CHAPTER 5

HYBRID ARCHITECTURE OF AGILE RISK REMOVAL ANALYSIS (HA-ARRA) MODEL

5.1 INTRODUCTION

Agile architecture is a set of values and practices that support the active evolution of the system design while implementing new system which has capability with sequences of the process. HA-ARRA Model approach allows the architecture of a system (even a large project) to evolve overtime process, while supporting the current users simultaneously. HA-ARRA Model avoids the overhead in the risk factors that delays associated start-stop-start nature and redesign the inherent phase-gate methods. With this agile architecture, the system always runs continues supporting flow of value and finding the defect flow in the coding phase. The agile architecture (HA-ARRA Model) supports with 12 principles of agile, lean-agile development through collaboration, intentional standard architecture, design simplicity to meet the customer requirements as well as designing for testability, deploy ability and reliability of the product. It is further supported by rapid prototyping, domain modeling and decentralized innovation of agile methodology.

Emergent design of HA-ARRA Model: The model provides the technical basis for a fully evolutionary and incremental implementation approach. HA-ARRA Model helps designers respond to immediate user or customer needs, allowing the design to emerge as the systems built and deployed to meet the customer requirements.
Defect removal and risk analysis are the foremost threading points in software development process in industries. Defect removal analysis to facilitate monitoring and mitigating the risk factors during the early stages of software development. Defect removal and risk analysis are the key points that have important components present in the software need to be analyzed for application or system oriented language. Therefore, the 3-Tier Architecture of Hybrid Architecture of Agile Risk Removal Analysis (HA-ARRA) Model has been established. The main objective of this model is to search the components that help to reduce the development time, cost, the entire possible defect identification, removal and risk analysis factors which are encountered during the various development phases of the project. HA-ARRA model is to find the defects related to defect risks analysis and facilitates adaptability analysis for various environments to application oriented software development.

This HA-ARRA model is analyzing the defect removal process (defect identification and raking the defect for removal) and finding the risk impact level (risk identification ad raking) of component and encountered during the usage and such dynamic components of a system transfer to another environment of the project. HA-ARRA play a very important role in considerably reducing defect and risk in the development of software as well as the time and cost of a system in part manner. HA-ARRA model helps to analyze risks factors implicated process in any system after the application and reused the defect problem without much deviation of the project. To ascertain a variety of risks factors analysis and impacts are occurring in a system on account of software industries. It makes the system efficient by focusing and evaluating, test case analytic on all the relevant risks factors which might occur due to the impact of software e components and making the defect and risk ranking system to avoid the further vulnerability or threads.
5.2 RATIONALE OF THE HA-ARRA MODEL AND ARCHITECTURE DESIGN

Defect and risk is a primary base of threat in any software development process as well as suitable for agile methodology. Risk analysis is another major factor of defect identification and defect removal analysis to facilitate the mitigating and monitoring the risk factors during from the begin of the project development each phases. Defect removal ad risk analysis are dual dominate qualities of software development which creates the continues vulnerabilities and threats affect to influence the end product of the customer required and without any pre-define articulate of the system and it turn to arising to failure of the project at time of delivery the product. Therefore, Hybrid Architecture of Agile Risk Removal Analysis (HA-ARRA) Model has been established in this research. The main objective of this HA-ARRA model is to find out harmful defect removal, remove the defect components are ranked that help to reduce the risk factor analysis which contain the entire possible risk factors enervating and destructive testing of project, agile testing, risk ranking approach are undergone through this hybrid architecture is shown in Figure 5.1.

![Figure 5.1 Hybrid Architecture of Agile Risk Removal Analysis (HA-ARRA) Model](image)
5.3 MODEL DESCRIPTION OF HA-ARRA MODEL

If any software component is find out the adaptive analysis to the proposed to other environment through their checking analysis process this three tier architecture (HA-ARRA) Model, where as the Tier1: to make analysis the defect identification and ranking, removal using risk analysis is using severity and priority using matrix method and test case work sheet Method, as the risk defects removal are always much costlier when it found in the later stage of project and checked by the testing team’s responsibility to find the risk defects removal and to classify the defects based on its severity and priority for the testing team to make sure the defect needs to be fixed, logged defect and for the required functionality to go in the current release to removing of defect begin stage of the project. It should ensure recognition of scrum process are examined through enervating and destructive testing and the impact of risk defect removal with risk ranking based approach to removing the bugs, error or faults or vulnerabilities of the software project. If it does not, then it should undergo the process of risk reduction through monitoring and mitigation. The process involves risk identification and ranking due to non adaptability of scrum process. (HA-ARRA) Model should check for the enervative and the destructive testing (Singaravel et al. 2010). The risk level can be identified and ranking with the help of testing case analytic made based on the agile methodology.

The Tier 2, based on the information received from the testing team by test case about on the defects classification based on the ranking, risk test case calculation with help of guideline on the rank based of defect and analytic, risk ranking approach by the testing team should make the product delivered on time to the customer without any risk defect identification.

The tier 3, agile testing that help to recognize software codes present any defect in a project which carry out similar threads and any other
vulnerability environment was created or not, as well as when compared with the functionalities of the scrum process and their attributes of sprint and stories of the proposed application system. Identifying similar risk analysis that perform the same functions that are affected the designation of all function of parameters matching of sprint and stories, which helps of the agile testing (enervative and destructive testing) to build the new structure with the defect and risk analysis of the software code and same particulars for a removing the any type defect in the project.

After identifying the defect and risk analyzed for the risk impact level in the sprint or stories board of agile process oriented software development, it can be divided into two types of classification of risks mitigation process made adaptability for agile testing; they are Enervative and Destructive testing to reduce the impact of risk in the scrum process. These two types of testing process can be defined in terms software code implementation after the recognition of the customer satisfaction.

Enervative testing is the type of test which has reduces the defect risk scrum process it does not affect on a structure of the project. Enervative testing which leads to the reduction of efficiency or increase the vulnerability of the entire task of the system. When a risk defect occurs, the reflected on the effectiveness by source code or sprint can be reducing the performance output and it may not have an effect on the normal execution of the source code.

Destructive testing is also type of testing which imperfection the entire software source code by generating unexpected outcome of the sprint. The destructive testing shows the inefficient commands to generating wrong outcome to which lead to the defect or failure of the entire sprint. The sample program is given Table 5.1 for enervative and destructive testing for sprint and program was written in C++ language.
Table 5.1 Sample source code for enervative and destructive testing of agile process

```c
#include<iostream.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    cout<<'Enter the two numbers as input variable';
    cin>>a>>b;
    for (int i=1;i<a;i++)
    {
        for (int j=1;j<b;j++)
        {
            cout<<i<<j<<"n";
        }
    }
    getch();
}
```

In the above Table 5.1 sample show that, the implementation of source code of the loop section depends on the values are given as integer variables a,b,c and it will lead to the decrease the efficiency of the source level of the process of sprint. The unexpected outcome from the sample code show and the source code is deferred from normal cases of writing the logic and this type risk testing impact called enervative testing for agile method followed by scrum.

And sample source code which is shown of Table 6.1 is also applicable for destructive testing of agile process, the loop variable a,b,c is declared as characters that output of source is erroneous and it occur the risk impact affect throughout the process as well as logic of the source code of the sprint.

Enervative and Destructive testing of agile that suggestion in detail for the risk impact level of defect removal occurs in source code of the sprint. Utilizing the software sprint implemented by developer using effectively by
reduce the defect affecting by code and increase the efficiency of the sprint in scrum process.

5.4 CHARACTERISTICS OF THE (HA-ARRA) MODEL

HA-ARRA Model, the model mainly focuses on the defect removal and risk analysis which appear in the scrum process through the test case by testing ad specific agile testing was conducted which are identified the defect identification and ranking for removal of risk affect in sprint and associated attributes was verified for the various module which perform on scrum functionality of the application. The various characteristics of the 3-Tier architecture of HA-ARRA Model. The modified scrum process based the 3-Tier architecture of HA-ARRA Model is given as Figure 5.2.

(Source: http://agileforall.com/resources/introduction-to-agile)

Figure 5.2 Modified scrum process based on the 3-Tier architecture of HA-ARRA Model
- **HA-ARRA Model** is appropriate to the agile methodology oriented process which helps to identify the defect ad risk factors of the various sprint of the scrum process and defect removal task was in-corrupted through various test case and testing process carryout in this model and it is suitable for other environment for fast development software of agile methodology.

- **HA-ARRA Model**, supports to reduce the risk defect I source code such as bugs, error, failure, faults etc., and it minimize through monitoring and report in the documentation for further clarification as reference, and mitigation through the risk ranking process to identification of their impact factors affect by the sprint in scrum process. And, the system efficient for agile process for quick delivery of product and transferable for understanding to end user by using this model architecture.

- **HA-ARRA Model** focuses on the defect removal with help of risk raking and reducing the risk factor for sprint and it is transferable of the whole process of the scrum process oriented project.

- **HA-ARRA Model**, to manages the both the sprint oriented and stories board structure oriented systems are converting them into an agile testing methodology and design are in corrupted into scrum cycle that allows to easily identification defect and risk affected sprint in the scrum master by their daily standup meeting, review by testing team and attributes of the sprint task release, thereby easily developed the product with an efficient product that satisfies customer requirements through this model.
• **HA-ARRA Model**, to finds out all possible risk defect removal factors in the agile methodology of scrum process by test case, enervative and destructive testing of the module of the sprint. By following this HA-ARRA Model, testing team will get reduced which makes the software product delivered to the customer requirement with timely manner.

• **HA-ARRA Model** uses iterative process for reduce the risk factors, which will more effectively to reduce the risk defect removal in the sprint of the scrum process and virtually reduce zero level of the risk factors affect in the scrum process.

### 5.5 COMPARISON WITH EXISTING WORKING MODELS

The following table made agile principles comparisons with Existing working model.

Table 5.2 Agile principles comparisons with Existing working model

<table>
<thead>
<tr>
<th>S. No</th>
<th>Name of the model, author and year of Published.</th>
<th>Working principle of model</th>
<th>Technology Used</th>
<th>Defect and Risk Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. No</td>
<td>Name of the model, author and year of Published.</td>
<td>Working principle of model</td>
<td>Technology Used</td>
<td>Defect and Risk Exposure</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
<td>-----------------------------</td>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>2.</td>
<td>Rank Based assessment and development risk assessment Model, Sahaul Hameed 2016</td>
<td>SPRE (Scrutinize, Prioritization and risk rank based extraction based framework. The SRCR (Software reusable component repository) was scored in the Questionnaire based Risk assessment for Normal distribution in risk assessment of SPRE.</td>
<td>Questionnaire based Risk assessment for Normal distribution in risk assessment of SPRE.</td>
<td>Overall identification of defect risk factors in software and made the risk assessment approaches to reduce the defect.</td>
</tr>
<tr>
<td>3.</td>
<td>Prince2 and scrum frameworks for reduce the project risk management model, Martin Tomanek and Jan Juricek 2015</td>
<td>Integration risk framework of Scrum process and PRINCE2 model with focus on risk assessment management.</td>
<td>Develop the tools for risk register for capturing defect and maintaining information of identified the threads.</td>
<td>Agile -Scrum risk mitigation and management process include to improving to manage project risk uncertainties on the product at the time of delivery l</td>
</tr>
<tr>
<td>4.</td>
<td>n-Tiered Test Automation Architecture for Agile software System, Patrick Day 2014</td>
<td>Distinct n-tiers (4 tiers level) are developed, risk business analytics, data and services to analysis the depth of testing case.</td>
<td>Automation is the act of converting manual test cases into automated script that can be executed autonomously.</td>
<td>Hybrid test case automation framework to evolve risk management for multiple project, and cover broader range of risk occur and significantly reduce the defect to increasing the Return On Investment (ROI) by using this architecture.</td>
</tr>
<tr>
<td>5.</td>
<td>A Risk-Driven Approach ad a develop the architecting for Large Scale Agile Software, Ipek Ozkaya 2013.</td>
<td>Feedback system on every stage of agile team and architecture-centric risk factors for adoption of agile by using of this two type.</td>
<td>Schedule and time was developed to conduct the evaluation approach focused for priority and severity of the risk scenarios and reduce the thread steps identified.</td>
<td>Architecture design to provides the visibility of project status and reduce the tactics for risk management and mitigation process. The architecture to evaluations to agile development sprints ad iterations process to assists in mitigating of the risk.</td>
</tr>
</tbody>
</table>
5.6 BENCH MARK REPORT FOR (HA-ARRA) MODEL

The bench values are HA-ARRA Model approximate maximum quantities of sprint using C++ language or JAVA packages. The values are verified through social survey with administrated on software industries organizations. The design values for HA-ARRA Model are presented below.

5.6.1 Benchmark Values for C++ Based Oriented Scrum Projects

The benchmark values were derived from social survey from scrum develop team of 11 software industries that execute the scrum projects based on C++ oriented environment. The average of the benchmark values out is approximated is provided in Table 5.3.

Table 5.3 Benchmark values for C++ oriented language

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Sub tasks(sprint)</th>
<th>Secondary sub tasks(sprint)</th>
<th>C++ language (application of sprint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>17%</td>
<td>36%</td>
<td>58%</td>
</tr>
</tbody>
</table>

The Table 5.3 are maximum percentage of agile methodology principles values that are used for considering scrum process development. The minimum value for every sub task (sprint) is 17% for defect removal in the sprint task.

5.6.2 Benchmark Value for JAVA Based Oriented Scrum Project

The benchmark of defect removal is design and to procedure for JAVA based scrum oriented environment is similarly carried out using social
survey from software industries where from 12 various opinion fro agile testing team as shown in Table 5.4

Table 5.4 Benchmark values for JAVA oriented language

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Sub tasks(sprint)</th>
<th>Secondary sub tasks(sprint)</th>
<th>JAVA language (application of sprint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40%</td>
<td>22%</td>
<td>22%</td>
<td>52%</td>
</tr>
</tbody>
</table>

The minimum value for every sub task (sprint) is 22% for defect removal in the sprint process task.

5.7 CASE STUDY OF “MYROUTES” APP USING SCRUM WITH HYBRID ARCHITECTURE OF AGILE RISK REMOVAL ANALYSIS (HA-ARRA) MODEL

Agile software development process is commonly used to iteratively and increment cycle to develop the information systems based on the customer requirement. Scrum methods are one of the most popular agile software development frameworks in agile technology. The foremost benefit of the Scrum framework is the iterative development approach and the opportunity to test in scrum process. In many android applications there are number of several map routing is available such as Google map, HERE map and so on. Based on the HA-ARRA model, the “myRoutes” is an application to developed in this chapter to find out the route, streets, Hotels, etc., this information in Google Maps may have errors report to the customer. For avoid this error product to follow this HA-ARRA model in this chapter.

Occasionally, ambiguities and flaws in location data may produce a route that doesn’t take you to the destination customer expect. Google Maps
does not have up-to-the-minute information on usual or unusual conditions, such as roads or maps damaged by weather, blocked by street fairs or altered by recent construction work through agile methodologies. Some remote locations may not be in Google Maps due to the error on the planning in the requirement.

5.7.1 Development of “myRoutes” App Using Scrum

The information gathered from the Google Maps that provides the including Street View images, material from the public etc., The Street View camera that captures images from criminal behavior and it passes along country roads and send to neighborhoods network. The disturbing images, off-color and racist to public comments and it take the references have appeared on the maps themselves.

The people decided to own route form source to destination path of the travelling with help of apps. The “myRoutes” app ensures that end-user can stay connected of the apps, plan to travel and ride decided routes effortlessly and enjoy with mobile apps. In this chapter “myRoutes” app was developed can display all various kind of information about the contents of requirement of the customer and tracks the route of the travel that much faster in fraction of seconds. The navigation shows the accurate point and it make the user-friendly interface and to secure the all kind of problems of possibility error occurs even the floods and road damages etc.

The “myRoutes” app can show the place of interest, create the new routes, analyse of the possibility routes, add friends list, chats with friends and manage the favourite places to visit. “myRoutes” app to munch through the Google map services to provide custom alerts based on my regular route of travel. The app would be developed with a smaller agile team of 6 members using Agile Scrum methodology as a sample.
5.7.2 MODULE DESCRIPTION DEVELOPMENT OF “MYROUTES” APP USING SCRUM WITH HA-ARRA MODEL

Module description has been established the “myRoutes” app using Scrum with HA-ARRA Model, and it can be classified into four modules would be four Epics (Modules) and each one Epics can be split into smaller stories. Each story can be split into six sprints and estimated time is 13-15 days of each sprints. Each sprint would be of two weeks’ timeline for next iterative.

After the successful implementation of apps, and it would be placed in Google Store. Scrum is the one of the tool which is installing for the testing purpose itself. This analysis consists of four different modules of the “myRoute” and the main object to get information and data from Google Map services and it provide alerts to the end-user for parameters set as given below.

Module 1: User Interface (UI) for the mobile app. Created using Android Studio.

Module 2: Creating the Service to get information and map details from Google map services.

Module 3: To integrate the services and UI and create a data store to maintain the Route information.

Module 4: Backend service to send the alert to the UI based on the information requested.
5.7.3 Module 1: User Interface for the mobile app. Created using Android Studio

The sample “myRoutes” app used in the Mobile Shopping Assistant for customer requirement. The “myRoutes” app lets to the end-users get information on products. The “myRoutes” app includes custom code of the product to cable the basic app client to the App Engine cloud be backend, and using Google Cloud Endpoints. The app can then determine the user location point, locate nearby customer care, and allow to obtain relevant special offers and recommendations. The overview of structure of “myRoutes” is given in the Figure 5.3.

**Figure 5.3 Mobile App Engine**

**App Engine Backend:** The App Engine to describe the set up the mobile backend app which runs on Google Cloud Platform (App Engine). The application “myRoutes” contains the data repository in the server and the business logic and customer requirement to support the mobile client requests has the following advantages to end-user: Android Studio to set-up the backend for both clients support. The metrics has given below.
• Android Studio access direct integration with App Engine “myRoute”.

• Through Android Studio, the auto-generate the Cloud Endpoints client libraries needed to access the backend for them.

**Step 1: Create the backend project (Sprint)**

• To create the back-end project in Android Studio, and follow the instructions to Add a Cloud Messaging Module.

• The backend project in Android Studio package name can include as (com.google.sample.mobile) to assistant of the backend.

**Step 2: Write the backend (Sprint)**

To ready to start building the “myRoutes” application to creating the entity classes that represent the app's backend data of the customer requirement. An entity class has to provides an “object-relational interface” between “myRoutes” application and the backend data repository of the information.

The backend (Module:1) in the Mobile Shopping Assistant sample “myRoutes” app includes the following packages with the following contents:

• com.google.sample.mobile assistant to backend. Models : The package includes the entity files and data repository that map to persistent data objects to given requirement.

• com. google. sample. mobile assistant –back end. apis : Apps cohtains the End-point files that expose the (REST APIs) for given resources as (Check-In, Place of visit, special offer,
Recommendation, and Message - Google Cloud Messaging
messages between the “myRoutes” app and a registered device).

**Step 3: Create an entity class (Sprint)**

The sample “myRoutes” in this section are based on the Check-In entity. To create an entity class for each of the entities in app. To create an entity class:

1. In the Package Explorer, navigate to the models folder under backend project, and select File > New > Java Class. The New Java Class wizard is displayed in the menu bar.
2. In the Name: box enter the name of the class that will represent the entity, for example, Check-In.
3. Enter the Click button for ‘OK’.
4. In the section of sprint, to replace the boiler plate code with the code for the Check-In.java class that obtains from the downloaded sample. And the source code shown provides a “ready to run” class. The most important to note that pay attention to are the properties and the annotations of the class.
5. Save the “class file”.
6. End point of the Check-In entity.

**Step 4: Generate a Cloud Endpoints Class (Sprint)**

Each entity class, to need to generate a “Cloud Endpoints class” to give the client app access to backend data base. To generate a Cloud Endpoints class is give below.
• Create a class Check-In and Endpoint in the backend/apis package of the modules.

• To open the screen /main menu /web-app/web-inf/web.xml and add the new class full path com. google. .mobile to assistant back end.apis. Check-In and Endpoint in the <parameter-value> section, as sample coding follows in Figure 5.4:

```
< servlet>
   <servlet-name>SystemServiceServlet</servlet-name>
   <servlet-class>com.google.api.server.spiSystemServiceServlet</servlet-class>
   <init-param>
     <param-name>services</param-name>
     <param-value>
       com.google.sample.mobileassistantbackend.apis.CheckInEndpoint,
     </param-value>
   </init-param>
</servlet>
```

**Figure 5.4 Sample coding of Cloud Endpoints Class (Sprint) using Java**

The open the Objectify service class of “Service.java” files. In the static bit, to add the line Objectify Service. register(Check-In.class) to register the Check-In class as persistent I this sprint of the scrum process.

**Step 5: Create Android client (Sprint)**

The mobile client “myRoutes” application that provides the interface of the customer requirement that allows the end-user to interact with the backend, for sample to obtain sales information about the apps. To create a mobile client app as given below.
1. In Android Studio, to select File > New Project of the apps.

2. The New Project wizard window is displayed in the main menu.

3. In the Application “myRoutes” as Name box, enter the name of mobile application, for example as a Mobile-Assistant.

4. In the Package of the module as Name box, enter the name of application’s package, for example (com. google. sample solutions. mobile assistant).

5. Target Android Devices dialog, and to accept the default values for form factor analysis and minimum SDK, and click Next button.

6. In the Add an activity to Mobile dialog, accept the "blank activity" default selection, and click Next.

7. To customize the activity dialog box, and make any desired edits of the activity, and click Finish. Next go to end process.

8. End of Create Android client.

5.7.4 Module 2: Creating the Service to get Information and Map Details from Google map services (Sprint)

To Creating the Service to get information and map details from Google map services(Sprint). The source code was developed as shown in the Figure 5.5.
5.7.5 Module 3: To Integrate the Services and UI and Create A Data Store to Maintain the Route Information

An external data service provides the implemented of the source code for the following adding documents and folders/file are shown in below.

- Checking the documents in to the data base repository
- Editing the properties for a selection or choose the multiple items from main menu.
- Editing item properties in the viewer to end-user.
- Using entry the data base into templates already created.
- Setting process of the workflow filter in a criteria fields
- Creating or using searches for requirement of the end-user.
- Controlling and monitoring the Content Manager folder.
The external data service to customize the user satisfaction to working environment as followed the field properties and property behaviors of the content activities.

**Step 1: Database to create choice lists or options list (Sprint)**

- Create choice lists by using existing database that has to managed in content repository about the information, and to connected with external source.
- For example, the values in a file that is located point and managed in an external server or data repository.

**Step 2: Prefill properties (Sprint)**

- Specify prefilled properties and default values in the main menu.
- For example, prefill fields with custom default values and it is based on a particular class IDs, authenticated user IDs, or the parent folder to be entered in a form.

**Step 3: Specify property dependencies (Sprint)**

- The main dependencies between properties local region or global region. For example, the dependency or match between a geographic region and an office branch choice list property. The geographic region, the subsequent choice list is match with the selected geographic region.
Step 4: Set minimum and maximum values (Sprint)

- An integer, float, or date are defining the maximum the value or minimum value for an attributes property.

- Restriction: The minimum or maximum value to be less restrictive when compared with the data repository.

Step 5: Set read-only status (Sprint)

- Select the property to be read-only menu. For example, it might create a property setup that requires a particular value from the user.

- To specify the correct default value and make that property read-only.

Step 6: Set required status of field (Sprint)

- Set a property to be a required field value and the attribute on a property value can be appears in the user interface to indicate that the field is required end-user.

- End-user cannot proceed from the each page or dialog box

Step 7: Set hidden status (Sprint)

- Hide a property from the user interface are shown in the dialog box.

- For example, to create a choice or option list that dynamically determines subsequent text input fields to present in a data form.

- To hide a property that does not show hidden attribute in dialog box.
5.7.6 Module 4: Backend Service to Send the Alert to the UI based on the Information Requested

Step 1: Working with ID-Values (Sprint)

Mobile Services which has supports the unique IDs to custom string values for the table's matrix IDs column. “myRoute” to allows the applications to use custom values like email-addresses, user-names for the IDs. For example, the following source code inserts a new item of object, where the unique IDs with an email address. String IDs have a unique benefit to generated without making a round-trip to the database entry, to merge from different tables or databases, IDs values can be integrating better solution with an application's logic. When a string ID-Value is not already set on an inserted record, Mobile Services generates a new unique value for the ID.

Step 2: Modify data in a mobile service(Sprint)

The modify data information of mobile services can be create as shown in Figure 5.6

```javascript
To do Item Table. update
({
  id: id to Update,
  text: new Text
}). done (function(result) {
  Alert(JSON.stringify(result));
}, function(err) {
  Alert("Error: " + err);
});
```

Figure 5.6 Source code for Modify data of mobile services
Step 3: Delete data in a mobile service (Sprint)

The following source code illustrates to delete data from a table matrix. The client requests for ‘DELETE’ option to the mobile service. And the deletes an item source code can be shown in the figure 5.6.

```javascript
To do Item Table.del(
{id: id to Delete
}).done(function() {
    /* Do something */
}, function(err) {
    Alert("Error: " + err);
});
```

Figure 5.6 Source code for Deleted option of the Mobile Services

Step 4: Display data in the user interface(Sprint)

The display data in the user interface for delete option from the main menu, the following source for delete option as shown in the Figure 5.7.

```javascript
{
    For (var i = 0; i < to do Items.length; i++) {
        var li = document.Create Element(‘li’);
        var div = document.Create Element(‘div’);
        div.innerText = to do Items[i].text;
        li.appendChild(div);
        list of Items.AppendChild(li);
    }
    Read().done(function(results) {
        Alert(JSON.stringify(results));
    }, function(err) {
        Alert("Error: "+ err);
    });
```

Figure 5.7 Source code for user interface for delete option using java
5.8 SUMMARY AND DISCUSSION

The “myRoutes” app designed by using scrum process. The purpose of this tool and design is to testing whether the navigate information are precise or not in software application using of scrum process. “myRoute” app displays all kind of dissimilar information and contents and it tracks the route much faster in fraction of seconds to customer or end-user. The navigation is to be accurate to use and it is user-friendly interface and secure to use and it shows all kind of problems on way even the floods and road damages so on. The planning of “myRoute” app development using Google map services are important process and it should be done in an efficient manner by using the HA-ARRA Model.