**Year 4**

2014

**IT Tallaght, Tallaght, Dublin 24**

08

**Fall**

Continuous Delivery ICS Adoption Dashboard

Shane Murphy

The purpose of the project was to develop a commit phase dashboard to monitor the adoption of a continuous delivery work flow in IBM’s ICS, with information fetched from RTC, RQM and Jenkins and stored locally. The users would be able to compare how each team progressed towards CD at a glance, aided by graphical representations of the data. Should more details be required, the ability to delve deeper into the teams to view builds and work items at a finer granularity is possible. Users were determined to be both team members and upper management.

Table of Contents

Background 5

Project 5

Overview 5

CD Transformation Checkpoints and Metrics 6

CD Transformation Metrics 6

Commit Phase 6

Acceptance Test Phase 6

Manual Test Phase 6

CD Business Value 7

Technological Research 7

Jenkins 7

RTC 7

Continuous Delivery Dashboards 8

Bower 10

Jasmine 10

QUnit 10

Webstorm IDE 10

MongoDb 10

DB2 11

LESS 11

COFFEE 11

Twitter Bootstrap 11

Groovy & Grails 11

Use Cases 12

View Summary Statistics 12

Order Results 12

Inspect Team Progress in finer Granularity 13

Historical Analysis 13

Dashboard Refresh 14

Team Progress vs Goals 14

Technical Architecture 15

Software Components 15

Groovy 15

Grails 15

DB2 16

Platform libraries 16

Distribution and Deployment 17

Architecture Diagram 18

Class Diagram 19

Database Data Dictionary 20

build 20

Indexes 20

build\_test\_metrics 20

Indexes 20

contributor 21

Indexes 21

registration\_code 21

Indexes 21

role 21

Indexes 21

team 21

Indexes 22

team\_contributor 22

Indexes 22

user 22

Indexes 22

user\_profile 23

Indexes 23

user\_profile\_projects 23

user\_role 23

Indexes 23

work\_item 23

Indexes 24

Database Structure 24

Test Cases 25

Test Results 33

Future Extensions 34

User Manual 35

Installation 35

Application Manual 36

Home Screen 36

Home Screen Logged In 37

Login 37

Team Info 38

Team Info No Data 39

Build Info 39

WorkItem Info 40

About 40

User Profile Edit 41

User Profile 41

Footer 41

Post Project Review 43

A review of the project in its entirety 43

Possible extensions of the project 43

Review of Research, Analysis and Design phases 43

Review of Implementation phase 44

Progress Diary 45

Week Beginning 14/10/13 45

Week Beginning 21/10/13 45

Week Beginning 28/10/13 46

Week Beginning 6/11/12 47

Week Beginning 13/11/13 47

Week Beginning 19/11/13 49

Week Beginning 27/11/13 49

Week Beginning 27/11/13 50

Week Beginning 4/12/13 52

Week Beginning 11/12/13 53

Week Beginning 18/12/13 53

Week Beginning 25/12/13 53

Week Beginning 22/01/13 53

Week Beginning 29/01/14 54

Week Beginning 05/02/14 56

Week Beginning 12/02/14 56

Week Beginning 19/2/14 56

Week Beginning 26/02/14 58

Week Beginning 05/03/14 59

Week Beginning 19/3/14 60

Week Beginning 27/3/14 61

Week Beginning 2/4/14 61

Week Beginning 9/4/14 62

Week Beginning 16/4/14 63

Week Beginning 23/4/14 64

Week beginning 30/4/14 64

Bibliography 65

# 

# Background

Continuous Delivery (CD) is a pattern language used in software development to automate and improve the process of software delivery. Techniques such as automated testing, continuous integration and continuous deployment allow software to be developed to a high standard and easily packaged and deployed to test environments, resulting in the ability to rapidly, reliably and repeatedly push out enhancements and bug fixes to customers at low risk and with minimal manual overhead.

IBM Collaboration Solutions (ICS), which consists of a large range of software development teams, comprising OnPrem, Cloud and Mobile, have made the strategic decision to adopt Continuous Delivery Practices for all software development teams. Each ICS Director is required to report progress on a set of CD Transformation Checkpoints and Metrics to Executives on a regular and scheduled basis.

# Project

Develop a Continuous Delivery ICS Adoption Dashboard, which reports out each CD Transformation and on a per software product development team basis. The data must be fed in from verifiable sources to ensure confidence in the status presented by the dashboard. It cannot be a tick box exercise.

For the purposes of the project, research can be carried out on the source and format of the data feed and how that data is presented to the dashboard. Sample data can be used, but the dashboard must be extensible to include additional metrics once known. The dashboard must be capable of being easily merged with any future common metrics repository and dashboard.

The main purpose of the dashboard is to provide information to team

members, management and executives on how they are doing against defined

goals.

# Overview

The purpose of the project was to develop a commit phase dashboard to monitor the adoption of a continuous delivery work flow in IBM, with information fetched from RTC, RQM and Jenkins and stored locally. The users would be able to compare how each team progressed towards CD at a glance, aided by graphical representations of the data. Should more details be required, the ability to delve deeper into the teams to view builds and work items at a finer granularity is possible. The design was kept general, as users could be both team members and upper management. The ability to log in and customize which teams were of interest to you allows unnecessary information to be filtered out. The project was implemented in Groovy and Grails using an MVC architecture sitting above a services layer. I used a combination of d3.js and the Google Charts API to implement the graph elements, Spring Security to provide user logins and profiles and Quartz2 to execute cron jobs.

# CD Transformation Checkpoints and Metrics

CD Transformation Checkpoints

Transformation Checkpoint

* Each code submission triggers the pipeline in serial
* Jenkins provides CI pipeline
* CI pipeline is fully automated
* Single source repository/trunk
* All artifacts kept in version control
* All Commit Phase failures stop the pipeline
* SPRs add data to track test escapes

CI pipeline is implemented

* Leverage common metrics repository and
* dashboard
* Push-button automated pre-submission validation
* All new code is code reviewed
* All new code has unit and functional test coverage
* Automated tests added for all resolved SPRs

Developer role and enablement

* SPRs created for all issues outside of the Commit Phase
* All test deployments use production deployment processes and are fully
* automated/repeatable
* All test deployments use production-congruent environments/stacks
* Full regression/compatibility testing of APIs and extensions
* Simian Army testing occurs continuously
* Dashboard metrics gathering is automated
* Continuous on-premises testing (NFR and on-prem only features)
* Zero downtime updates

Continuous delivery

* Additional SCSB usage/defect metrics stored in repository

## CD Transformation Metrics

### Commit Phase

|  |  |  |
| --- | --- | --- |
| Metric | Measurement Value | Anticipated Source |
| Build Duration | Minutes | RTC |
| Manual Testing Duration (Automatable) | Person Weeks | RQM |

### Acceptance Test Phase

|  |  |  |
| --- | --- | --- |
| Metric | Measurement Value | Anticipated Source |
| API Test Coverage (via automated measurement) | % Value | TBD |

### Manual Test Phase

|  |  |  |
| --- | --- | --- |
| Metric | Measurement Value | Anticipated Source |
| Manual Testing duration (non-automatable) | Person Weeks | RQM |
| Time From Final Code submission to Production (Cloud) | Days | DevOps / UrbanDeploy / CHEF / Jenkins |
| Time from final code submission to GA (On-Premises) | Days | DevOps / UrbanDeploy / CHEF / Jenkins |
| SPRs opened per capita (general quality measure) | Number | RTC |

### CD Business Value

|  |  |  |
| --- | --- | --- |
| Metric | Measurement Value | Anticipated Source |
| Cumulative Open & Deferred Defects (Technical Debt) | Number | RTC |

# Technological Research

While deciding on how I would implement the project, I spent some time researching and exploring various technologies before deciding on my final technology stack.

## Jenkins

* Jenkins plugins
* REST API resulting in XML/JSON/Python
* Internal Data model stored as a tree, API calls return a subtree, subtree cuts off at a certain point, this point can be adjusted.

## RTC

Setting up RTC for development (Schoon, 2013)

What API's are available? (Schoon, 2013)

RTC API?

* Javascript (limited)
* Java
* REST

Javadoc (IBM, n.d.)

* Planning
  + Documentation
  + Estimates
  + Widgets/Dashboard??
  + Change Requests
    - Delegation
  + Work Items
  + Build
    - Test
    - History
  + SCM
    - Role Based (privileges useable in dashboard?)
* Streams
  + Project areas
  + Team areas
* Components
  + User areas
  + Project areas
  + Team areas

## Continuous Delivery Dashboards

-Continuous Delivery in the Cloud – Part 6: Create a Dashboard of your System

(Codecentric, n.d.)

Python

Integration with Jenkins

* Query the test-results of a completed build
* Get objects representing the latest builds of a job
* Search for artifacts by simple criteria
* Block until jobs are complete
* Install artifacts to custom-specified directory structures
* username/password auth support for jenkins instances with auth turned on
* Ability to search for builds by subversion revision
* Ability to add/remove/query jenkins slaves

(Python, n.d.)

Ruby

Integration with Jenkins

* Creating jobs by sending xml file or by specifying params as options with more customization options including source control, notifications, etc.
* Building jobs (with params), stopping builds, querying details of recent builds, obtaining build params, etc.
* Listing jobs available in Jenkins with job name filter, job status filter.
* Adding/removing downstream projects.
* Chaining jobs i.e given a list of projects each project is added as a downstream project to the previous one.
* Obtaining progressive console output.
* Username/password based authentication.
* Command Line Interface with a lot of options provided in the libraries.--Creating, listing views.
* Adding jobs to views and removing jobs from views.
* Adding/removing jenkins slaves, querying details of slaves.
* Obtaining the tasks in build queue, and their age, cause, reason, ETA, ID, params and much more.
* Quiet down, cancel quiet down, safe restart, force restart, and wait till Jenkins becomes available after a restart.

(RubyGems, n.d.)

Knockout.js

Data-binding library

* Bind variables in the View Model (DOM)
* Observable Arrays
* Model View View Model (MVVM) Pattern
* shallower learning curve when compared to angular/backbone
* Becomes complicated when you want to do things that are not supported out of the box
* No IDE plugin

(Knockout, n.d.)

Angular.js

* Full Framework
* Useful for Data Driven applications, nice data binding
* JQuery or JQueryLite dependant
* Offers similar data binding to knockout, but also controllers and sets of services
* Dependancy injection is used quite a lot
  + Name function arguments with the same name
* Steep learning curve for more advanced features
* Popular
* Supports automatic testing out of the box
* IntelliJ plugin/VisualStudio plugin
  + Angular Batarang plugin for Google Chrome developer tools
* Large support community

(AngularJS, n.d.)

Backbone.js

* Useful for heavy DOM manipulation (With JQuery)
* Key-value binding and custom events
  + Not Data binding
  + Build own MVC solution using backbone objects
* Faster than Angular/Knockout (http://jsperf.com/angular-vs-knockout-vs-ember/171)
* A lot of boilerplate (code for wiring things together)
  + Free with Angular/Knockout
  + Code easier to read (less 'magic')
  + No surprises
* Need understanding of how Backbones set of objects work together
* Useful for specialised applications

(Backbone, n.d.)

Node.js

Networking

Push updates from the server to the client without having to poll or refresh?

Polling?

(Node, n.d.)

d3.js

Data visualization

(D3, n.d.)

Grunt.js

* Task runner
* Automate repetitive tasks
* Large JS projects
* Preprocessor files

(Grunt, n.d.)

Karma runner

* Test Runner
* Spawns a web server and executes source and test code for each browser connected
  + Results displayed via command line

(Karma , n.d.)

## Bower

* package manager
* Store a bower file, can pull down dependencies automatically.

(Bower, n.d.)

## Jasmine

* Test Framework for Javascript
* Behaviour Driven
* Clear syntax
* IBM Preference

(Jasmine, n.d.)

## QUnit

* IBM Prefer Jasmine, don’t use this

(QUnit, n.d.)

## Webstorm IDE

* Intellisense
* JetBrains
* Git integration
* Educational Licence available
  + Got License off Enda

(Jetbrains, n.d.)

## MongoDb

* Store Results of Jenkins queries for analysis of team improvements etc.?
* Document-orientated Database
* NoSQL
* MongoDb not approved for use in IBM, Brendan suggested DB2 instead

(MongoDb, n.d.)

## DB2

* Support for XML storage

(IBM, n.d.)

## LESS

CSS preprocessing

(LESS, n.d.)

## COFFEE

Javascript preprocessing

(Coffee, n.d.)

## Twitter Bootstrap

For anything not provided by IBM CSS

(Twitter, n.d.)

RQM  
(Jazz.net, n.d.)

## Groovy & Grails

* Compiles to Java bytecode on JVM
* Use of Closures, like C# or Java 8
* Similar to popular framework Ruby on Rails

(Grails, n.d.)

# 

# Use Cases

|  |
| --- |
| View Summary Statistics |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product Team Member | | **Secondary Actor(s)** | Product team manager  Executive/Director | | **Level** | 1 | | **Story** | As a Product team member, I want to be able to view continuous delivery transformation assessments against agreed goals for my team. | |

|  |
| --- |
| Order Results |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product team manager  Executive/Director | | **Secondary Actor(s)** | Product Team Member | | **Level** | 2 | | **Story** | As a Manager/Director, I want to order results by General transformation statistics (column) to better allow me to see how teams are progressing with their CD adoption in relation to other teams. This would better facilitate the organisation of cross team knowledge sharing. | |

|  |
| --- |
| Inspect Team Progress in finer Granularity |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product team manager | | **Secondary Actor(s)** | Executive/Director | | **Level** |  | | **Story** | The ability to view the general statistics in finer detail to determine areas where the team could improve their performance in relation to CD adoption. The most relevant information should be readily visible from a high level, with the option to delve deeper and view additional information and more detail where available. Graphing, where appropriate, would make this information more useful. | |

|  |
| --- |
| Historical Analysis |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product team manager  Executive/Director | | **Secondary Actor(s)** | Product Team Member | | **Level** |  | | **Story** | As a Product Team Manager/Director, I want to view on a team-by-team basis, how the team is progressing with its adoption of continuous delivery. This would be represented best as graphs of the summary statistics over time. | |

|  |
| --- |
| Dashboard Refresh |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product Team Member | | **Secondary Actor(s)** | Product team manager  Executive/Director | | **Level** |  | | **Story** | As a general user, I want to see up to date feed information without having to periodically manually refresh the page. An automatic refresh time of ~60mins or a user definable time period would be increase usability. | |

|  |
| --- |
| Team Progress vs Goals |
|  |
| |  |  | | --- | --- | | **Primary Actor** | Product team member | | **Secondary Actor(s)** | Product team manager | | **Level** |  | | **Story** | As a product team member, I would like to be alerted to how our progress with CD adoption compares with our goals and expectations, highlighting areas where we are falling behind. | |

# Technical Architecture

## Software Components

### Groovy

The bulk of the application will be developed using the groovy language and will be built around the Grails framework. Groovy is dynamic statically typed language that runs on the JVM. Is designed to extend the Java language and make programming clearer and less cluttered.

There is a very shallow learning curve getting to grips with groovy when coming from a Java background, infact most Java classes can be run as a Groovy class, and Java can be written directly into a groovy class.

### Grails

Grails is a Framework built on Groovy. It has a Plugin based architecture making it very flexible and extensible. As a default, the following technology stack is used out of the box.

The default database is HSQL run in-memory. However, I will swap this out for a DB2 database to comply with IBM standards. This should be a simple process as Grails stack includes Hibernate out of the box.

#### Spring MVC

It implements Spring MVC application web framework (Though Grails is now capable of more than web developement). This framework seperates the presentation layer from the request handling layer, and the request handling layer from the model. By doing so, Spring MVC allows for cleaner front end code and easier testing and maintainablility.

#### Hibernate

Hibernate provides Object Relational Mapping for Java. Using GORM (Groovy Object Relational Mapping) and POGO (Plain Old Groovy Objects - like POJO for Java) this also works for Groovy. It provides a layer of abstraction from the database used in the applications implementation, allowing databases to be changed easily after deployment, as no database commands are hard coded into the source code. It does this by replacing direct persistence-related database accesses with high-level object handling functions.

#### Sitemaesh View Templates

Sitemesh is used as a framework for providing user views of an application to any HTML page requests using a gang of four decorator pattern. It allows for partial views to be created and embedded inside other views, allowing code reuse and quick building of pages.

#### Javascript

Because of the plugin architecture, Javascript libraries can be plugged directly into the project. This will be particularly helpful in implementing the AJAX calls required to refresh the page content after some elapsed time period. Some libraries that might prove useful in this project are Prototype, Scriptaculous, Compass, Lucene and Dojo. These requirements will become more apparent over time and will be explored in later iterations as development progresses.

## DB2

IBM's DB2 is a database which can handle multiple types of database schemas (Relational, object-relational, Native XML storage, etc.). It supports both SQL and XQuery query languages for database manipulation, allowing for faster access to the natively stored XML using XQuery. I will be using DB2 through the Hibernate abstraction layer provided in Grails, however it also have a number of native APIs for various programming languages.

## Platform libraries

Build Duration - from RTC

Manual testing Duration - Test Case Executions Records

API Test Coverage - TBD - We don't know how API Test Coverage will be

tracked. RQM would represent a tick box exercise

time from final code submission to production (cloud and on-premises) -

Jenkins will drive the commit but UrbanDeploy or CHEF will do the

deployment. How long it takes to execute the pipeline. Jenkins.

Defects opened per capita - RTC Query for Defects

Cumulative open+deferred defects - RTC Query

The data sources for the project will primarily come from the following sources:

* Build Logs
  + RTC has Build Duration time stored, this should be easy to access.
* DevOps
  + To be advised.
* Jenkins
  + REST API available which returns information in XML or JSON format. Again, easily and simply handled by groovy. Using the inbuilt HTTPBuilder class, messages can easily be formatted correctly to be sent accross the network. As an extension to this class there is also a RESTClient class which makes concessions in HTTPBuilders flexibility to make REST operations as simple as possible.
* RTC
  + REST API provided to interact with IBM Jazz based tools
* RQM
  + Still to be researched.

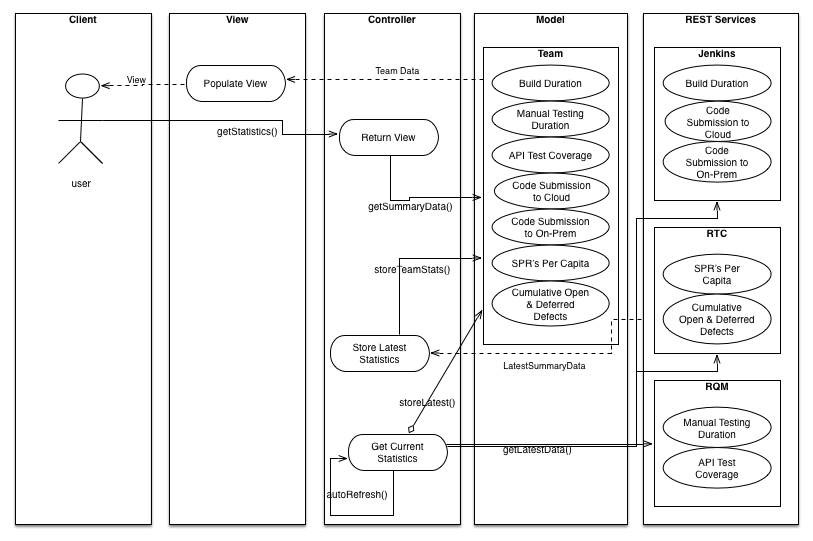
## Distribution and Deployment

* Can be deployed to a server as a WAR file.

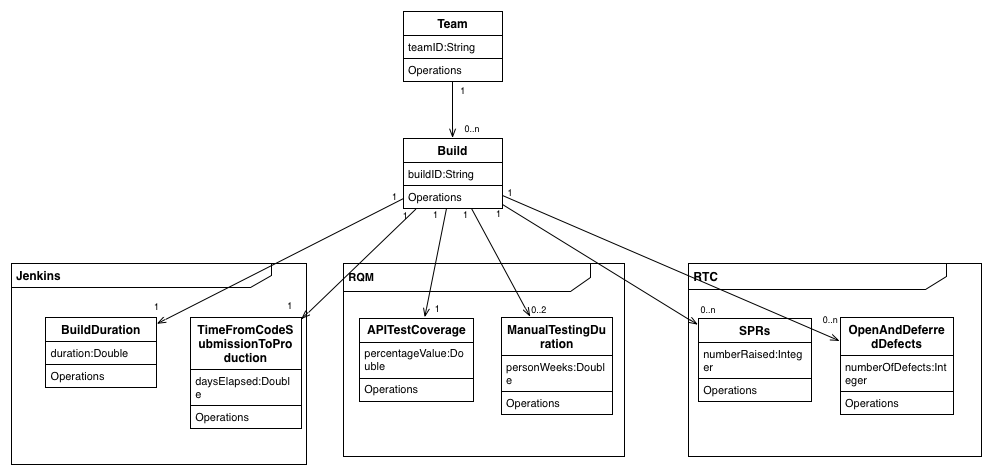
Risks

* Large scope of the project may need to be revised.
* Limited experience of javascript, which will likely be used to implement the graphs.
* A lot of firsts involved in the project
  + New Language (Groovy)
  + New Framework (Grails)
  + New Servers (RTC/DB2)

## Architecture Diagram



## Class Diagram



# 

# Database Data Dictionary

## build

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| build\_id | varchar(255) | No |  |  |
| version | bigint(20) | No |  |  |
| build\_definition\_id | varchar(255) | No |  |  |
| build\_state | varchar(255) | No |  |  |
| build\_status | varchar(255) | No |  |  |
| build\_time\_in\_millis | bigint(20) | No |  |  |
| modified | datetime | No |  |  |
| name | varchar(255) | Yes | N/A |  |
| start\_time | bigint(20) | No |  |  |
| team\_id | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | build\_id | 167 | A | No |  |
| build\_id | BTREE | Yes | No | build\_id | 167 | A | No |  |
| FK59BC66E9DA1F667 | BTREE | No | No | team\_id | 167 | A | No |  |

## build\_test\_metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| version | bigint(20) | No |  |  |
| build\_id | varchar(255) | No |  |  |
| commit\_phase\_testing\_time | bigint(20) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 150 | A | No |  |
| build\_id | BTREE | Yes | No | build\_id | 150 | A | No |  |
| FK3464004723CFEECD | BTREE | No | No | build\_id | 150 | A | No |  |

## 

## contributor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| version | bigint(20) | No |  |  |
| email | varchar(255) | No |  |  |
| name | varchar(255) | No |  |  |
| user\_id | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 53 | A | No |  |

## registration\_code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| date\_created | datetime | No |  |  |
| token | varchar(255) | No |  |  |
| username | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |

## role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| version | bigint(20) | No |  |  |
| authority | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 2 | A | No |  |
| authority | BTREE | Yes | No | authority | 2 | A | No |  |

## team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| team\_id | varchar(255) | No |  |  |
| version | bigint(20) | No |  |  |
| team\_name | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | team\_id | 17 | A | No |  |
| team\_id | BTREE | Yes | No | team\_id | 17 | A | No |  |

## team\_contributor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| team\_team\_members\_id | varchar(255) | Yes | *NULL* |  |
| contributor\_id | bigint(20) | Yes | *NULL* |  |
| team\_members\_idx | int(11) | Yes | *NULL* |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| FK36715DF9886DD16D | BTREE | No | No | contributor\_id | 53 | A | Yes |  |

## user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| version | bigint(20) | No |  |  |
| account\_expired | bit(1) | No |  |  |
| account\_locked | bit(1) | No |  |  |
| enabled | bit(1) | No |  |  |
| password | varchar(255) | No |  |  |
| password\_expired | bit(1) | No |  |  |
| user\_profile\_id | bigint(20) | No |  |  |
| username | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 2 | A | No |  |
| user\_profile\_id | BTREE | Yes | No | user\_profile\_id | 2 | A | No |  |
| username | BTREE | Yes | No | username | 2 | A | No |  |
| FK36EBCB2D10556 | BTREE | No | No | user\_profile\_id | 2 | A | No |  |

## 

## user\_profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| id | bigint(20) | No |  |  |
| version | bigint(20) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | id | 2 | A | No |  |

## user\_profile\_projects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| user\_profile\_id | bigint(20) | Yes | *NULL* |  |
| projects\_string | varchar(255) | Yes | *NULL* |  |
| projects\_idx | int(11) | Yes | *NULL* |  |

## user\_role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| role\_id | bigint(20) | No |  |  |
| user\_id | bigint(20) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | role\_id | 2 | A | No |  |
| user\_id | 2 | A | No |
| FK143BF46A457C93C7 | BTREE | No | No | role\_id | 2 | A | No |  |
| FK143BF46AEAA757A7 | BTREE | No | No | user\_id | 2 | A | No |  |

## work\_item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| work\_item\_id | varchar(255) | No |  |  |
| version | bigint(20) | No |  |  |
| build\_owner\_id | varchar(255) | No |  |  |
| creation\_date | datetime | No |  |  |
| duration | bigint(20) | No |  |  |
| modified | datetime | No |  |  |
| resolution\_date | datetime | No |  |  |
| severity | varchar(255) | Yes | *NULL* |  |
| type | varchar(255) | No |  |  |

## Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | work\_item\_id | 1535 | A | No |  |
| FK40FB2F81BF6E6719 | BTREE | No | No | build\_owner\_id | 1535 | A | No |  |

# Database Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Rows | Type | Size | Comments |
| build | 167 | InnoDB | 80 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| build\_test\_metrics | 150 | InnoDB | 48 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| contributor | 53 | InnoDB | 16 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| registration\_code | 0 | InnoDB | 16 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| role | 2 | InnoDB | 32 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| team | 17 | InnoDB | 32 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| team\_contributor | 53 | InnoDB | 32 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| user | 2 | InnoDB | 64 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| user\_profile | 2 | InnoDB | 16 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| user\_profile\_projects | 1 | InnoDB | 16 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| user\_role | 2 | InnoDB | 48 KiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| work\_item | 1,523 | InnoDB | 1.7 MiB | |  |  | | --- | --- | | Creation: | Apr 29, 2014 at 09:18 PM | |
| 12 tables | **1,972** | **--** | **2.1 MiB** |  |

[pen new phpMyAdmin window](http://localhost:8888/phpMyAdmin/db_printview.php?db=ibm&table=&server=1&target=&token=54fd19ebe888e774c9d2c7356e2d3d20)

# Test Cases

|  |  |
| --- | --- |
| Test Case ID | START1 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| Startup With Empty Database | Test expected behavior is carried out when application is started | Database is empty but database area is created | Database is populated with Random Mock Data from Bootstrap, and all RTC Information is pulled in. |

|  |  |
| --- | --- |
| Test Case ID | START2 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| Start with existing data in database | Test expected behavior is carried out when application is started |  | No additional RTC or Bootstrap data is added to the database |

|  |  |
| --- | --- |
| Test Case ID | START3 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| Start with no connection to RTC Server | Tests that the application is stable even if RTC connection lost. |  | Exception caught and logged to error log. Application continues to run. |

|  |  |
| --- | --- |
| Test Case ID | START4 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| Fail if started with no Server Backend | The application requires a server backend in order to start. |  | Fail to start |

|  |  |
| --- | --- |
| Test Case ID | NETWORK1 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| No Internet Connection | Tests that the application can operate without a connection to the internet. |  | Application should function correctly, but graphs on the main page will not display. Google Chart API requires internet access. |

|  |  |
| --- | --- |
| Test Case ID | LOGIN1 |
| Test Case Name | Purpose | Initiation Criteria | Expected Results |
| Login | Tests the ability to login as default users admin and user with password = password |  | User logged in |

|  |  |
| --- | --- |
| Test Case ID | LOGOUT1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Logout | Tests the ability to Log out from a user account | User is logged in | 1. Select Logout from the black bar at the top right of the screen | User logged in |

|  |  |
| --- | --- |
| Test Case ID | USERPROFILE1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Add Subscriptions | Tests the ability to add subscribed teams to a users account | 1. User is logged in. | 1. Select User Profile from the black bar at the top right of the screen 2. Click ‘Edit’ 3. Select at least one Team from the checkboxes | Subscriptions added to account |

|  |  |
| --- | --- |
| Test Case ID | USERPROFILE2 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Remove Subscriptions | Tests the ability to remove subscribed teams from a users account | 1. User is logged in. 2. User has subscriptions associated with their account | 1. Select User Profile from the black bar at the top right of the screen 2. Click ‘Edit’ 3. Deselect at least one Team from the checkboxes | Subscriptions updated to reflect this change, team no longer visible. |

|  |  |
| --- | --- |
| Test Case ID | USERPROFILE3 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| View Profile | Tests the ability to view currently subscribed teams | 1. User is logged in. 2. User has subscriptions associated with their account | 1. Select User Profile from the black bar at the top right of the screen | Subscriptions shown as a list |

|  |  |
| --- | --- |
| Test Case ID | USERPROFILE4 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Can’t view other user’s profile | Tests that each users profile is secure from tampering by a 3rd party. | 1. User is logged in. 2. More than 1 user is registered | 1. Select User Profile from the black bar at the top right of the screen 2. Change the URL to the next User ID. | Changing the URL doesn’t load a new page. Only current users profile is loaded, regardless of the URL |

|  |  |
| --- | --- |
| Test Case ID | TEAMINDEX1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Team Name Link | Tests the text of the Team Name links to the correct page |  | 1. Click the team name. In the Sample data this will be in the format “BootstrapTeamXXX” | Browser loads the Build Info page for that particular Team. If no build information is available, the page should load and inform the user there is no data to display. |

|  |  |
| --- | --- |
| Test Case ID | TEAMINDEX2 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Team ID Link | Tests the text of the Team ID links to the correct page | 1. There is at least 1 commit/ build for this project. Otherwise this link is not present. | 1. Click the team ID text at the top left of the box containing summary information | Browser loads the Build Info page for that particular Team. |

|  |  |
| --- | --- |
| Test Case ID | TEAMINDEX3 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Prompt to log in | User should be prompted to log in if not already | 1. User is not logged in. |  | Prompt for user to log in appears before summary data |

|  |  |
| --- | --- |
| Test Case ID | TEAMINDEX4 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Prompt Login Link | Tests the text of the login prompt works correctly | 1. User is not logged in | 1. Click the text ‘Log in’ in the prompt above the summary data | Log in page loaded. |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Refresh | Tests that the Refresh button causes a page refresh |  | 1. Click the Refresh Icon beside the application name in the top black bar | Browser reloads the current page |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE2 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Footer Home Link | Tests that the footer Home link functions correctly |  | 1. Click the ‘Home’ text in the page footer | IBM company homepage is loaded |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE3 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Footer Help Link | Tests that the footer Help link functions correctly |  | 1. Click the ‘Help text in the page footer | Help Page is loaded |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE4 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Footer About Link | Tests that the footer About link functions correctly |  | 1. Click the ‘About text in the page footer | About Page is loaded |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE5 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Footer Submit Feedback Link | Tests that the footer Submit Feedback link functions correctly |  | 1. Click the ‘Submit Feedback’ text in the page footer | Default Mail application loaded with new mail open to [feedback@ibm.com](mailto:feedback@ibm.com) and subject = CD Dashboard Feedback |

|  |  |
| --- | --- |
| Test Case ID | TEMPLATE5 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Help Icon | Tests that the Help Icon functions correctly |  | 1. Click the Help Icon in the top right corner of the black bar. It is a blue circle with a question mark in the middle. | Help Page is loaded |

|  |  |
| --- | --- |
| Test Case ID | TEAMINFO1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Graph | Tests that the graph is the same as that displayed on the index page |  | 1. Note the shape of the graph on the index page. 2. Click the Team Name link | Graph should be the same values and shape on each page. |

|  |  |
| --- | --- |
| Test Case ID | TEAMINFO2 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Build Name Link | Tests that the build name link works as expected |  | 1. Click the Build Name link | Correct build info page is loaded. |

|  |  |
| --- | --- |
| Test Case ID | TEAMINFO3 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Team Member Email Link | Tests that the Team Member email link works as expected |  | 1. Click the Email address text | Default Mail application is loaded with the team member as recipient |

|  |  |
| --- | --- |
| Test Case ID | BUILDINFO1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| WorkItem ID Link | Tests that the WorkItem ID link works as expected |  | 1. Click the WorkItem ID text | Correct WorkItem info page is loaded. |

|  |  |
| --- | --- |
| Test Case ID | BUILDINFO2 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| WorkItem Created date Link | Tests that the WorkItem Created date link works as expected |  | 1. Click the WorkItem Created Date text | Correct WorkItem info page is loaded. |

|  |  |
| --- | --- |
| Test Case ID | BUILDINFO3 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| WorkItem Type Link | Tests that the WorkItem Type link works as expected |  | 1. Click the WorkItem Type text | Correct WorkItem info page is loaded. |

|  |  |
| --- | --- |
| Test Case ID | CRONJOB1 |
| Test Case Name | Purpose | Initiation Criteria | Steps | Expected Results |
| Cron job | Tests that the Cron job to update the database executes |  | 1. Monitor logs, should see Cron execution | Cron job executed after defined time period |

# 

# Test Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Test Case ID** | **Test Name** | **Tester** | **Date** | **Result** |
| 1 | BUILDINFO3 | Startup With Empty Database | Shane Murphy | 29/04/2014 | Pass |
| 2 | BUILDINFO2 | Start with existing data in database | Shane Murphy | 29/04/2014 | Pass |
| 3 | BUILDINFO1 | Start with no connection to RTC Server | Shane Murphy | 29/04/2014 | Pass |
| 4 | TEAMINFO3 | Fail if started with no Server Backend | Shane Murphy | 29/04/2014 | Pass |
| 5 | TEAMINFO2 | No Internet Connection | Shane Murphy | 29/04/2014 | Pass |
| 6 | TEAMINFO1 | Login | Shane Murphy | 29/04/2014 | Pass |
| 7 | TEMPLATE5 | Logout | Shane Murphy | 29/04/2014 | Pass |
| 8 | TEMPLATE5 | Add Subscriptions | Shane Murphy | 29/04/2014 | Pass |
| 9 | TEMPLATE4 | Remove Subscriptions | Shane Murphy | 29/04/2014 | Pass |
| 10 | TEMPLATE3 | View Profile | Shane Murphy | 29/04/2014 | Pass |
| 11 | TEMPLATE2 | Can’t view other user’s profile | Shane Murphy | 29/04/2014 | Pass |
| 12 | TEMPLATE1 | Team Name Link | Shane Murphy | 29/04/2014 | Pass |
| 13 | TEAMINDEX4 | Team ID Link | Shane Murphy | 29/04/2014 | Pass |
| 14 | TEAMINDEX3 | Prompt to log in | Shane Murphy | 29/04/2014 | Pass |
| 15 | TEAMINDEX2 | Prompt Login Link | Shane Murphy | 29/04/2014 | Pass |
| 16 | TEAMINDEX1 | Refresh | Shane Murphy | 29/04/2014 | Pass |
| 17 | USERPROFILE4 | Footer Home Link | Shane Murphy | 29/04/2014 | Pass |
| 18 | USERPROFILE3 | Footer Help Link | Shane Murphy | 29/04/2014 | Pass |
| 19 | USERPROFILE2 | Footer About Link | Shane Murphy | 29/04/2014 | Pass |
| 20 | USERPROFILE1 | Footer Submit Feedback Link | Shane Murphy | 29/04/2014 | Pass |
| 21 | LOGOUT1 | Help Icon | Shane Murphy | 29/04/2014 | Pass |
| 22 | LOGIN1 | Graph | Shane Murphy | 29/04/2014 | Pass |
| 23 | NETWORK1 | Build Name Link | Shane Murphy | 29/04/2014 | Pass |
| 24 | START4 | Team Member Email Link | Shane Murphy | 29/04/2014 | Pass |
| 25 | START3 | WorkItem ID Link | Shane Murphy | 29/04/2014 | Pass |
| 26 | START2 | WorkItem Created date Link | Shane Murphy | 29/04/2014 | Pass |
| 27 | START1 | WorkItem Type Link | Shane Murphy | 29/04/2014 | Pass |
| 28 | CRONJOB1 | Cron Job | Shane Murphy | 29/04/2014 | Pass |

# Future Extensions

Added interface for customizing the location/port of the server and the location/port of the RTC server.

Ability to select and compare 2 builds, displaying the delta between the two.

Additional metrics as defined in original spec, and more.

Customised views based on user roles.

# 

# User Manual

## Installation

The application is packaged in a Web Application Archive File (WAR) which can be deployed on any Java EE compliant application server. This WAR file can be found in the IBM\_CD\_DASHBOARD/target Directory. It is entitled IBM\_CD\_Dashboard-0.1.war.

The application requires a MySQL database titled ‘ibm’ to be running on the local machine on port 8889. The full URL for this should be the following format: localhost:8889/ibm or 127.0.0.1/ibm.

To utilise the RTC functionality of the project, it is expected that the RTC Server also be running locally on port 9443. The project only uses the RTC section of the server and so the full URL for this should be the following format: localhost:9443/ccm or 127.0.0.1/ccm.

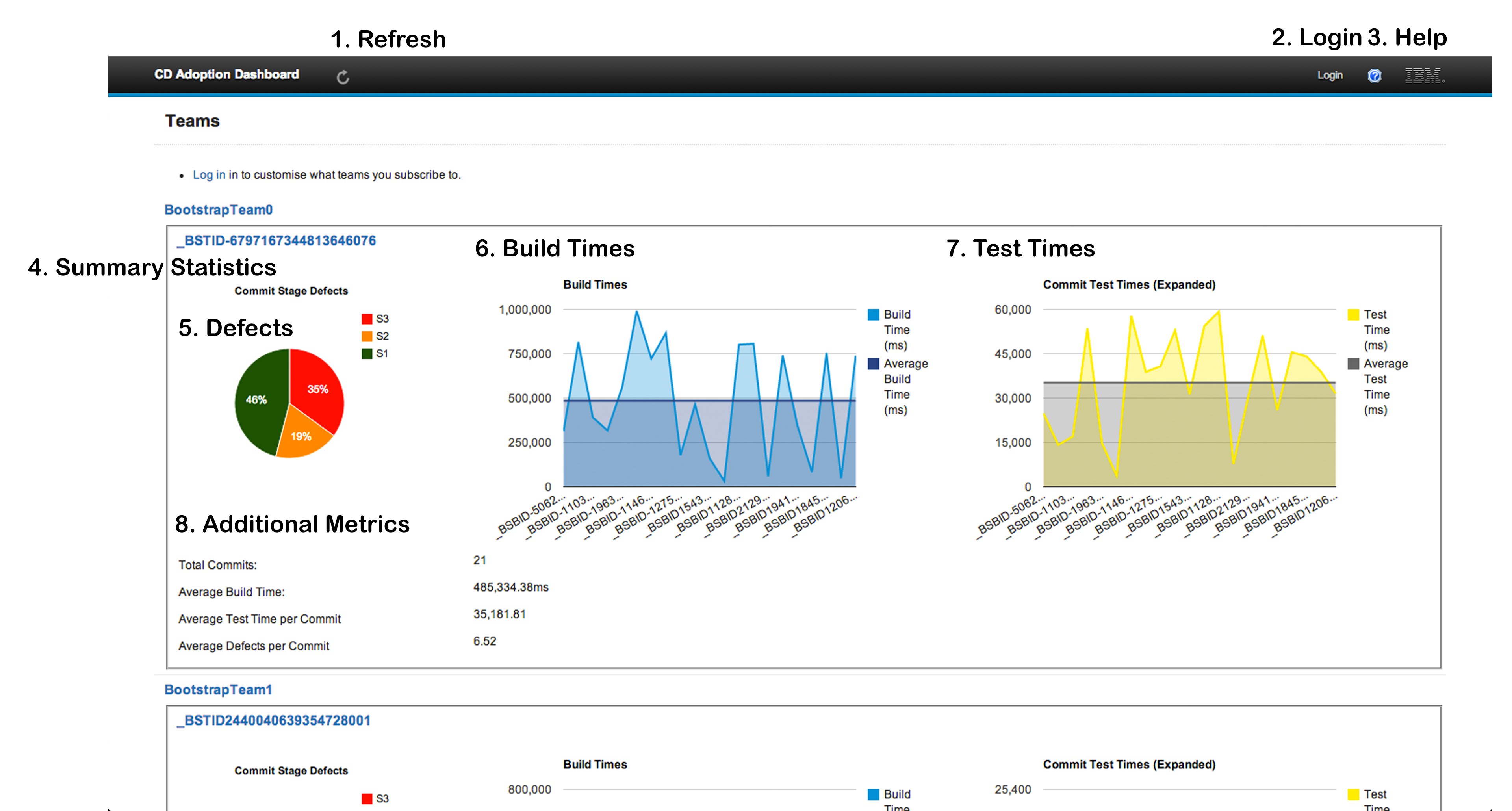
Once deployed the application initializes itself, pulling in all available data and creating two default users.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Username | Password | Role |
| 1 | user | password | User |
| 2 | admin | password | Admin |

## 

## Application Manual

### Home Screen



#### Refresh

This button causes the current page to refresh. It is dislayed on every page in the application.

#### Login

This redirects the user to the login page where they can supply their log in information in order to customize their views.

#### Help

A Link to any help information available to the user.

#### Summary Statistics

Each Team or Project has it’s own area in which to display summary statistics. The area is defined by a small border around the area, to distinguish between builds.

#### Defects

Defects are broken down into 3 severity groups, as defined by IBM. S1, S2 and S3. These are broken down into a pie chart to give an idea of the overall percentage of each defect group.

#### Build Times

Build times are displayed in a graph representing the times per build commit over time. The most recent commit is the right most build time information.

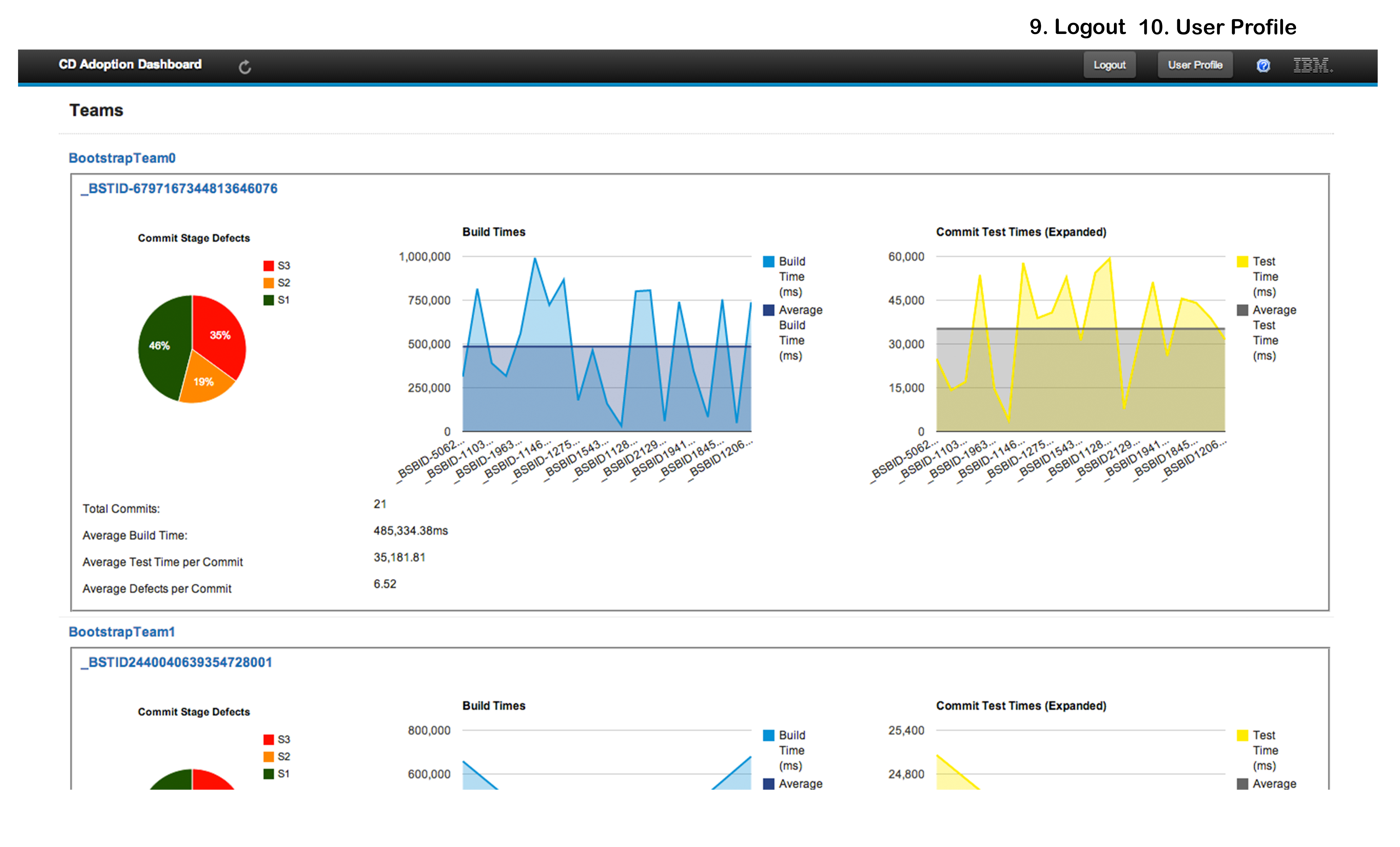
#### Test Times

Test times are displayed in a graph representing the times per Test commit over time. The most recent commit is the right most build’s time information.

#### Additional Metrics

Any additional pertinent information is displayed here where it is not appropriate to display it in a graphical image.

### Home Screen Logged In



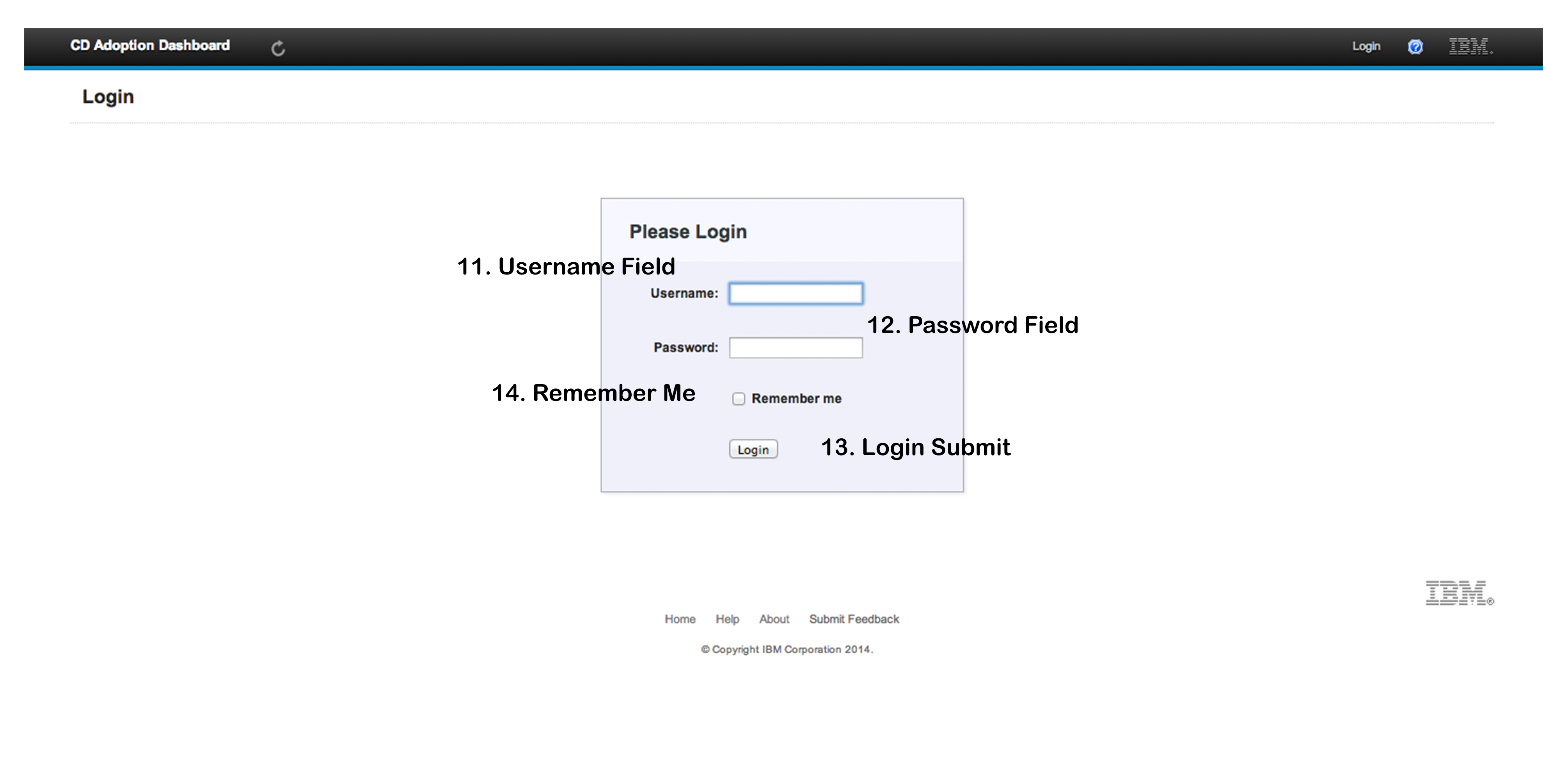
#### Logout

Link signs the user out from the application

#### User Profile

Link to edit the current users profile

### Login



#### Username Field

User enters their username here.

#### Password Field

User enters their password here.

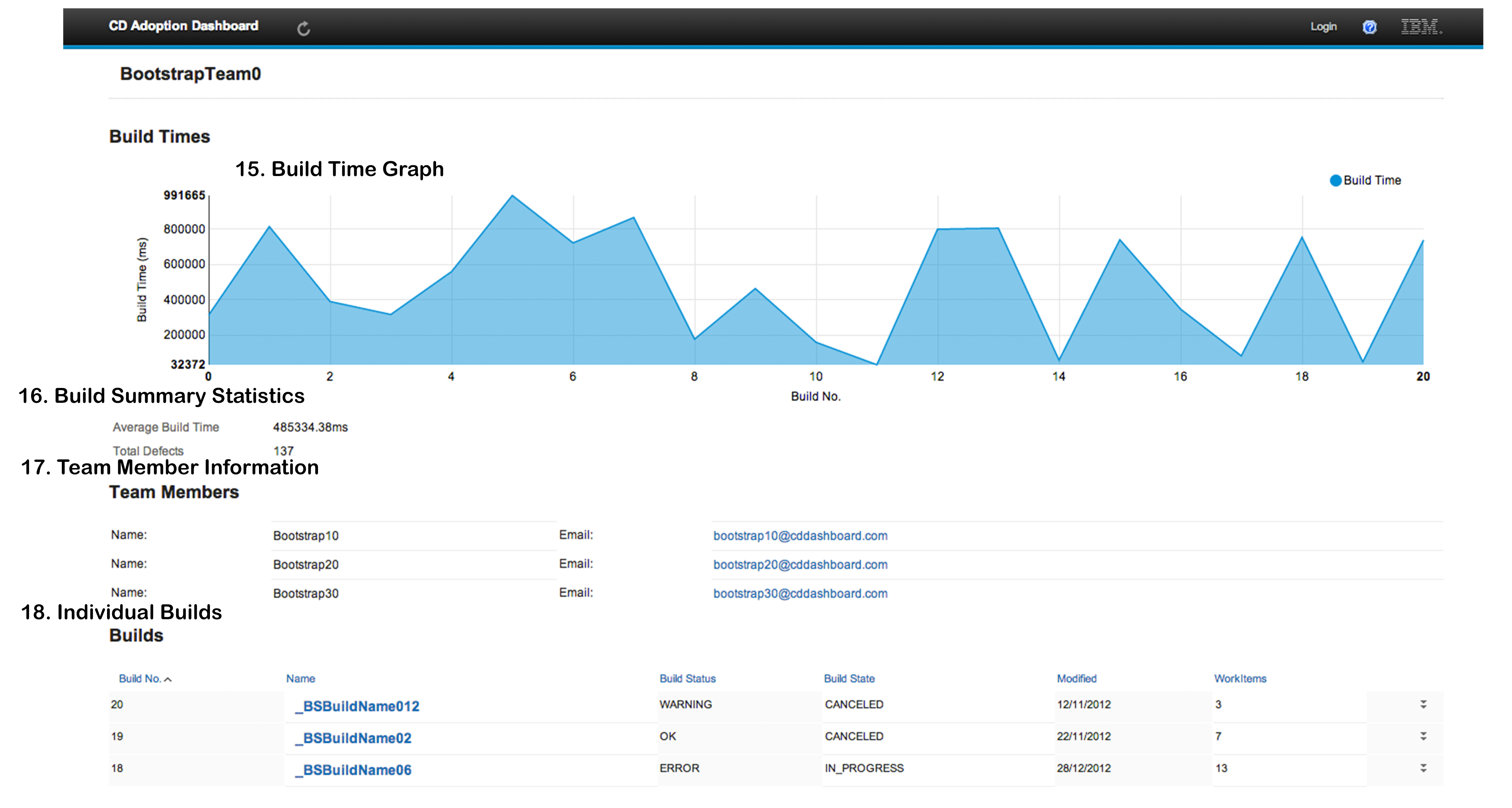
#### Login Submit

Submit the entered username and password for verification.

#### Remember Me

Persists the session cookie for this user past the default, which is the lifetime of the browser.

### Team Info



#### Build Time Graph

Build times are displayed in a graph representing the times per build commit over time. The most recent commit is the right most build time information.

#### Build Summary Statistics

Any additional pertinent information is displayed here where it is not appropriate to display it in a graphical image

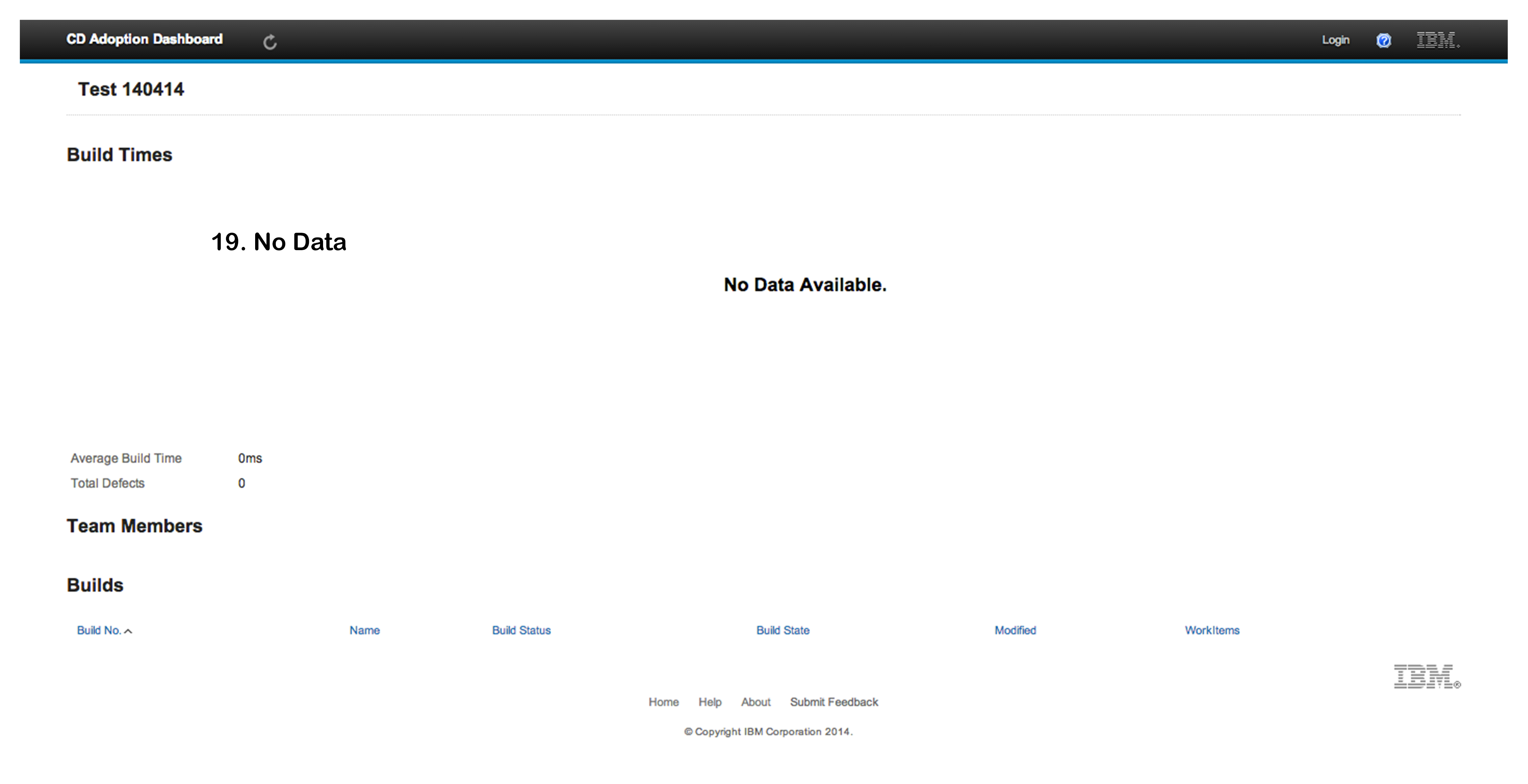
#### Team Member Information

Team members and their contact email address are displayed here. Each email address is a mailto link.

#### Individual Builds

Each build and some high level information about them are displayed here. Clicking on the build will allow you to view the build in higher granularity.

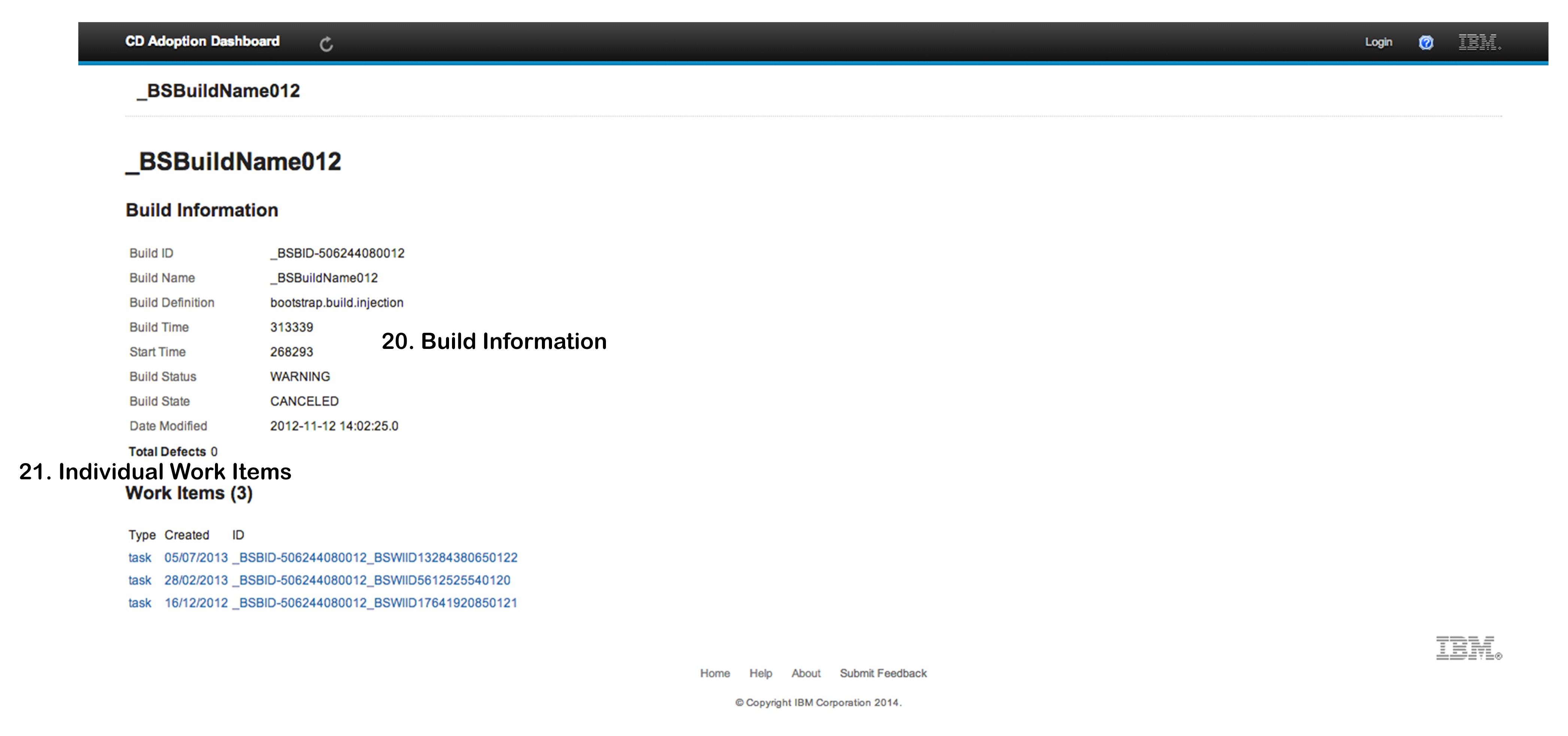
### Team Info No Data



#### No Data

If no Build information is present for the current project, the summary page will display as above.

### Build Info



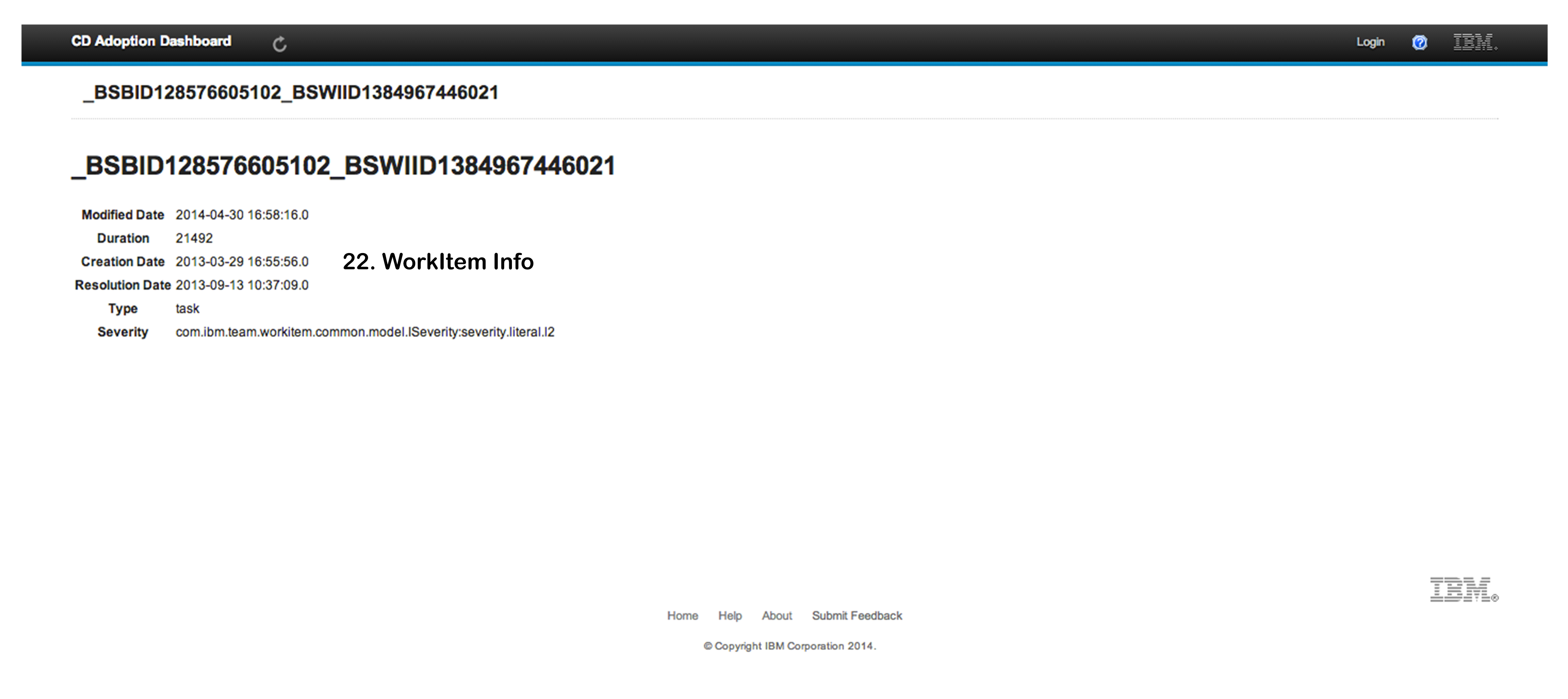
#### Build Information

Information specific to this build is displayed here.

#### Individual Work Items

Builds are comprised of Work Items, these are displayed here and can be viewed in more detail by clicking on them.

### WorkItem Info



#### WorkItem Information

Work Item information about this particular work item is displayed here

### About

### Macintosh HD:Users:shanemurphy:IBM-CD-Dashboard:Documentation:UserManualImages:About.png

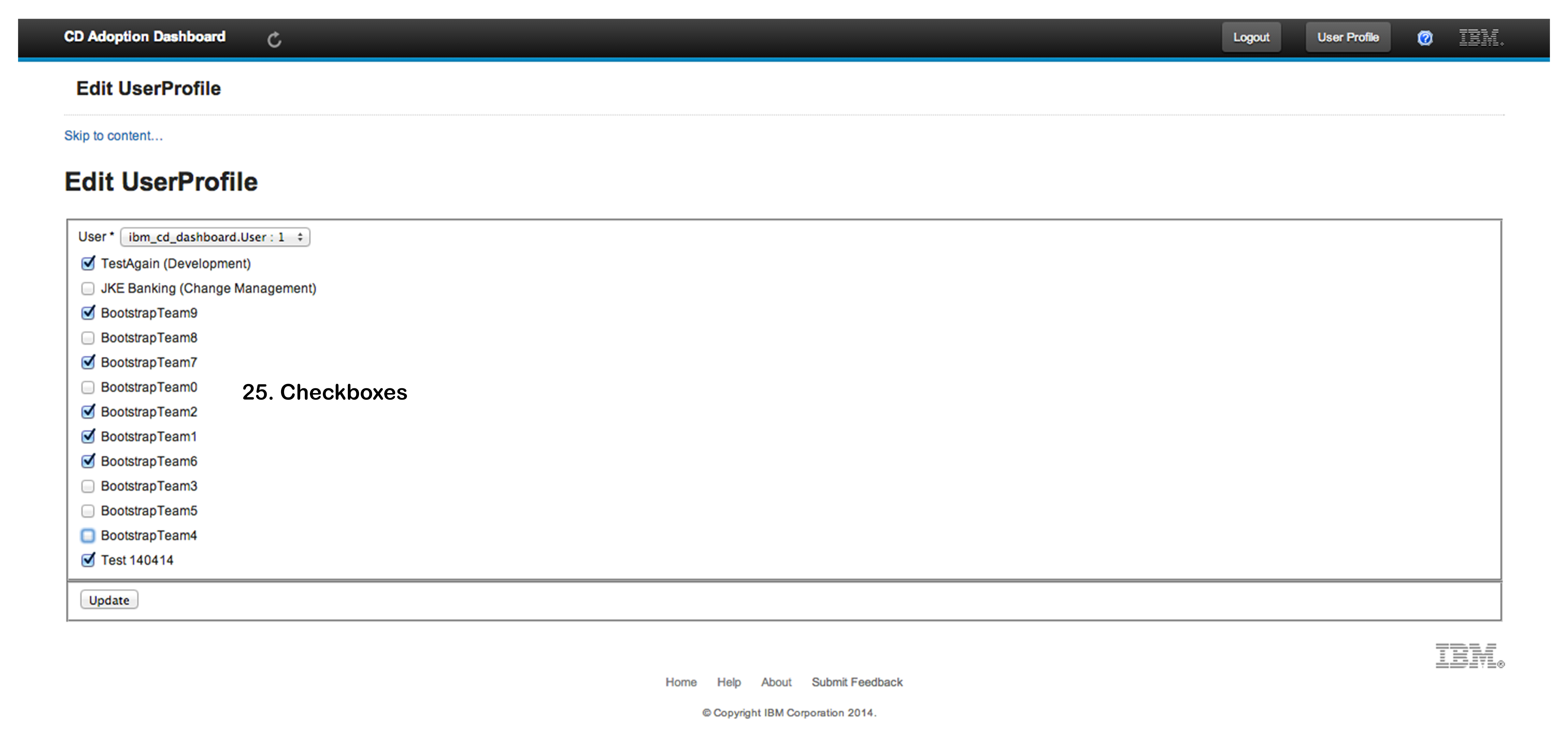
#### Project Overview

This page gives a brief overview of how the project was implemented.

#### Contact Information

Information on how to contact the developer is displayed here.

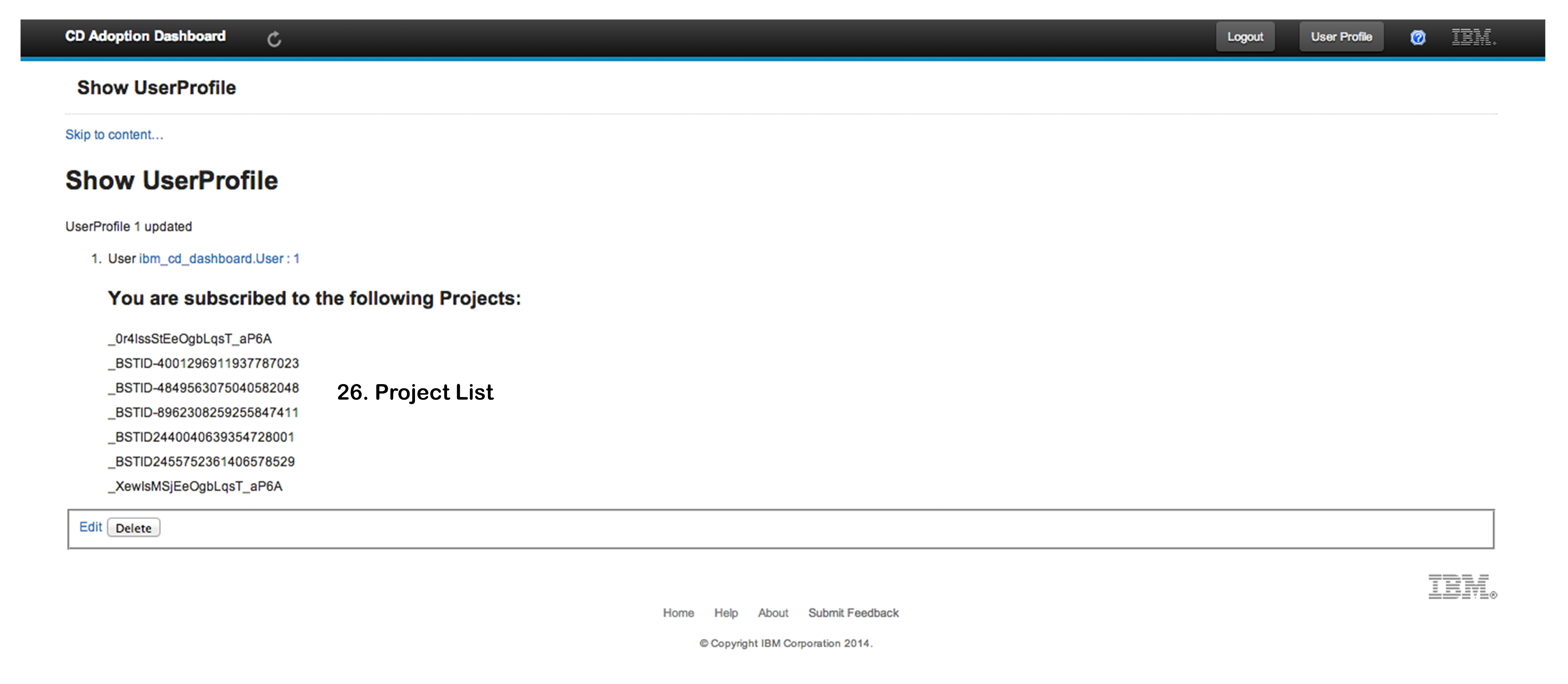
### User Profile Edit



#### Check Boxes

Selecting or deselecting these checkboxes subscribe or unsubscribe from projects.

### User Profile



#### Project List

A List of the current users subscribed projects are displayed here.

### Footer

### Macintosh HD:Users:shanemurphy:IBM-CD-Dashboard:Documentation:UserManualImages:Footer.png

#### Home

Redirects to the IBM home page.

#### Help

Redirects to the help page for the application.

#### About

Redirects to the about page.

#### Submit Feedback

Mailto link to email feedback.

# 

# Post Project Review

## A review of the project in its entirety

## Possible extensions of the project

Due to the rescoping of the progress during the development, there are plenty of additional metrics that could be easily included in the application. These could include, but are not limited to, API Test Coverage, Time from code submission to production, etc.

The project could move to a more build management type of role with the inclusion of functionality that would enable the user to select two or more teams, or builds and compare them directly. The delta between the two could be displayed to the users via graphs and additional summary statistics.

The user roles could be utilised further by allowing different default views to be presented to each user role group they belong to. For example team members might be interested in some of the finer grain details of their own builds, and these details could be pushed up to a higher level so as to be viewed at a glance along with all other relevant details.

## Review of Research, Analysis and Design phases

During the implementation phase, as the project progressed, it became apparent that the original class diagram was not suitable for the project, as the results began to come back. Much of the documentation described how the data would come back from the server, but in practice, it was clear that this documentation was out of date or describing some other part of the server. As such, the original schema was changed and simplified accordingly. The end result is as below.



The logical architecture of the application remained the same from the design phase through to the final submission.

## Review of Implementation phase

Overall the implementation went smoothly throughout the project. However, as we progressed, my mentor and I began to realize the scope of the project was too large to realistically complete in the time allocated. Having spent a large amount of time in the research phase trying to familiarize myself with the data sources from which the information would be gathered, the implementation phase of the project began somewhat later than most other projects. The project entered a yellow status sometime in the second semester, and the rescoping was carried out as a result of this.

Throughout the implementation phase, I was in constant contact with my mentor at IBM between email correspondence and our bi-weekly meetings. He was on hand during the implementation phase to query any unclear requirements or functionality definitions. He was also to help provide context to some of the more unfamiliar metrics.

# 

# Progress Diary

## Week Beginning 14/10/13

Created git repository

* using SourceTree as Client
  + <http://www.sourcetreeapp.com/>
* Formatted README.md file to display appropriately using Markdown

Investigating project management tools similar to Confluence.

* Trello
* LeanKitKanban
* Phabricator

Created Project Task Board

* https://trello.com/b/nQeWNCIM/ibm-cd-dashboard

Researching Jenkins

* How to get information from Jenkins remotely?
  + REST API in XML/JSON/Python format
  + XML format supports XPath include/exclude query patterns
  + Running Jenkins on a RasberryPi for test purposes at home?
    - RasPi has 30% CPU load when Jenkins is idle
    - 70-93% CPU Load when running Jenkins tasks and CPU is overclocked
      * Too slow and CPU intensive
    - Run locally/Virtualised environment?
    - Installed Jenkins Locally (localhost:8080)

Researching Node.js

* Installed Node.js

Researching knockout.js

Researching Angular.js

Researching Backbone.js

Researching Karma (Test runner for Javascript)

Researching MongoDB

Reading article on developing for RTC APIs

* http://rsjazz.wordpress.com/2013/02/28/setting-up-rational-team-concert-for-api-development/

Email reply from Brendan:

* MongoDB not approved for use within IBM, DB2 preferred choice
* Use Jasmine for Javascript tests
* Scope of project is just to display results, no control needed.

Installed Webstorm IDE 30 day trial - can get Educational licence

* Emailed Ben Walsh asking to fill out lecturer form for Educational licence.

Watching Introductory tutorials for Grunt

## Week Beginning 21/10/13

IBM Meeting

* Clarified Use Cases
* To send revised and completed Use Cases to Brendan for review
  + Submission due 28/10/13

Added Additional Use cases, to flesh these out tomorrow asap and forward to Brendan

Sent Use cases to Brendan for review.

Researching Groovy and Grails, as this seems like a viable option and similar to MVC in C# which I have experience with.

Installed grails and groovy via GVM

* http://gvmtool.net/

Installed IntelliJ plugin for groovy/grails

Installed Groovy/Grails Sublime Text 2 plugin

Set up Groovy SDK in IntelliJ

## Week Beginning 28/10/13

Installed Groovy/Grails and IDE plugins for IntelliJ and Eclipse.

* Grails not natively suported in IntelliJ CE.
* Emailed John to get Educational License for IntelliJ

Watched Groovy/Grails Tutorial

* <http://www.youtube.com/watch?v=8d1hp8n1stA>
* http://www.youtube.com/watch?v=cIjInOkUDy8

JMS - Asyncronous calls

Looking at IBM CSS files

* ICS UI Developer Guide (file:///Users/shanemurphy/IBM-CD-Dashboard/IBMTemplates/OneUI-3.0.3\_20120816-1522\_Documentation-BusinessPartnerFiles/docPartner/guidelines/quickstart.htm)

Created Grails project for Adoption Dashboard

* Looking at implementing the IBM CSS files into a template for views

Jenkins running on same port as default Grails server -

Error Server failed to start for port 8080: Address already in use (Use --stacktrace to see the full trace)

* Changed default launch port to :8090 (BuildConfig.groovy) grails.server.port.http = 8090

Use Static Resource imports instead of CSS folders in Grails? - http://grails.org/doc/latest/guide/theWebLayer.html#resources

## Week Beginning 6/11/12

Watching Groovy tutorials to get familliar with the language

* http://www.youtube.com/watch?v=-mRlc\_vjiqE
* http://www.youtube.com/watch?v=k7KNtLrkNsg

Created Technical Architecture document

Emailed Brendan with Logical Architecture document and questions relating to what data is relevant to and needs to be stored from each summary data to enable greater levels of granularity.

Created Class Diagrams

Updated Project Status Report slides

## Week Beginning 13/11/13

Brendan provided additional information regarding summary metrics:

* Build Duration - from RTC
* Manual testing Duration - Test Case Executions Records
* API Test Coverage - TBD - We don't know how API Test Coverage will be tracked. RQM would represent a tick box exercise time from final code submission to production (cloud and on-premises) -
* Jenkins will drive the commit but UrbanDeploy or CHEF will do the deployment. How long it takes to execute the pipeline. Jenkins.
* Defects opened per capita - RTC Query for Defects
* Cumulative open+deferred defects - RTC Query

Jazz Server/RTC/RQM not available for Mac OS X. Installing Ubuntu 13.10 via VirtualBox to install Jazz Server etc. through Linux. Will need to access this through the Virtual Box using NAT (Chapter 6 of VirtualBox guide)

Installed Jazz Team Server

* Change and Configuration Management
* Quality Management
* Requirements Management
* Lifecycle Project Administration

Configured NAT and Port Forwarding for virtual machine

* Can now access JTS from Host OS using localhost:9443 or localhost:9080

Installing the JKE Banking sample project

* This sample provides an in-flight perspective of the collaborations that occur on a typical Agile team. Walk through an iteration with the team as they deliver a new feature to support an important corporate initiative called Money that Matters.

Error creating the sample:

* com.ibm.team.lpa.automation.TaskException: CRJCA0009E The request could not be completed, because a 403 Forbidden status was returned. The request URI was https://localhost:9443/rm/clmSampleProject.hide details
* Error Stacktrace:
  + com.ibm.team.lpa.automation.TaskException: com.ibm.team.lpa.automation.TaskException: CRJCA0009E The request could not be completed, because a 403 Forbidden status was returned. The request URI was <https://localhost:9443/rm/clmSampleProject>.

at com.ibm.team.lpa.tasks.SampleValidationTask.validateApplicationRemote(SampleValidationTask.java:132)

at com.ibm.team.lpa.tasks.SampleValidationTask.runTask(SampleValidationTask.java:66)

at com.ibm.team.lpa.common.tasks.AbstractTask.run(AbstractTask.java:56)

at com.ibm.team.lpa.automation.CompositeTask.run(CompositeTask.java:208)

at com.ibm.team.lpa.automation.CompositeTask$MonitoredSubTask.run(CompositeTask.java:124)

at com.ibm.team.lpa.automation.CompositeTask.run(CompositeTask.java:208)

at com.ibm.team.lpa.common.tasks.LoggingCompositeTask.run(LoggingCompositeTask.java:37)

at com.ibm.team.lpa.automation.AsyncTask$TaskRunner.run(AsyncTask.java:40)

at java.lang.Thread.run(Thread.java:738)

Caused by: com.ibm.team.lpa.automation.TaskException: CRJCA0009E The request could not be completed, because a 403 Forbidden status was returned. The request URI was https://localhost:9443/rm/clmSampleProject.

at com.ibm.team.lpa.common.tasks.AbstractTask.handleError(AbstractTask.java:143)

at com.ibm.team.lpa.tasks.SampleValidationTask.validateApplicationRemote(SampleValidationTask.java:129)

... 8 more

* Needed to assign Enterprise Developer licence to my user as per - https://jazz.net/forum/questions/95134/unable-to-create-jke-in-401

Sample Project installed

Sample JKE Project has no data in it, need to drop and recreate.

Added lib dependencies to grails app.

RTC Server appears to start in the logs, sometimes it is accessible from the weblink, sometimes not. Server requests 4GB of memory at startup laptop currently only has 4GB or RAM memory installed (Will upgrade), changed

JAVA\_OPTS="$JAVA\_OPTS -Xmx4G"

JAVA\_OPTS="$JAVA\_OPTS -Xms4G"

in server.startup file to

JAVA\_OPTS="$JAVA\_OPTS -Xmx2G"

JAVA\_OPTS="$JAVA\_OPTS -Xms2G"

This has no effect, the server still will not startup. - Big Roadblock - emailed Brendan.

## Week Beginning 19/11/13

Ordered additional (16GB) RAM for laptop

Uninstalled Jenkins - uses a lot of RAM in the background even with no jobs.

Uninstalled RTC Server and reinstalled. Working fine now, waiting for RAM to arrive before attempting to install the sample project.

Working with Site Template - Added Title and Nav bar components as per the IBM OneUI Template. Linked in CSS files and file structure - CSS not being utilised.

Installed RTC Server on desktop computer on home network. Installed Sample Projects, Error, but looks as if all is there, will suffice until additional RAM arrives and can install locally on virtualbox. Can access this now through my local network at home.

## Week Beginning 27/11/13

Connecting to JTS I get:

Error 500: Internal Server Error

URI

/IBM\_CD\_Dashboard/rtcTest/index

Class

org.xml.sax.SAXParseException

Message

The reference to entity "etag" must end with the ';' delimiter.

Seems to be related to basic and form authentication

Changing Tomcat authentication from FORM to BASIC as per - https://jazz.net/forum/questions/100510/jazz-authentication-change-from-form-to-basic-on-tomacat

Form returned with 200 success code now.

Printing the returned XML from /ccm/oslc/workitems/catalog to screen produces one string "Project Areas" - need to install sample project JKE Banking on Virtual Machine JTS Server.

Installing the JKE Banking sample - Error.

* Licence issue.
* Assigned Rational Quality Manager - Quality Professional, Rational Requirements Composer - Analyst, and both Rational Team Concert - Developer licences to user.

Installing the JKE Banking sample - Error.

* Timeout
* Changed admin.properties file to increase timeout from 5minutes to 100minutes.

Installing CCM JKE Banking Sample

* Successful

Installing QM JKE Banking Sample

* After 2 hours, Chrome closed unexpectedly.
* Restarted laptop and server.
* Error: ID CRJAZ1972E - Error fetching server status information

Used repotools to drop QM tables - error taken care of.

Still issue with creating JKE prject - RM and QM projects recognised as already in existance for some reason - archived whole JKE project to try again, 500 error accessing whoami service.

Need to recreate tables I think. repotools -createTables.

Deleted JTS, reinstalled, started steps again, will spend less time starting again than recovering the server to useable state.

Installed sample project again, progress screen froze but left running regardless and seems to have installed everything, to double check this is correctly installed.

## Week Beginning 27/11/13

Looking into using ReportsRESTAPI.

* https://jazz.net/wiki/bin/view/Main/ReportsRESTAPI#Resources\_provided\_by\_RTC

Build Duration – Minutes - RTC

* buildResult (type: com.ibm.team.build.BuildResult)
* timeTaken (type: xs:long). How long the build took, in milliseconds
* https://jazz.net/wiki/bin/view/Main/ReportsRESTAPI#build

SPRs opened per capita – General quality measure (Number) - RTC

* workItem (com.ibm.team.workitem.WorkItemType)
* id (type: xs:string).
* type (e.g. "com.ibm team.workitem.defect"),
* name (type: xs:string).
* https://jazz.net/wiki/bin/view/Main/ReportsRESTAPI#com\_ibm\_team\_workitem\_WorkItemTy

Cumulative open+deferred - Number -RTC

* workItem (com.ibm.team.workitem.State)
* defects (technical debt)
* group (type: xs:string).
  + state group is a process-independent grouping of states, which is useful for creating reports which are not dependent on a particular process but still need to know, for example, whether work items are open or closed. Every state belongs to one of the following state groups: "OPEN\_STATES", "CLOSED\_STATES", "IN\_PROGRESS\_STATES".
* https://jazz.net/wiki/bin/view/Main/ReportsRESTAPI#com\_ibm\_team\_workitem\_State

User requires RQM Connector licence and RQM Contributor licence in addition to licences already assigned - added these to my user profile.

Manual testing duration (automatable) - Person Weeks – RQM/Test Case Execution records

Manual testing duration (non-automatable) - Person Weeks – RQM

Time from final code submission to production (Cloud) – Days - DevOps/UrbanDeploy/CHEF/(Jenkins)Pipeline

Time from final code submission to GA (On-Premises) - Days - DevOps/UrbanDeploy/CHEF/(Jenkins)

API Test Coverage (via automated measurement - % Value - TBD - Unlikely to be RQM

Came across this on a forum - "Tip: In general, use the OSLC QM V2 API unless requiring unsupported resources." -http://sleroyblog.wordpress.com/2013/04/09/querying-rqm-40-through-oslc-and-rest-api/

* Will look into OSLC instead of Reporable API.
* Not trying to provide additional functionality on top of the CLM, so Reportable REST should be ok for my needs. OSLC may be better supported.
  + "Consumers should not rely on or remember any specific URLs, or perform 'path math' on URLs. Instead, they should use the discovery chain. All they need to know is the URL of the root services document." Always start from root services.

Exploring Reportable REST API - https://localhost:9443/ccm/rpt/repository

Updating class diagrams

* Moving models under newly identified packages (i.e Build Duration under RTC instead of Jenkins)
* Identifying useful information to hold in model for finer detail when needed

Starting to create domain models in Grails app

Looking at Plain Java API for RTC

Able to connect to repository

Can identify project areas and display names

Trying to access build information

Able to access BuildResult and BuildDefinitions

* have access to Build Durations

Seperating this logic into a Service as recommended by the Grails best practices.

Issue with TeamPlatform.shutdown() - connection should stay open for entire life of the application

## Week Beginning 4/12/13

Refactored code in Service - seperated login, startup etc into seperate methods and added groovydoc comments for some methods.

Trying to get a list of contributors for each ProjectArea. GetMembers returns a IContributorHandle[], trying to get an IContributor from this handle using the ITeamRepository.itemManager().fetchCompleteItems() but method will not accept the IContributorHandle[] type.

Created method getProjectMembers(IPRojectArea) which returns all members for the project argument as a list of IContributors.

Printed members to test view grouped by project area.

Trying to access WorkItems under each project.

* Code should return all workitems under each project but only returns one
* Returns the same for each project which is correct. In the Web GUI, seems like there is more WorkItems, unsure if my understanding of a workitem is wrong or the code is wrong.

Updating Project Status Report.

WorkItems printed out to screen

## Week Beginning 11/12/13

Checking code examples of accessing RQM - https://jazz.net/wiki/bin/view/Main/RqmApi#Code\_Examples.%A0

## Week Beginning 18/12/13

No Java API for RQM, REST/OSLC only.

## Week Beginning 25/12/13

Trying to understand how to traverse the REST API resources.

## Week Beginning 22/01/13

Returning to RTC Java API to finish more functionality here before continuing on to RQM REST inferface in order to have something properly demoable for IBM meetings 5/2/14.

Would be nice to have the builds saving to the database, results being displayed on screen and some graphs etc. if possible. May not get all implemented in a week, but these are the goals.

Added mapping to domain classes to override default primary key for Team and Build domains.

Created method to get work items associated with a build result and return these as workitemimpl instead of workitemhandleimpl.

Class structure revision Team >>1>n>> Build >>1>n>> WorkItems

One team (Project) has many builds (BuildResult) which can contain many WorkItems. Workitems will just be used for granularity in the application, as well as for defect counting.

Cannot cascade save Team>Build>WorkItems because of null issues. Was able to solve this on WorkItems by allowing Builds to contain nullable WorkItems.

Set all fields to null to check behaviour - No change

Teams are now saving correctly.

Able to cascade delete teams.

created domainService service, added delete and populate methods to this.

Original implementation of primary key/foreign key and constraints were wrong. Fixed this issue.

Errors thrown - One WorkItem can be associated with more than one build. This violates the new constraints which makes workitem IDs be unique (id's assigned by itemId from RTC). Created many-to-many association between workItems and builds.

* Error Caused by: java.lang.OutOfMemoryError: PermGen space

Key issue on Workitems, commented out code, will come back to this.

Created table view of builds with link to delve deeper into team/project and view build information. It's probably about time I started to seperate the methods into appropriate controllers, currently just using one controller RTCTestController for everything and all views under this (currently only two views)

## Week Beginning 29/01/14

Seperated everything into seperate views and controllers (TeamController, BuildController and WorkItemController)

Created views to display information for Teams, Buils and WorkItems. Team now displays average build for the project overall.

Trying to get workItems to save properly, but unable to figure out why Key issues are happening, tried to create break points, but breakpoints not breaking the code as they should. Red circle for breakpoint doesn't have 'v' inside it, just empty red circle.

Managed to get breakpoints working. Set grails.project.fork = false.

Stepping through the code, breakpoint at if(WorkItem.get(id))

Code breaks on newTeam.save()

Seperated out code from populateDatabase() into 3 seperate methods populateTeams(), populateBuilds() and populateWorkItems()

error persists. debugging.

New error on DomainClass.get() method:

Error java.lang.IllegalStateException: Method on class [ibm\_cd\_dashboard.WorkItem] was used outside of a Grails application. If running in the context of a test using the mocking API or bootstrap Grails correctly.

| Error at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

| Error at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)

| Error at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

| Error at java.lang.reflect.Constructor.newInstance(Constructor.java:526)

fixed getBuildWorkItems(IBuildResult buildResult) method to return null if no work items exist for build.

| Error groovy.lang.MissingMethodException: No signature of method: ibm\_cd\_dashboard.Team.addToBuilds() is applicable for argument types: (ibm\_cd\_dashboard.Build) values: [ibm\_cd\_dashboard.Build : (unsaved)]

The key constraint error is coming from the archived project and the current project being the same. so workitems are the same with same ID, maybe...

Workitems added to build. Changed unique identifier to be concatination of build id and workitem id.

Changed some things and this is not being reflected in workitems

* | Error 2014-02-03 16:05:16,948 [http-bio-8090-exec-1] ERROR util.JDBCExceptionReporter - Referential integrity constraint violation: "FKCF0A9F832E451BAA: PUBLIC.BUILD\_WORK\_ITEMS FOREIGN KEY(WORK\_ITEM\_ID) REFERENCES PUBLIC.WORK\_ITEM(WORK\_ITEM\_ID) ('\_UukD4FfVEeOe-fhOpjWp0Q\_\_n4XRMFfTEeOe-fhOpjWp0Q')"; SQL statement:

insert into build\_work\_items (build\_id, work\_item\_id) values (?, ?) [23506-173]

Summary field is no longer correct. added a literal value into this for the time being, this solved the error issues.

Couldn't drop tables. added constraints to enable proper table structure as found on Stack overflow

* http://stackoverflow.com/questions/16869006/grom-create-needless-table-with-many-to-many-mapping
* http://stackoverflow.com/questions/4131174/how-do-i-override-the-cascade-delete-for-a-relation-in-grails-gorm

Work Items no longer being saved.

## Week Beginning 05/02/14

45 WorkItems should be associated with each project. only 14 being associated with one project and 0 witht he other.

45 workItems being saved to the database. on the second loop.

45 workItems asigned to builds. miscalculation in workItems count!!!!

Question on forum https://jazz.net/forum/questions/141572/get-all-work-items-for-a-build-result

Reply to answer on jazz.net. Went to try implement it, but all my virtual machines are missing. error -102(File not Found)

Reinstalled IBM VM and RTC server. Installed JKE Banking sample again.

## Week Beginning 12/02/14

Creating sitemesh templates for the site to include IBM style.

Created "common" module for css, this will combine all required css into one file and compress automatically optimising the imports.

Able to get CSS working. adding appropriate classes to existing views.

## Week Beginning 19/2/14

To try verify workitem count in the GUI compared to what's returned and email brendan asking for help, recommended by Gary and John.

* Total work Items should be 89.
* There should be 34 defects.
* 24 of these shoudl be Tasks.
* 25 Stories
* 2 Track Build Items
* 2 Impediments
* 2 Retrospectives

Installed D3 plugin into application

Trying to pass buildtimes variable to javascript/d3

* 'Unexpected token ILLEGAL'

Passing JSONObject to view, in the source it is references as

var jsonOb = \u007b\u0022M20140211-0211\u0022:404413\u002c\u0022M20140210-0353\u0022:301858\u002c\u0022C20140128-1513\u0022:315050\u002c\u0022C20140204-0152\u0022:376989\u002c\u0022Q20140210-0924\u0022:375980\u002c\u0022Q20140209-2358\u0022:306343\u002c\u0022P20140211-0957\u0022:380815\u002c\u0022I20140203-1713\u0022:400477\u002c\u0022P20140211-1604\u0022:376959\u002c\u0022I20140128-1032\u0022:342344\u002c\u0022C20140204-0641\u0022:327617\u002c\u0022I20140210-1538\u0022:406061\u002c\u0022C20140210-0929\u0022:328254\u002c\u0022M20140209-2112\u0022:408360\u007d;

passing in JSON now results in &quot in the objects.

for(var x in [{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}},{&quot;class&quot;:&quot;grails.converters.JSON&quot;,&quot;depth&quot;:0,&quot;writer&quot;:{&quot;class&quot;:&quot;org.codehaus.groovy.grails.web.json.JSONWriter&quot;}}]){

times.add(x);

}

Researching how to integrate Jenkins with RTC.

Installed github plugin to jenkins for auce demo install

* https://saucelabs.com/jenkins/4

Managed to pass Groovy object into View and manipulate with Javascript. put values of Groovy object into two 1D arrays, with corresponding indexes.

Used a map to pass details into D3.js using the nv.d3.js models to create a line chart of the build times in order of modified date. Added another row to table of builds to have build number which is references in the line graph on the x axis.

* http://nvd3.org/ghpages/line.html

Trying to get TeamMembers to display on teaminfo page. Team members are being returned and associated with thread right through the PopulateTeams() method. but after this method, the field is null.

## Week Beginning 26/02/14

Changed default page to load the TeamController/checkState action.

Implementing logging in the application.

Log files creating, but empty.

Adding logging for exceptions etc.

Changed server back to basic form authentication for REST access for RQM

Trying to parse RDF/XML, easy to do with XML, not sure how to do with RDF/XML

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<oslc\_disc:ServiceProviderCatalog rdf:about="https://localhost:9443/qm/oslc\_qm/catalog" xmlns:jproc="http://jazz.net/xmlns/prod/jazz/process/1.0/" xmlns:oslc\_qm="http://open-services.net/xmlns/qm/1.0/" xmlns:ns5="http://open-services.net/ns/qm#" xmlns="http://purl.org/dc/elements/1.1/" xmlns:ns8="http://jazz.net/ns/qm/rqm#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:oslc\_disc="http://open-services.net/xmlns/discovery/1.0/" xmlns:jpres="http://jazz.net/xmlns/prod/jazz/presentation/1.0/" xmlns:ns3="http://open-services.net/ns/core#" xmlns:calm="http://jazz.net/xmlns/prod/jazz/calm/1.0/" xmlns:dc="http://purl.org/dc/terms/">

<dc:title>RQM Quality Management Service Provider Catalog</dc:title>

<oslc\_disc:entry>

<oslc\_disc:ServiceProvider>

<dc:title>JKE Banking (Quality Management)</dc:title>

<oslc\_disc:details rdf:resource="https://localhost:9443/qm/process/project-areas/\_CoPfAJMuEeO5Z-budGdtwQ"/>

<oslc\_disc:services rdf:resource="https://localhost:9443/qm/oslc\_qm/contexts/\_CoPfAJMuEeO5Z-budGdtwQ/services.xml"/>

<jproc:consumerRegistry rdf:resource="https://localhost:9443/qm/process/project-areas/\_CoPfAJMuEeO5Z-budGdtwQ/links"/>

</oslc\_disc:ServiceProvider>

</oslc\_disc:entry>

</oslc\_disc:ServiceProviderCatalog>

Going to change code logic to check if domain has any teams, if not, populate teams, if it has teams but they have not been modified in some time period, check the server for additional information, otherwise, load the index page.

Researching Quartz for AJAX type request to the server.

Installed grails quartz plugin

* create-job script not there
  + needed to issue command grails refresh-dependencies - create-job script now available.

Created methods and variables to hold and set DomainLastModified and ServerLastModified, check whether DomainLastModified < ServerLastModified to see if Domain is up to date, can add this to a quartz job to check priodically.

Added Quartz job to run eevery 2 hours. need to implement method in service to be called by this Job to actually check the server. Method stub in DomainServer for this.

## Week Beginning 05/03/14

Looking at unit testing application.

JUnit vs Spock.

Junit s

Groovy Mock, Gmock, EasyMock, JMock, Mockito.

Jazz forum recommends Mockito, this is what IBM use internally.

Using Mockito external jar I get error:

* | Error Compilation error compiling [unit] tests: startup failed:

/Users/shanemurphy/IBM-CD-Dashboard/IBM\_CD\_Dashboard/test/unit/ibm\_cd\_dashboard/DomainServiceSpec.groovy: 5: unable to resolve class org.mockito.Mockito

@ line 5, column 1.

import org.mockito.Mockito

^

1 error

(Use --stacktrace to see the full trace)

* + Needed to ad to lib folder aswell as import as external library.

Error on the .when() method of Mockito:

No signature of method: com.ibm.team.process.common.IProjectArea$$EnhancerByMockitoWithCGLIB$$9b0c8d8.when() is applicable for argument types: (null) values: [null]

Possible solutions: wait(), grep(), wait(long), use([Ljava.lang.Object;), grep(java.lang.Object), with(groovy.lang.Closure)

groovy.lang.MissingMethodException: No signature of method: com.ibm.team.process.common.IProjectArea$$EnhancerByMockitoWithCGLIB$$9b0c8d8.when() is applicable for argument types: (null) values: [null]

Possible solutions: wait(), grep(), wait(long), use([Ljava.lang.Object;), grep(java.lang.Object), with(groovy.lang.Closure)

at ibm\_cd\_dashboard.DomainServiceSpec.setup(DomainServiceSpec.groovy:16)

Reply from Brendan - Spock is not approved for use in IBM - use JUnit.

IBM Meeting - Yellow status. Issues surrounding the workitems is an internal issue and outside of the scope of the project - don't know how to resolve, move on. The main point of the project is the GUI, should concentrate on providing mock data, showing an understanding of how the data would be collected from the server, but not actually having to fetch from the server. Focus on User Stories of user roles and providing different views to different users. Focus on bringing more information on first page and only delve deeper where really nessecary. Should be able to get all information required at a glance.

Created user and role domain. Following tutorial - http://grails-plugins.github.io/grails-spring-security-core/docs/manual/guide/single.html#tutorials

Can now login and logout, changed logout to accept GET requests. To implement profile editing page next where user can select projects from all projects in the database to associate with their profile.

## Week Beginning 19/3/14

Adding userprofile to user domain

error starting application - | Error 2014-03-20 22:06:49,195 [localhost-startStop-1] ERROR util.JDBCExceptionReporter - Parameter "#2" is not set; SQL statement:

## Week Beginning 27/3/14

User Profile linked, edit page lists current teams/projects with a checkbox beside them. If the team/project is listed in the users profile list of projects, the checkbox will be ticked already, showing it has already been selected.

Problem at the minute - UserProfile doesn't persist.

Seperate issue:

GRAILS-7799: Subtype 'grails.plugin.springsecurity.SpringSecurityService$$EnhancerByCGLIB$$8288660c' of reloadable type grails.plugin.springsecurity.SpringSecurityService is not reloadable: may not see changes reloaded in this hierarchy (please comment on that jira)

Bootstrapped random data into current domain objects using random integers etc. All working. Only RTC methods not being called so only bootstrap data showing.

Got around this by calling populateTeams() method from bootstrap.

Seperated info in main index (TeamController) nested in table into partial view \_teamData

UserProfile Issue resolved. now persisting. cannot store Lists in domains as you might expect the be able to.

* List<String> projects

changed to

List projects

static hasMany = [projects:String]

Logged in users can now customise which teams they see in their dashboard when they're logged in. When a user is not logged in, all teams are shown

Meeting with Brendan, dropped 3 requirements, one as not applicable and 2 as outside the newly defined scope of a commit phase dashboard. Summary : We will refocus the project around delivering a CD Dashboard around the Commit Phase and Defects. Don’t bother unit testing the application, this is not a requirement.

Working on UI, created a HomeController, this will house methods to display an About page as well as the Help section.

Added template for about page

## Week Beginning 2/4/14

Looking at identifying s1, s2, s3 severity for work items. WorkItems have a severity of NORMAL, MINOR, MAJOR, CRITICAL or BLOCKER (com.ibm.team.workitem.common.model.ISeverity:severity.literal.l1 - com.ibm.team.workitem.common.model.ISeverity:severity.literal.l6)

Count broken down by severity and added to view.

Doesn't seem to be any access to the State grouping for workitems. The state is now called state2, so perhaps this was removed in the newer state version.

Will just need to decide what state corresponds to Open/In Progress/Closed

Replaced d3.js with Google visualisation API to create graphs/charts. No issue with repeated graphs now - not sure whether to replace existing d3 graph in teamInfo view.

Implemented bar chart to show breakdown of defects in Team.

Changed this to pie chart.

## Week Beginning 9/4/14

Beginning to implement methods to update only out of date (compared with the server) domain objects, as triggered in the cron job.

Refactoring methods - seperating large methods into a number of reuseable parts, e.g createNewBuildObject, createNewWorkItemObject etc.

Brendan advised to group the l1 - l6 defects as follows:

* S1 - Critical + Blocker
* S2 - Major
* S3 - Normal, Minor

Also wants the chart for build time and test time seperated out into two seperate charts as in reality, the difference between build times and test times would be too large to be able to read the test times on the same chart.

Said the wording on the page should be changed to reflect the new scope - mostly 'build' should be changed to 'commit'

Implemented changes Brendan asked for. Layout is now different to screenshot sent in for the project booklet, emails John asking if I could resubmit

* Can submit anytime before Friday - to email John the new screenshot.

Changed "Refresh Page" Text to circular refresh arrow.

Fixed issues with images not linking properly on all pages - use <g:img /> instead of <img />

## Week Beginning 16/4/14

Tested creating new project area - this was recognised by the app and the domain was updated as expected - however the index page isn't null safe - error on page now because the team.builds loop is accessing null objects - to fix.

Trial Licenses expired again. Cannot create workItems etc. - Tried reinstalling the 10 Developer free licenses, but this is not enough. Can still create project areas, which demonstrates the functionality of refreshing page etc.

Made all scripting/groovy code in the views null safe.

Only allow to load partial view (graphs) if there are builds/commits for the build.

“There are no commits defined for this project" Displayed for projects with no builds/commits associated with them.

Emailed Brendan - it is ok that this functionality isn't 100%

Reply from Brendan - License issues are a valid reason not to be able to demonstrate this.

Finished method (i think) to update domain.

General refactoring and code cleanup - refactroing methods, deleting now unused methods, adding Javadocs.

Trying to get index page to reload automatically when updated.

Difficulty in doing this because a the update is detected in a Job and passed to a service, there is not user entity associated, so refreshes cannot be done this way. It is bad practice to pass from Service or Job into controller for this reason.

Installed DB2

Errors: bash: ulimit: open files: cannot modify limit: Invalid argument

Increased mac open files using

* sudo launchctl limit maxfiles 1000000 10000000

Error: SQL1220N The database manager failed to allocate shared memory.

Whole day spent on this. deleted everything and started again, worked after 10mins.

Created db2 databse node

Node 1 entry:

Node name = DB2\_01

Comment =

Directory entry type = LOCAL

Protocol = TCPIP

Hostname = 10.134.5.10

Service name = 50000

## Week Beginning 23/4/14

Brendan recommended that if DB2 is proving too difficult, I could use MySQL or Mongo. Also note why I would swap out existing database in slides for next week.

Not all of the RTCService methods try or catch blocks are properly implemented. TODO

Long error messages are caused from me printing out the stack trace as a place holder for proper exception handling.

Looking into MAMP to run MySQL database

Installed MAMP

Added MySQL database

Configured Hibernate to work with MySQL and the MAMP server.

Fixed average defects problem (wasn't displaying)

Team Members functionality fixed and working.

Added 404 not found page.

## Week beginning 30/4/14

Project tidy up and documentation.

# Bibliography

**Unsupported source type (ElectronicSource) for source Rat.**

AngularJS, n.d. *AngularJS*. [Online] Available at: [http://angularjs.org/](http://angularjs.org/%20).

Backbone, n.d. *Backbone*. [Online] Available at: <http://backbonejs.org/>.

Bower, n.d. *Bower*. [Online] Available at: <http://bower.io/>.

Codecentric, n.d. *Continuous Delivery in the Cloud - Part 6: Create a Dashboard of your System*. [Online] Available at: [https://blog.codecentric.de/en/2012/06/continuous-delivery-in-the-cloud-part6-create-a-dashbord-of-your-system/](https://blog.codecentric.de/en/2012/06/continuous-delivery-in-the-cloud-part6-create-a-dashbord-of-your-system/%20).

Coffee, n.d. *Coffee*. [Online] Available at: <http://coffeescript.org/>.

D3, n.d. *D3*. [Online] Available at: <http://d3js.org/>.

Grunt, n.d. *Grunt*. [Online] Available at: [http://gruntjs.com/](http://gruntjs.com/%20).

Grails, n.d. *Grails*. [Online] Available at: <http://grails.org/>.

IBM Rational User Education, n.d. *Introduction to Rational Team Concert v4.x Source Control Management*. [Online] Available at: [https://www.youtube.com/watch?v=X7nrHC9YO4Y](https://www.youtube.com/watch?v=X7nrHC9YO4Y%20).

IBM, n.d. *DB2*. [Online] Available at: <http://www-01.ibm.com/software/data/db2/>.

IBM, n.d. *RTC API Javadoc*. [Online] Available at: [https://jazz.net/downloads/rational-team-concert/releases/4.0/RTC-Client-plainJavaLib-API-javadoc-4.0.zip](https://jazz.net/downloads/rational-team-concert/releases/4.0/RTC-Client-plainJavaLib-API-javadoc-4.0.zip%20).

Jazz.net, n.d. *Rational Quality Manager*. [Online] Available at: <https://jazz.net/products/rational-quality-manager/>.

Jasmine, n.d. *Jasmine Introduction*. [Online] Available at: <http://jasmine.github.io/2.0/introduction.html>.

Jetbrains, n.d. *Webstorm*. [Online] Available at: <http://www.jetbrains.com/webstorm/>.

Karma , n.d. *Karma Runner*. [Online] Available at: <http://karma-runner.github.io/0.10/index.html>.

Knockout, n.d. *Knockout*. [Online] Available at: [http://knockoutjs.com/index.html](http://knockoutjs.com/index.html%20).

LESS, n.d. *LESS*. [Online] Available at: [http://lesscss.org/](http://lesscss.org/%20).

Node, n.d. *Node*. [Online] Available at: <http://nodejs.org/>.

MongoDb, n.d. *MongoDb*. [Online] Available at: <https://www.mongodb.org/>.

QUnit, n.d. *QUnit*. [Online] Available at: <https://qunitjs.com/>.

Python, n.d. *Jenkins API*. [Online] Available at: [https://pypi.python.org/pypi/jenkinsapi](%20https://pypi.python.org/pypi/jenkinsapi).

Schoon, R., 2013. *What API’s are Available for RTC and What Can You Extend?* [Online] Available at: [http://rsjazz.wordpress.com/2013/03/14/what-apis-are-available-for-rtc-and-what-can-you-extend/](http://rsjazz.wordpress.com/2013/03/14/what-apis-are-available-for-rtc-and-what-can-you-extend/%20).

Schoon, R., 2013. *Setting up Rational Team Concert for API Development*. [Online] Available at: <http://rsjazz.wordpress.com/2013/02/28/setting-up-rational-team-concert-for-api-development/>.

RubyGems, n.d. *Jenkins API Client*. [Online] Available at: [https://rubygems.org/gems/jenkins\_api\_client](https://rubygems.org/gems/jenkins_api_client%20).

Twitter, n.d. *Twitter Bootstrap*. [Online] Available at: <http://getbootstrap.com/>.