



Scrap & Rework Tracker

SRS Document

Scrap & Rework Tracker	4
Introduction	4
Management Summary	4
Low hanging fruits	4
Key Assumptions	5
High Level Architecture	5
Functional Requirements	6
All teams	6
Admin	6
Management	6
Use Case Diagrams	6
Use Cases	7
<i>Upload Masters</i>	7
<i>Define Activity Framework</i>	7
<i>Post Daily Activity</i>	9
<i>Track Scrap and Rework</i>	9
<i>MIS - Critical to Address</i>	10
Example Data of Activity Framework	11
Logical Object Model	11
Database Design Guidelines	12
Testing Approach	12
Suggested Technical Reading	13

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

Scrap & Rework Tracker

Introduction

The purpose of this document is to define scope and requirements a Tracking Scrap & Rework in the manufacturing process of a mid size eco friendly devices manufacturer. The new system proposes to start recording the scrap being introduced or generated in the system and thereby working on eliminating waste.

This document should be used by the development team to architect the solution the project.

Management Summary

The Scrap and rework costs are a manufacturing reality impacting the organization and its product lines. The management got to the root cause as to what is causing this out of proportion Scrap and rework costs. It was analyzed that there are various factors contributing to this situation. Some of them are wrong parts are ordered, when engineering changes aren't effectively communicated or when designs aren't properly executed on the manufacturing line. No matter why scrap and rework occurs, its impact on an organization is always the same—wasted time and money. No one in the complete chain owns up this responsibility, these expenses add up quickly and negatively impact the bottom line. To get a hold on this situation, the company decided to have a defect tracking at all the critical points of the execution. This application would be simple to use and will facilitate collation of data and perform quick and simple analysis to drive corrective action. Implementing such a solution would bring about the sense of ownership at various levels which is grossly missing currently.

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Low hanging fruits

Most of the production houses bank on their quality teams to create the metrics and set up processes to record the data or collect data from automated systems and generate MIS to facilitate managing waste or rework. In this project, the organization is small and does not have formal quality processes or team in place. Thus creating a small framework which is easy to use on daily basis will improve the overall tracking of the problem on hand. Also, being in IT, its important to keep evaluating what can be done to catch the low hanging fruits (here problems) that are staring in the face. Remember, the organizations expect IT professionals to come up with quick fixes while the bigger solutions are being evaluated or planned for.

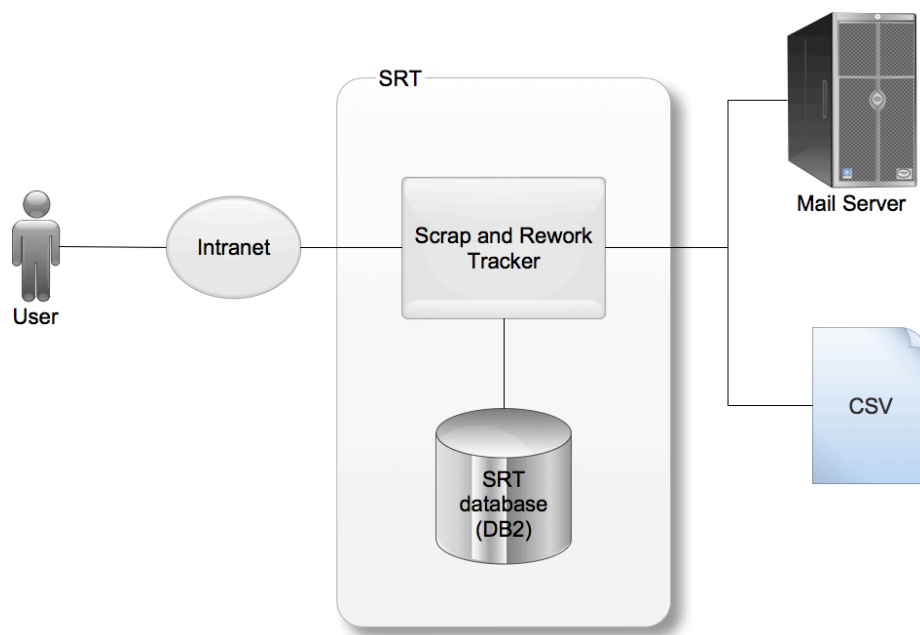
Key Assumptions

1. Departments and Employee master is uploaded using CSV
2. Knowledge of capturing and analyzing defects would be useful for the developer. Read about manufacturing processes, they are hard core business processes, having exposure to them will strengthen domain knowledge, will act as first principles to be used else where to arrive at solutions.
3. There are 5 departments namely, 1) Design, 2) Sales, 3) Procurement, 4) Production and 5) Despatch in the chain of executing orders for Eco friendly devices.

High Level Architecture

Scrap & Rework application's high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Administrator
2. Department Managers
3. Team (Department wise)



Scrap and Rework Tracker Context Diagram

Scrap and Rework Tracker (SRT)

The system will allow all the departments to enter their key activities and frequently encountered bugs or defects. The associated reasons as a reference list which keeps building as and when the reasons are added while recording a bug or drilling down to find the problem. The system will let the user link the reason to an activity of the same or some other department. This will show linkages to trace the root causes.

SRT Database

Stores Activities, Defects, Reasons, Analysis data for various departments

CSV	Employees, Departments, Parts
Mail Server	Notifications are sent using the mail server

Functional Requirements

The high level functional requirements for the Scrap and Rework Tracker are outlined in the Use Case diagram described in this section.

Scrap and Rework Tacker will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by the web applications. Login will be a prerequisite to use this application.

Once user logs in, as per their profile, the landing page options are displayed.

All teams

1. Post Daily Activities
2. Track Scrap and Rework

Admin

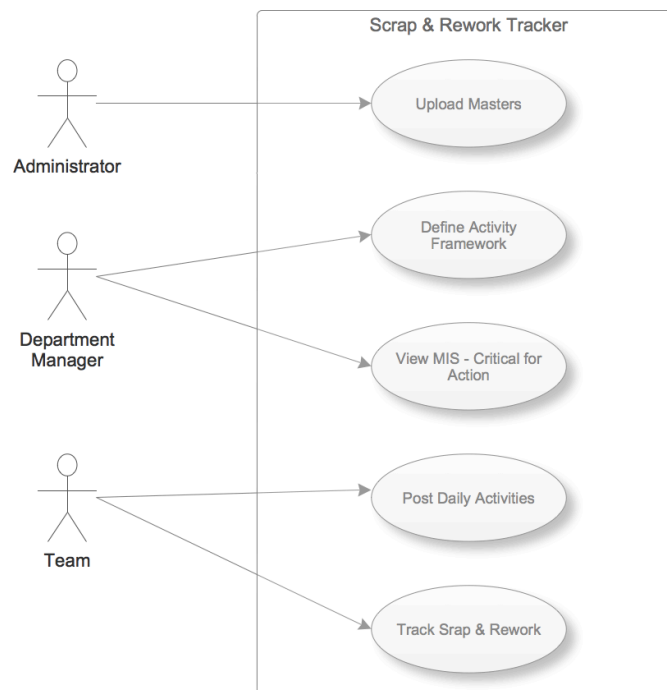
1. Upload CSVs

Management

1. Define Activity Framework
2. MIS - Critical to Action

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system. The MIS use cases are not detailed here.



Use Case Diagram

Use Cases

Upload Masters

Use Case Element	Description
Number	UC.01
Application	<p>Masters are uploaded using CSV format.</p> <p>Department master will contain Department id and Department Name (key departments are Design, Sales, Procurement, Production, Packaging, IT)</p> <p>Employee master will contain Employee id, Employee Name, Department</p> <p>Part master will contain part id, part description, unit of measure, lead time, lot size.</p>
Use Case Name	Upload Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the Upload Masters menu item on the admin interface page
Basic Flow	<p>System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system.</p> <ul style="list-style-type: none"> The selected file data is uploaded in the related masters; if an existing record is encountered, the old details are replaced with the new details.
Alternate Flow	<ul style="list-style-type: none"> In event of incorrect CSV format, system gives an error and NO data is uploaded. Operation is cancelled
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

Define Activity Framework

Use Case Element	Description
Number	UC.02
Application	<p>All the teams are required to enter the key activities they perform, the type of defects that normally occur or can be anticipated and the reasons attributed for the defects. This acts as a master for all the users of the department while they are entering the daily activity log.</p> <p>Entry happening in this framework will be a one time effort if done thoughtfully, else department users can add more activities or errors to the existing ones.</p>
Use Case Name	Define Activity Framework
Primary Actor	Department Authorized User
Secondary Actor	None
Pre-condition	None

Use Case Element	Description
Trigger	User clicks on the Define Activity Framework menu item on the application landing page.
Basic Flow	<ul style="list-style-type: none">• The system displays a form with following information to be entered:• Name of Activity• Known or Possible Errors encountered• Reasons that can be attributed <p>Each activity data is saved with Department id of the logged in user. The process is repeated till all the trackable activities are entered by all the departments.</p> <ul style="list-style-type: none">• The above information will form basis for daily entry of information for these activities. <p>Please refer to example data for understanding what type of data is required to be entered.</p>
Alternate Flow	<ul style="list-style-type: none">• User abandons the operation, no database impact
Output	Display list of activities mapped for the logged in department only on completion of entry.

Post Daily Activity

Use Case Element	Description
Number	UC.03
Application	All departments follow a discipline of posting their daily activities and their outcomes for tracking the Scrap and Rework in the operations.
Use Case Name	Post Daily Activity
Primary Actor	Authorized Department Users
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Post daily activities menu item on the Home page
Basic Flow	<ul style="list-style-type: none"> The department name, Today's date and logged in users name are displayed on top. The system displays list of activities defined in the activity framework for the logged in user's department. User can select one or more activities performed from the list. A tabular structure is displayed to take inputs for each activity. <ul style="list-style-type: none"> Activity Name (Display only) Work Order : Enter work order for which this activity was performed Outcome [Successful, Rework, Scrap] Select Defect - displays defect for this activity defined in the framework, user can select from there Reason - Displays reasons for the defect selected Link to any other activity : Display list activities recorded as defects for the work order entered above. Click on the activity to be linked for this defect. Repeat this process till all the activities are entered On save the complete data is saved for the design with columns like Department id, Employee id, Today's date, Activity id, Work order no., Outcome, Defect, Reason, Linked Department id, Activity id. <p>To add more complexity, fields like part number reference, cost incurred for defects only etc. can be added</p>
Alternate Flow	<ul style="list-style-type: none"> Cancel abandons the operation, no database impact
Output	<ul style="list-style-type: none"> None

Track Scrap and Rework

Use Case Element	Description
Number	UC.04

Use Case Element	Description
Application	The system will generate various perspectives from the activity data entered by each department. Each department can view the tracker for their department only. Though information shared across would be useful, due to limited scope this feature is not included where a comprehensive dashboard is displayed and each department then can take up discussions arising from root cause linking.
Use Case Name	Track Scrap and Rework
Primary Actor	Authorized Departments
Secondary Actor	Management
Pre-condition	None
Trigger	User clicks on the Tracker menu item on the home page
Basic Flow	<ul style="list-style-type: none"> The view generated is for the logged in users Department only. In case a management user views this option, the system will display a list of departments to select from. Activity count of - Successful, Rework, Scrap Ratio of Rework and Scrap generated on Activities Work orders with maximum Defects Work orders with no defects Reason attributed maximum times Other Department Activity linked maximum times
Alternate Flow	<ul style="list-style-type: none"> Cancel abandons the operation, takes back to the previous view
Output	Displayed on the screen

MIS - Critical to Address

Use Case Element	Description
Number	UC.05
Application	Which few defects are accounting for maximum number of scrap or rework, the system uses 80/20 principle to discover vital few. Targeting a fix on those defects will substantially reduce the Rework and Scrap volumes thereby saving costs.
Use Case Name	MIS Critical to Address
Primary Actor	Management
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on MIS link on the landing page
Basic Flow	<ul style="list-style-type: none"> System displays a table as well as a Pareto chart depicting the following List of defects, Frequency of Occurrence, Cumulative Frequency, % of Cum frequency <p>The top few defects that are accounting for 80% of the problem are the ones that need to be addressed immediately.</p>

Use Case Element	Description
Alternate Flow	<ul style="list-style-type: none"> If no data is available the chart or table will not show up
Output	<ul style="list-style-type: none"> Tabular view and Pareto Chart

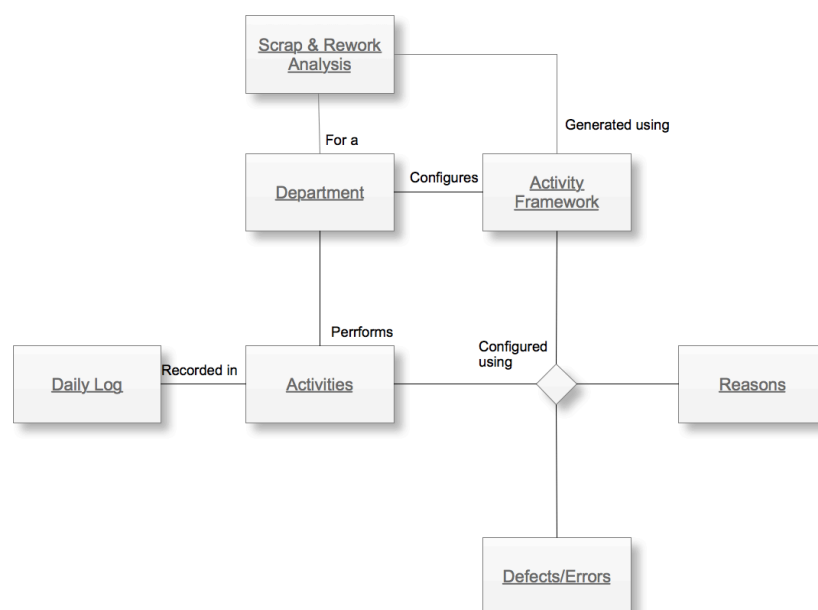
Example Data of Activity Framework

Here is an example to illustrate the concept using an over simplified data for key activities performed by department.

Department Name	Activity	Defects	Reasons
Sales	Proposal for Customer	Incorrect Product Specifications	New specs not communicated by Design team
Sales	Order Entry into the System	Incorrect Qty / Product code entered in the work order	Negligence
Sales	Order Entry into the System	Incorrect Qty / Product code entered in the work order	System not updated with correct Product codes
Design	Defining BOM for a Product Design	Missed some part numbers in BOM creation	Part numbers do not have correct description in the master
Design	Makes changes to Product BOM	Delay in communication to production and purchase	Process not followed for communication post any changes to product design
Procurement	Raised PO for a work order	In correct qty	Referred to qty entered in WO by sales team

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. The unit functions with multiple departments.
2. Each department executes work orders using unique set of activities.
3. Each activity produces an output that is either correct or has defects.
4. Defects are assigned reasons, they are like possible causes for the defect to have occurred.
5. The activity framework allows departments to configure the activities and define the possible defects that occur or can occur. The defects are aligned to known reasons for their occurrence.
6. Each department enters the daily log of activities as configured in the framework. Any defects generated as an outcome of this activity is selected from pre-defined list. For each of the defect recorded, the reason attributed is selected from a pre-defined list of reasons.
7. The Scrap and Rework analysis is generated from the Daily log entry using activity framework.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.
2. All persistent classes that use the database for persistency must map to database structures.
3. Many-to-many relationships must have an intersecting table.
4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.
5. Indexes should be defined to optimize access.
6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI

Common Checks	Validation Type
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mm/yyyy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, WebSphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/