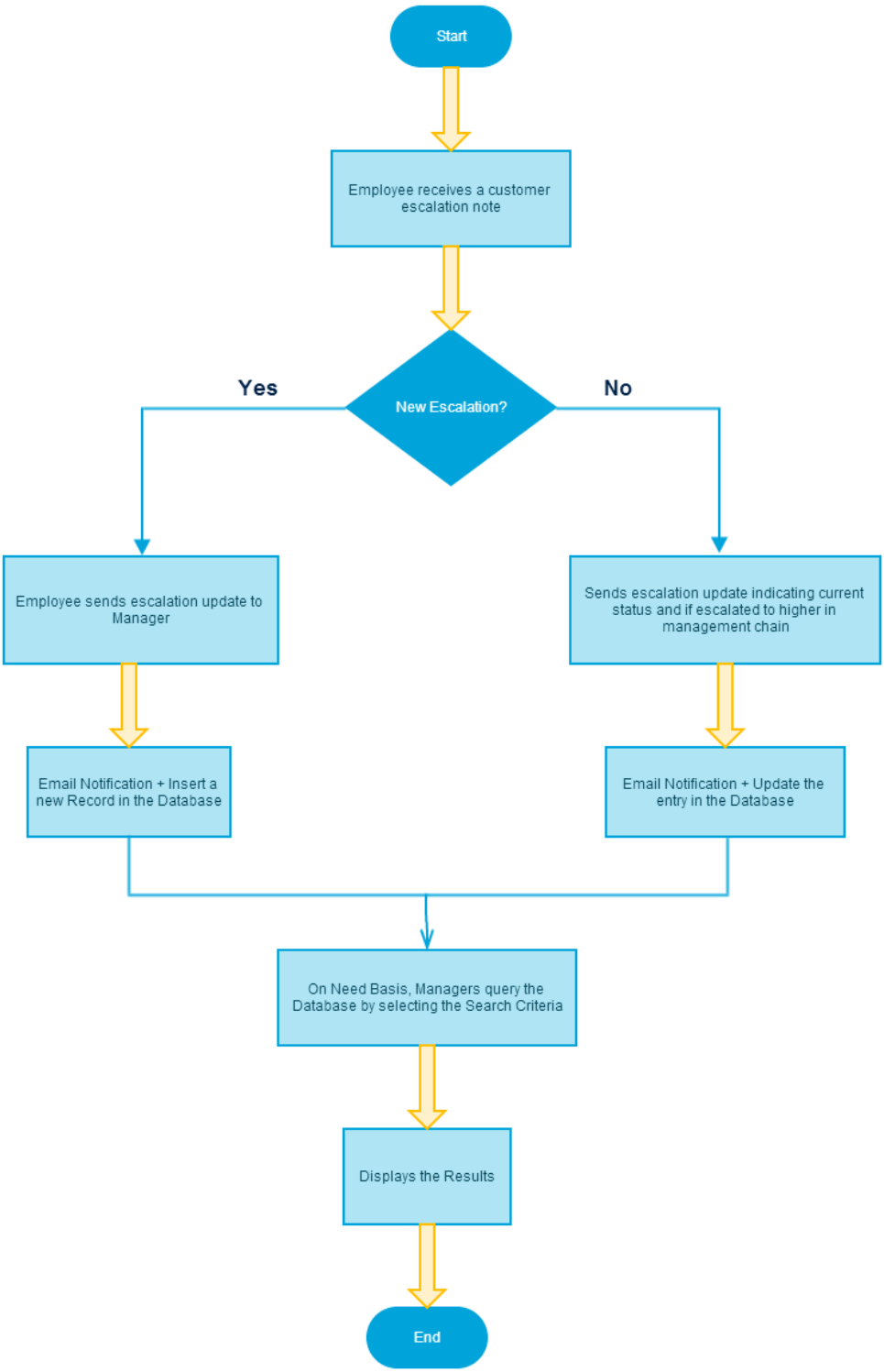

Figure 3.6: Flow of Maintaining Escalation History

Chapter 4

Results

This chapter discussed the results obtained as an outcome of this project. The chapter shows the degree of efficiency that the tool developed is able to achieve. The chapter also discusses how the tools has affected the day to day working of an AE and how it has and will help in speeding up AEs work.

4.1 Snapshots of Issue Tracking Tool

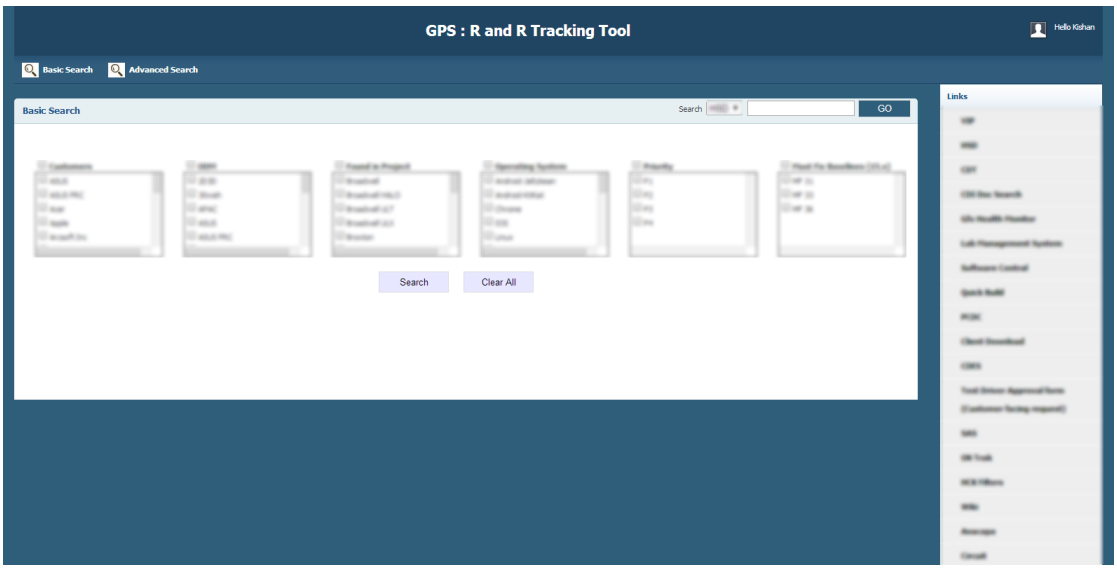

Figure 4.1: ITT Home Page

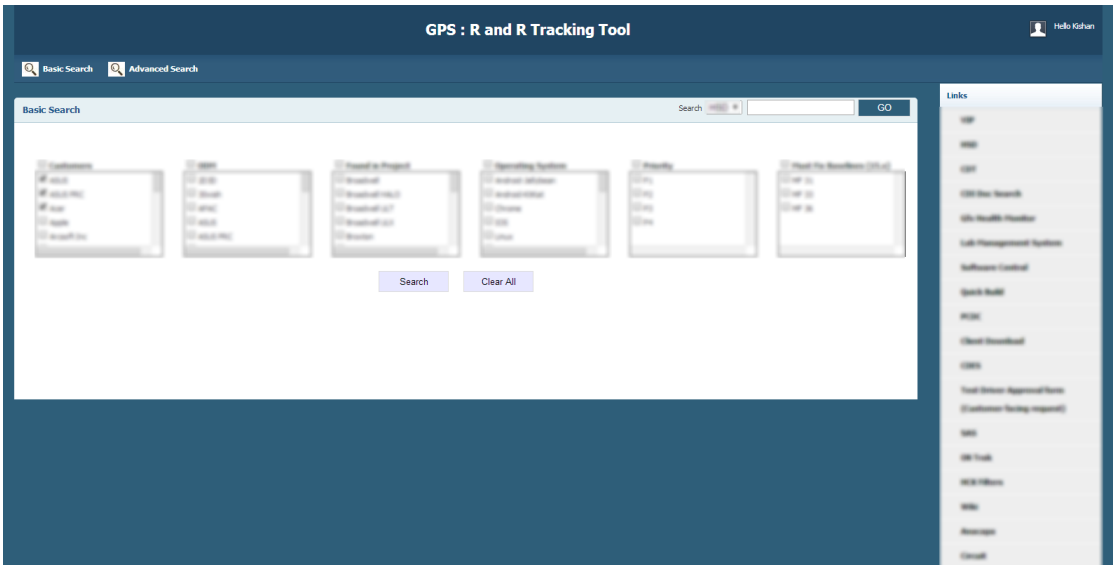

Figure 4.2: ITT Customer Selected

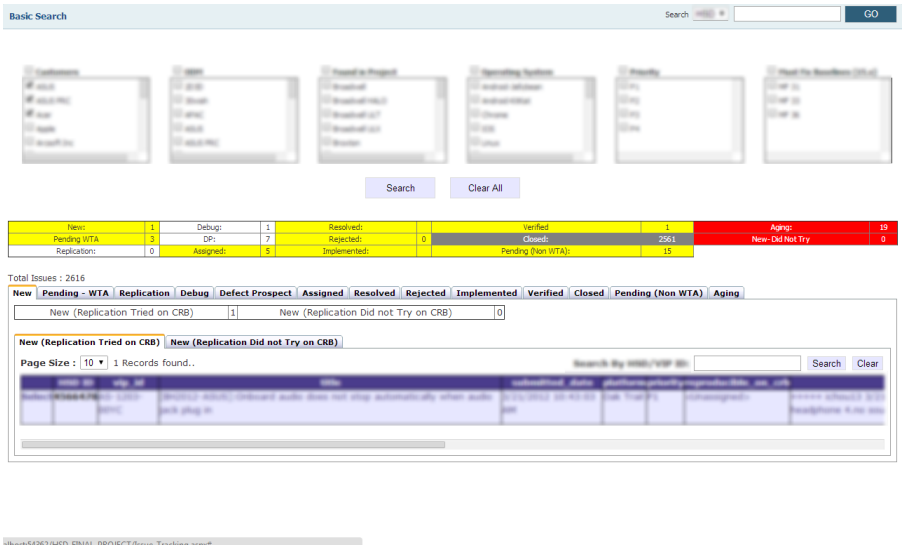

Figure 4.3: ITT Issue Details in New State

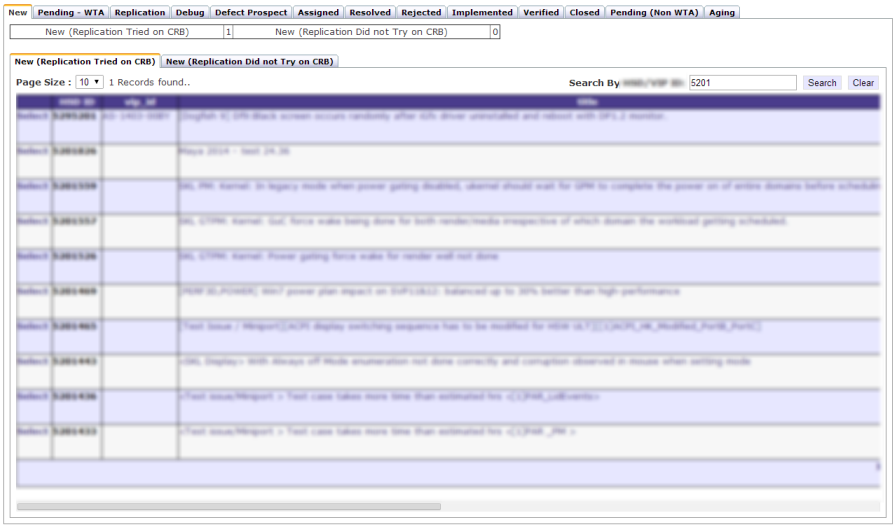

Figure 4.4: ITT Search Issue Details in New State

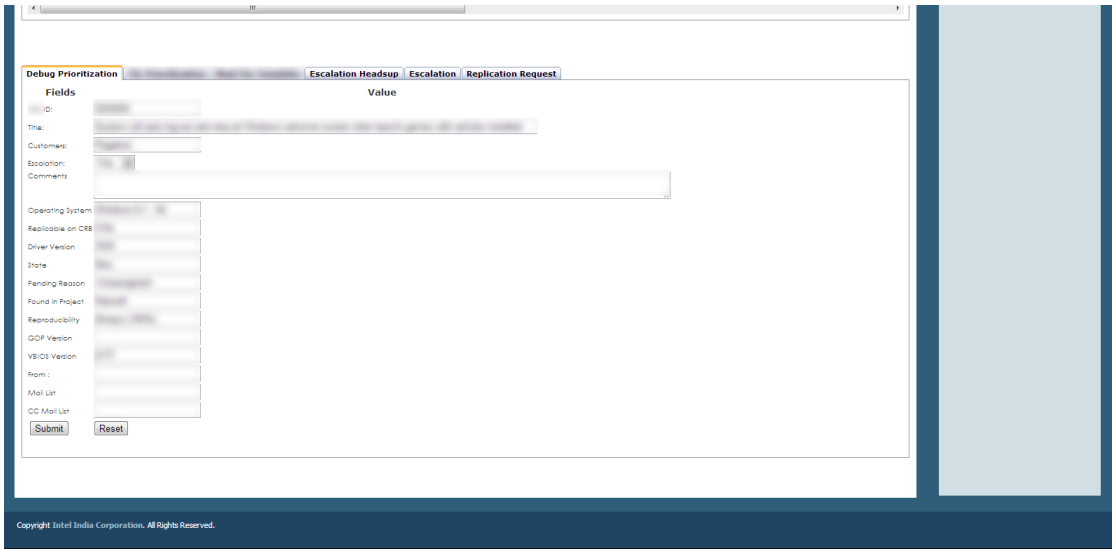

Figure 4.5: Debug Prioritization Form

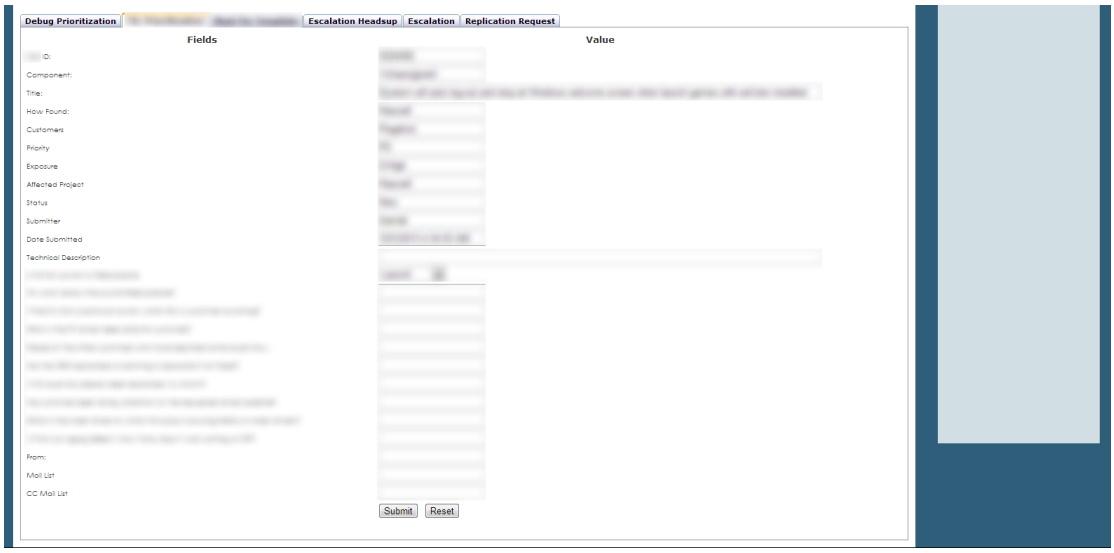

Figure 4.6: FIX Prioritization Form

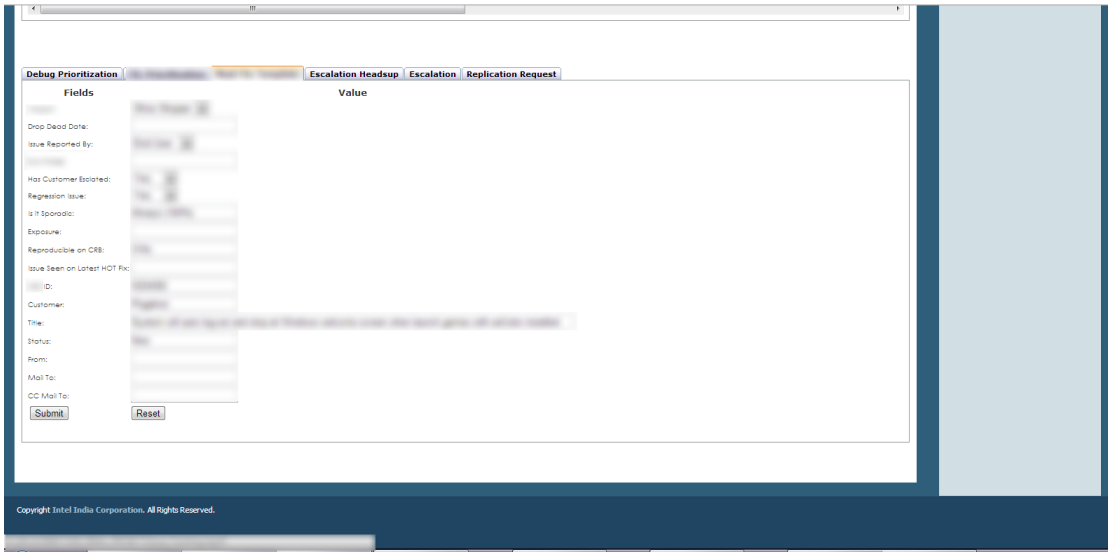

Figure 4.7: Critical Issue Fix Form

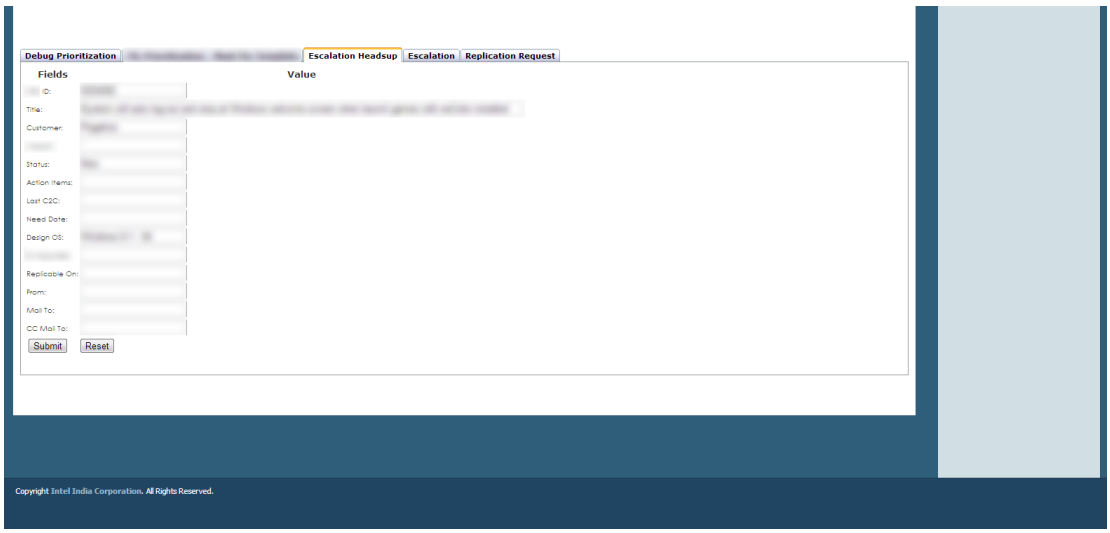

Figure 4.8: Escalation Heads-up Form

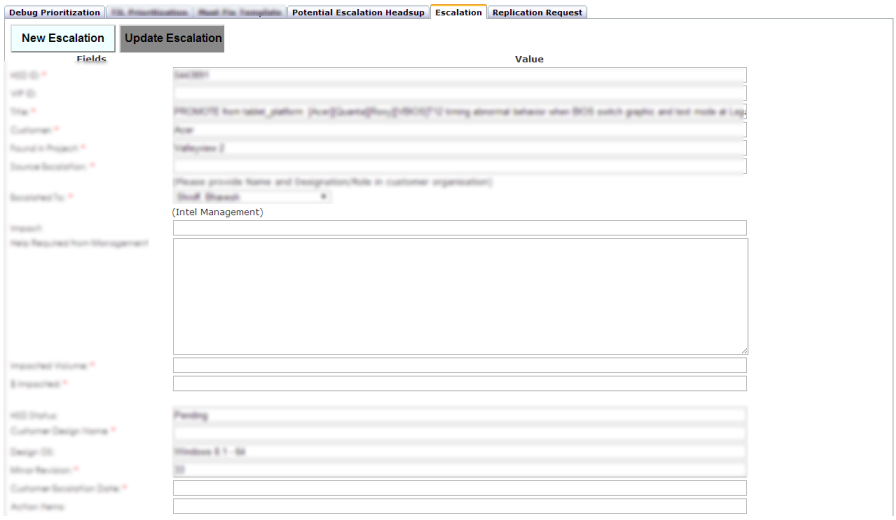

Figure 4.9: Escalation Form

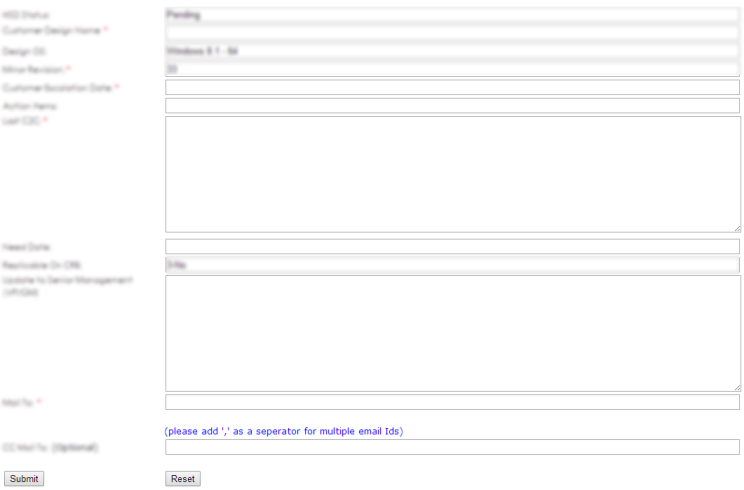

Figure 4.10: Escalation Form

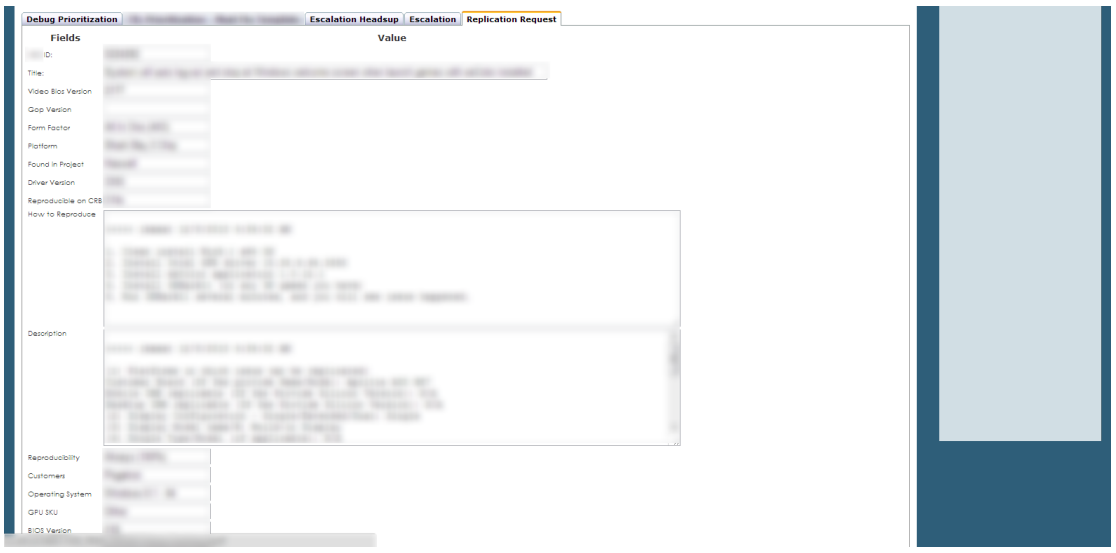

Figure 4.11: Replication Request Form

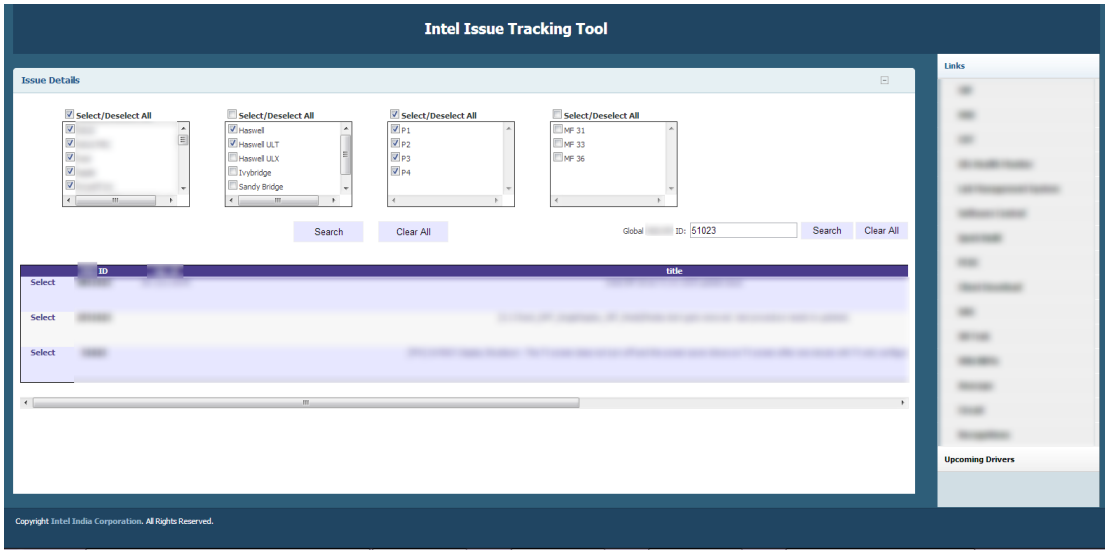

Figure 4.12: ITT Global Search

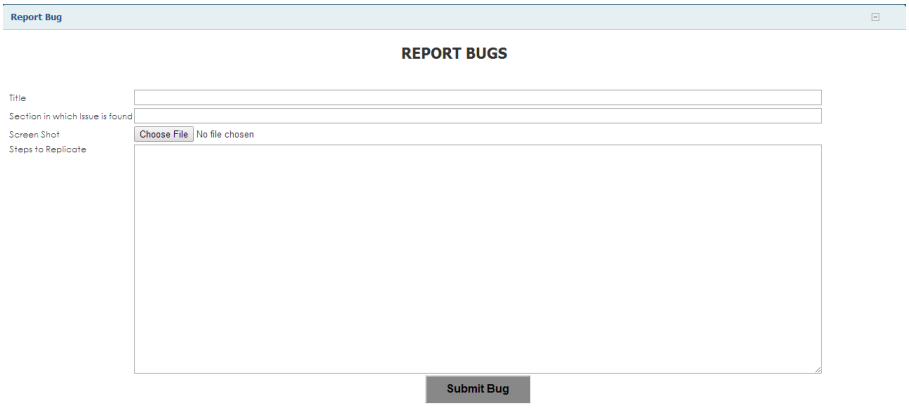

Figure 4.13: ITT Bug Fix Form

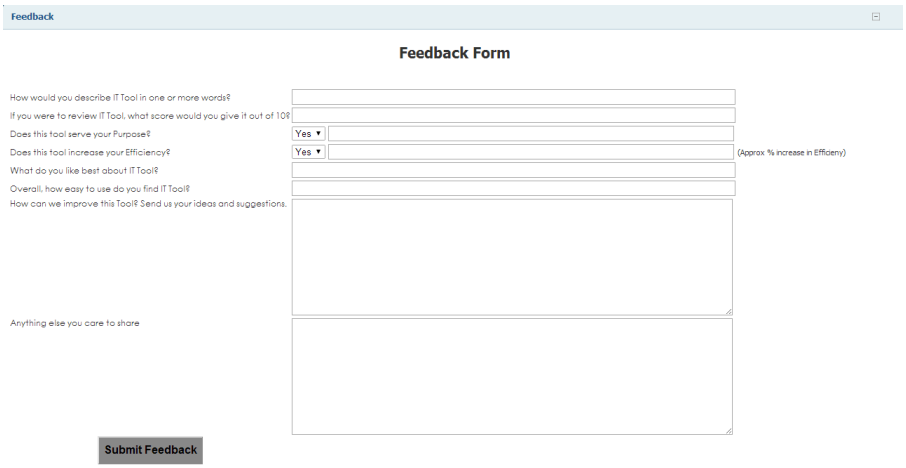

Figure 4.14: ITT Feedback Form

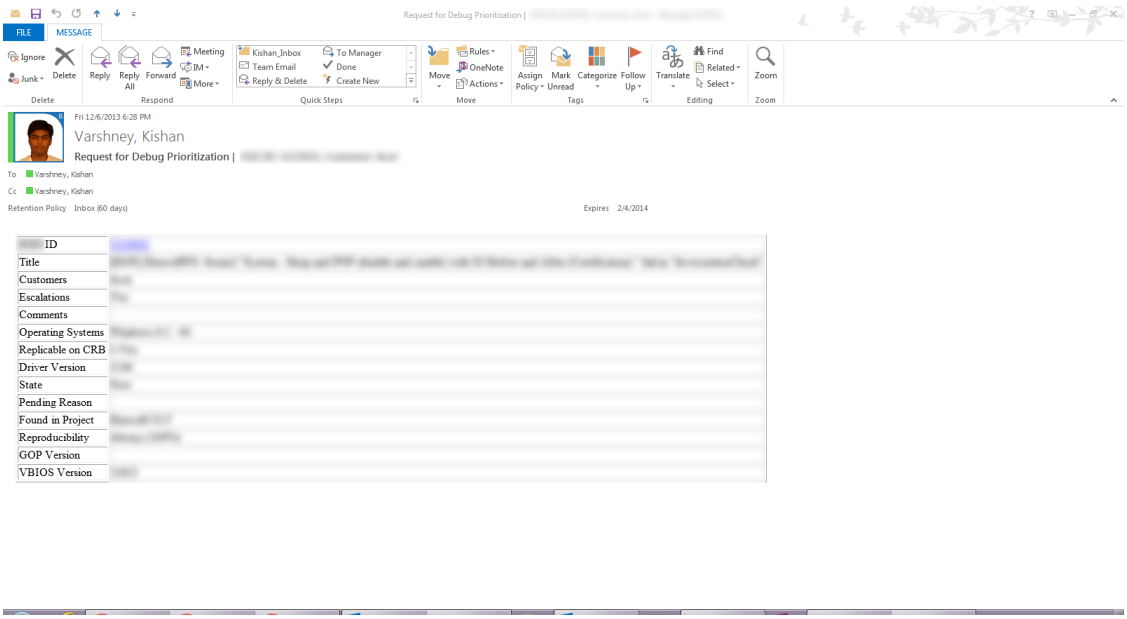

Figure 4.15: ITT Mail Format

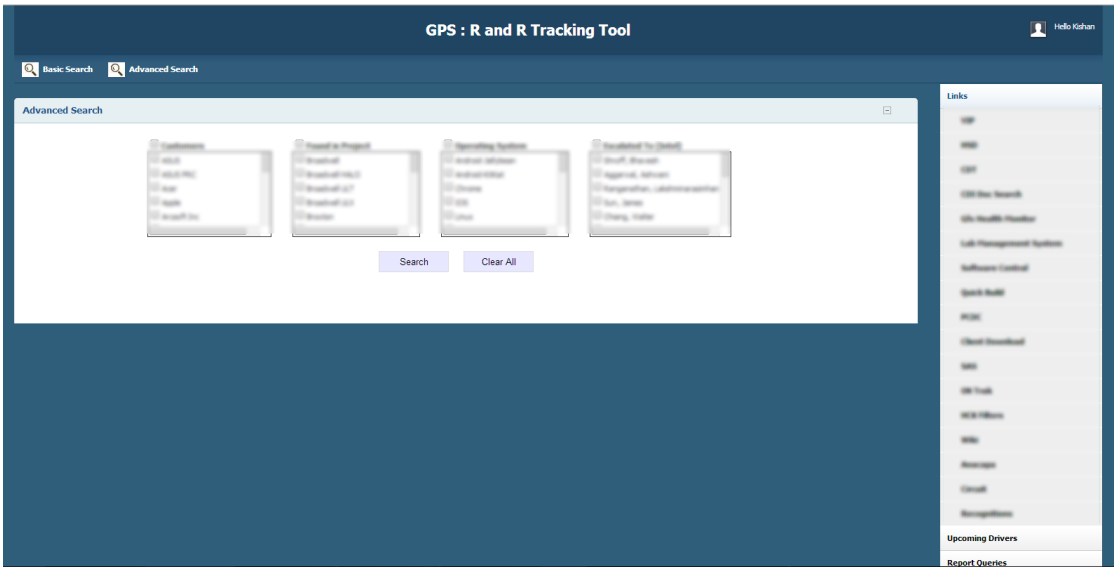

Figure 4.16: Escalation History GUI Form

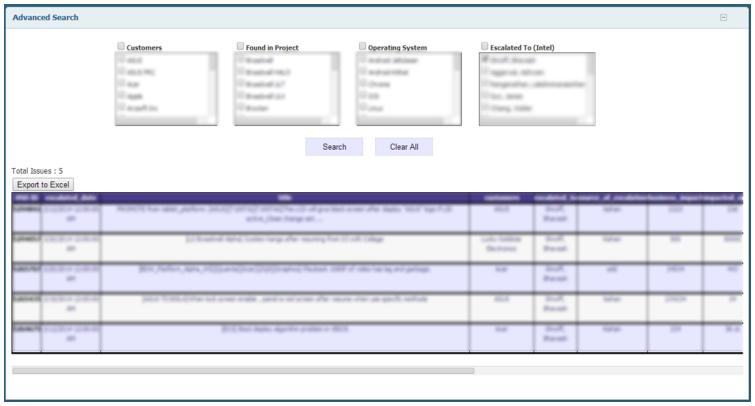

Figure 4.17: Escalation History GUI Form

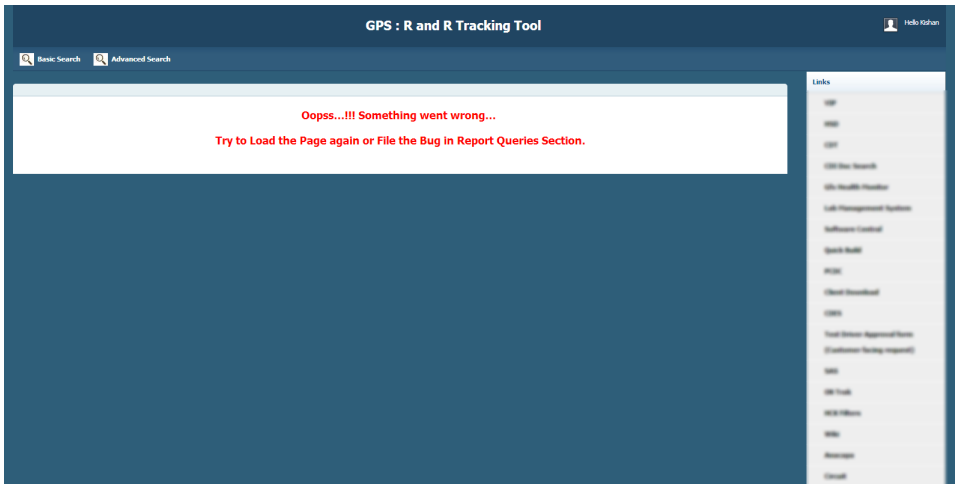

Figure 4.18: Exception Handling Page

Chapter 5

Result Analysis

5.1 Result Analysis of IT Tool

Following are the Result analysis of the IT Tool. It shows the time taken by AEs to do manual work and the same work done by the IT tool. It shows you the efficiency increase because of this tool.

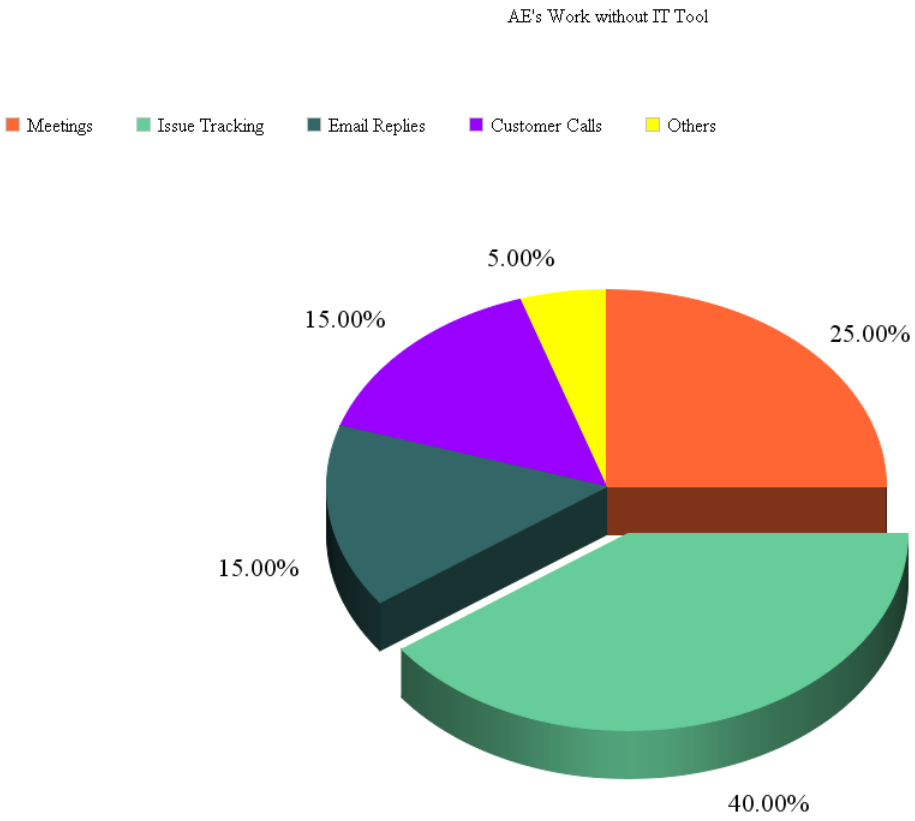

Figure 5.1: AE's work without ITT Tool

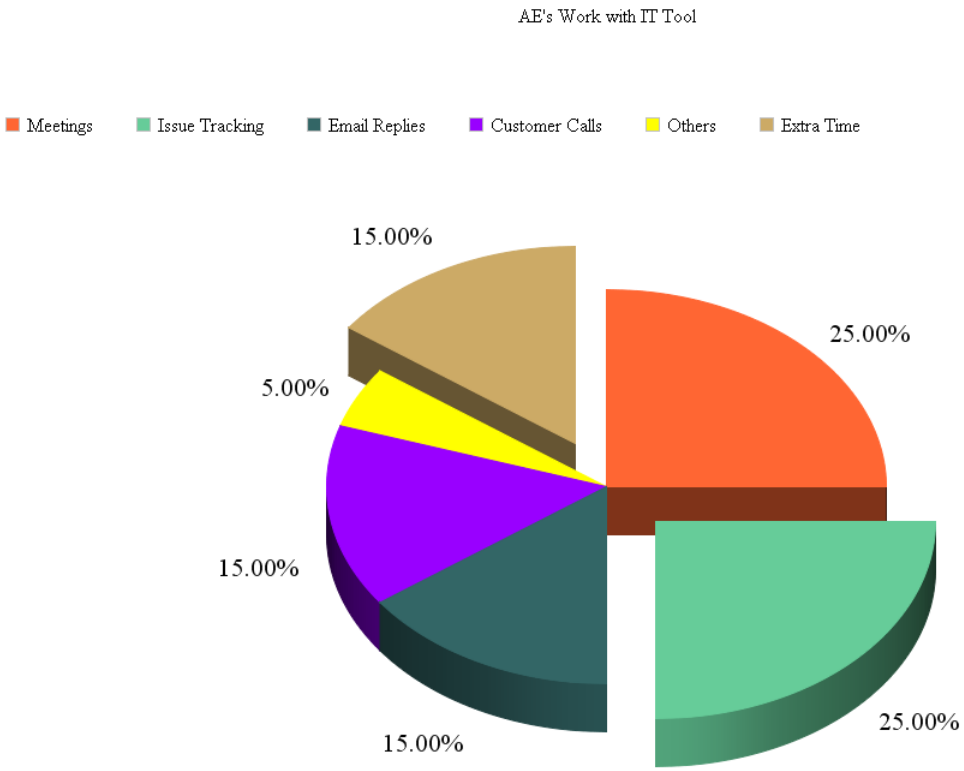

Figure 5.2: AE's work with ITT Tool

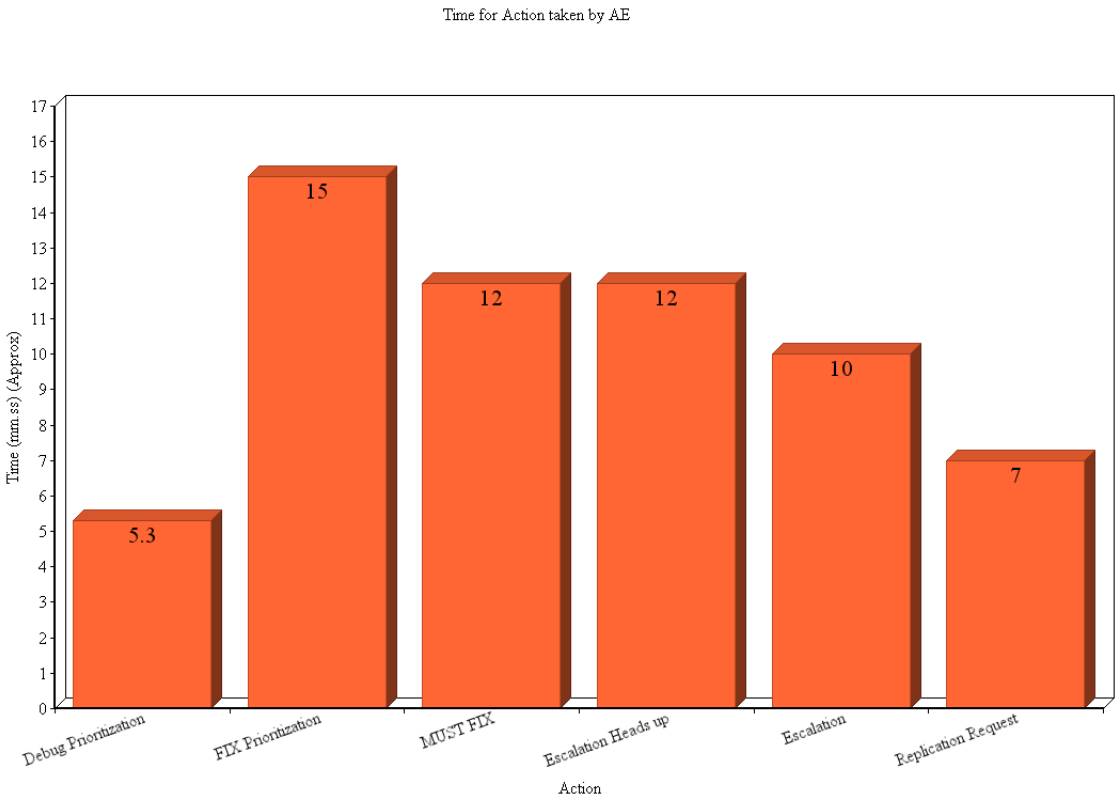

Figure 5.3: Time for Action Taken by AE

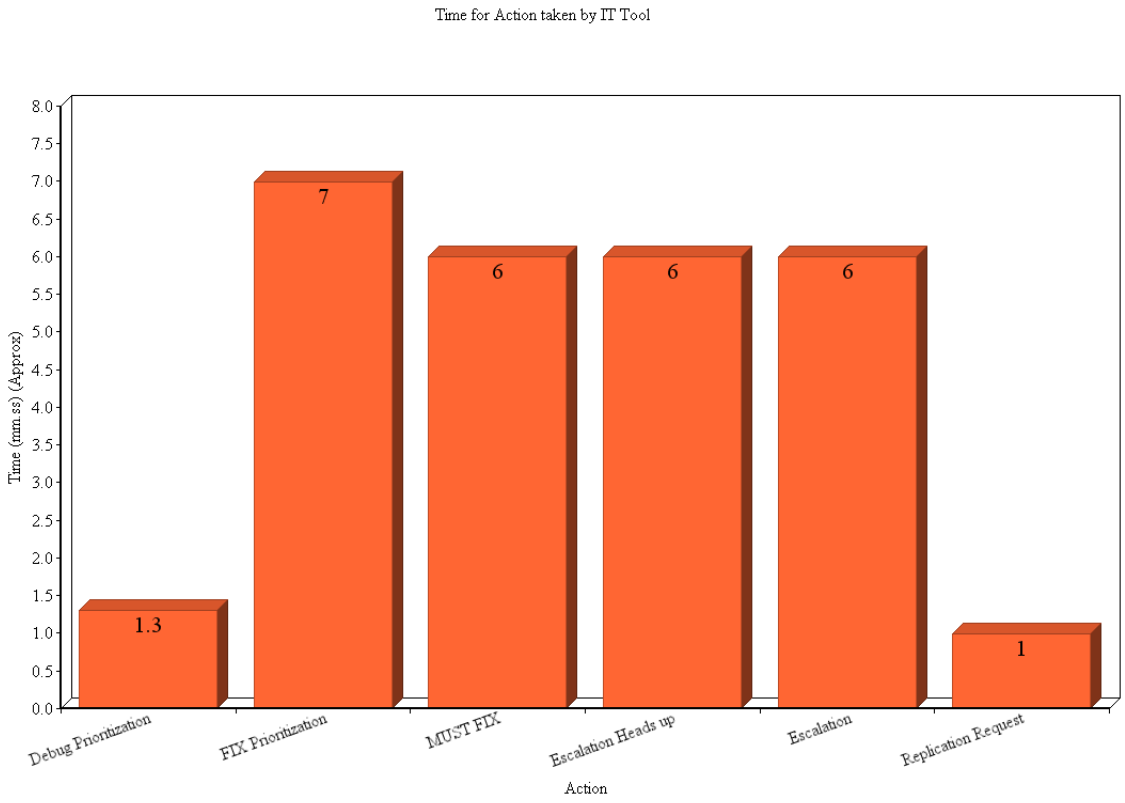

Figure 5.4: Time for Action Taken by ITT Tool

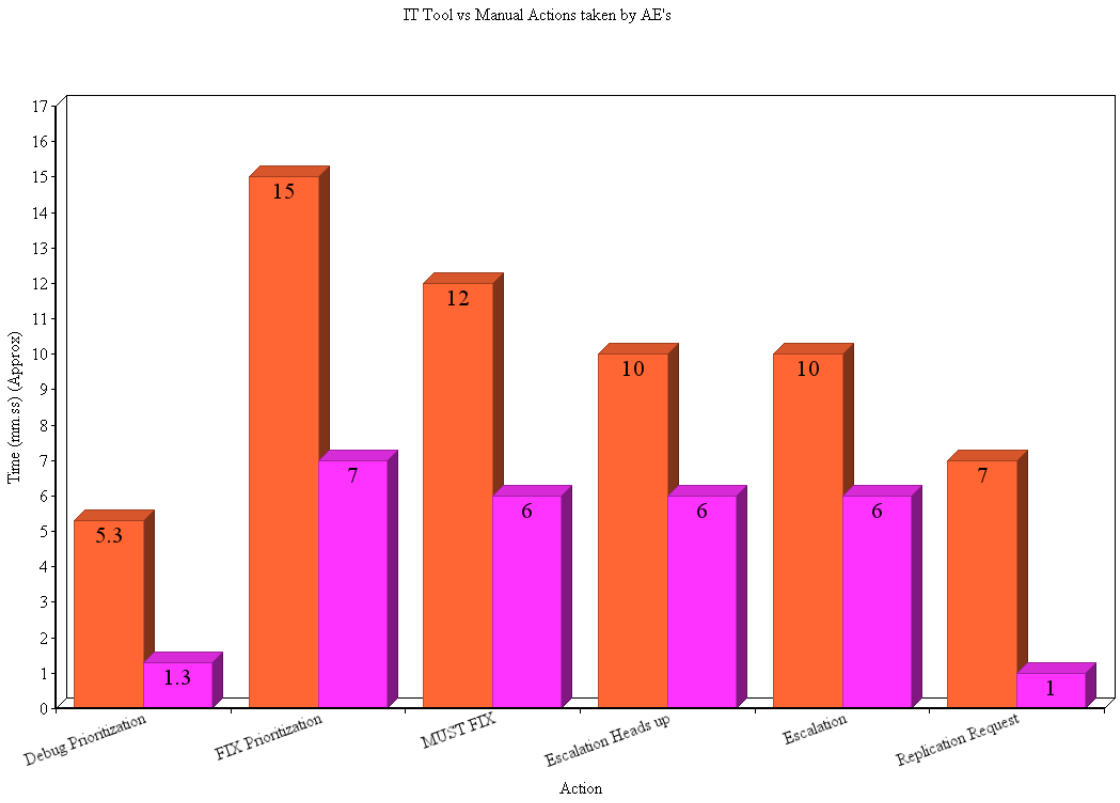

Figure 5.5: ITT Tool vs Manual Actions taken by AE's

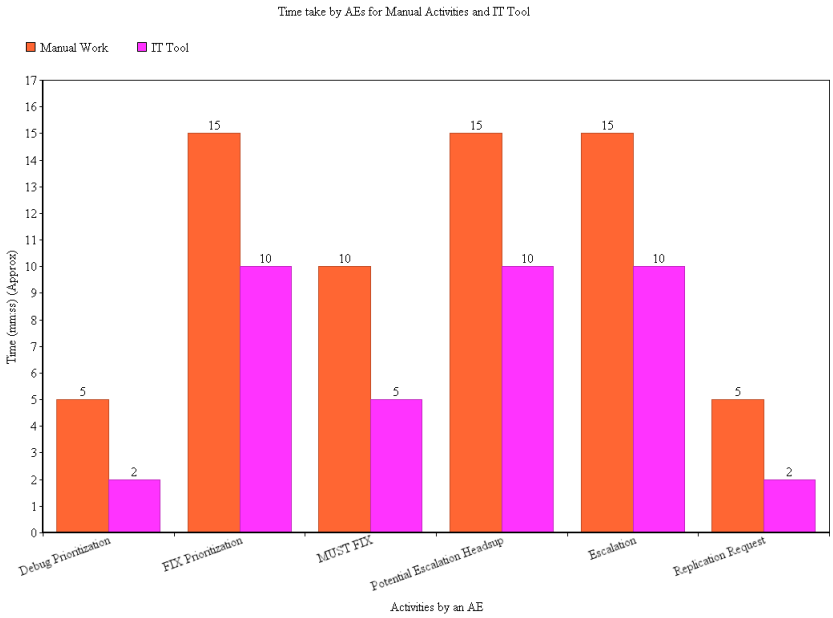

Figure 5.6: Time Taken by AEs

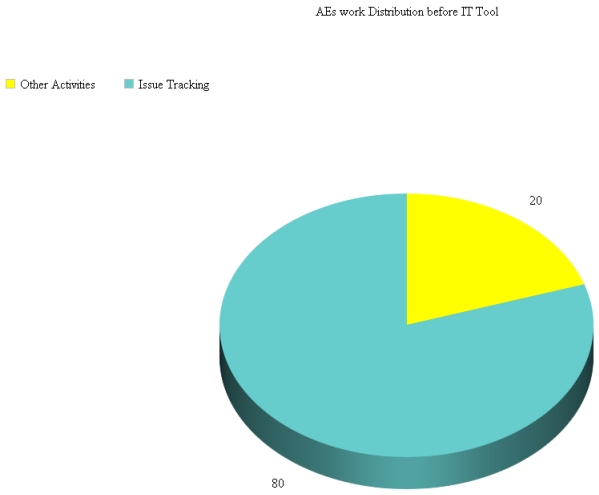

Figure 5.7: AE's work without IT Tool

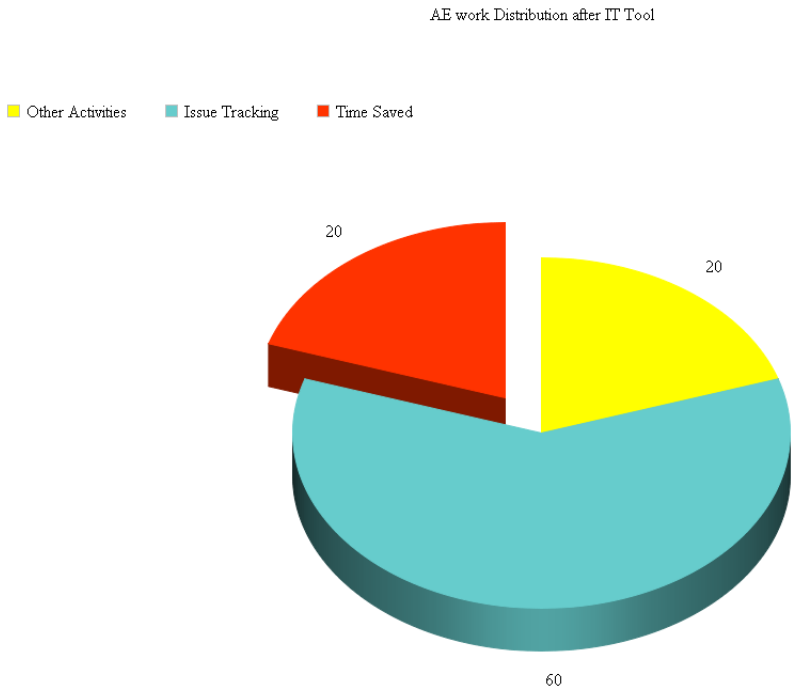

Figure 5.8: AEs work with IT Tool

Chapter 6

Conclusion and Future Work

IT Tool proves to be a very efficient tool in terms of issue tracking. I took the survey of 150 AEs out of 504 in my team and they found the tool very effective and useful. According to them, this tool helps them to increase the efficiency by 30% in their day to day activities.

This tool is extremely secured because of HSD API which is used to authenticate the user and database. Its highly impossible for anyone outside intel or even inside intel who dont have access to database to access this tool. Apart from this, after normalising the database, the speed of the tool has become very fast.

Bibliography

[1] Intel Internal Documents on HSD and Training Manuals.

[2] Intel Internal Tools.

[3] Intel Internal Research Papers.

[4] URL: www.freetds.org

[5] URL : http://msdn.microsoft.com/en-us/library/aa286485.aspx

[6] URL : http://studytonight.com/dbms/database-normalization.php

[7] URL : http://databases.about.com/od/specificproducts/a/normalization.htm

[8] URL : http://databases.about.com/od/specificproducts/g/multivalued-dependency.htm

[9] URL : http://en.wikipedia.org/wiki/Active_Directory

[10] URL : http://msdn.microsoft.com/en-US/library/aa746492(v=vs.85).aspx

[11] URL : http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol