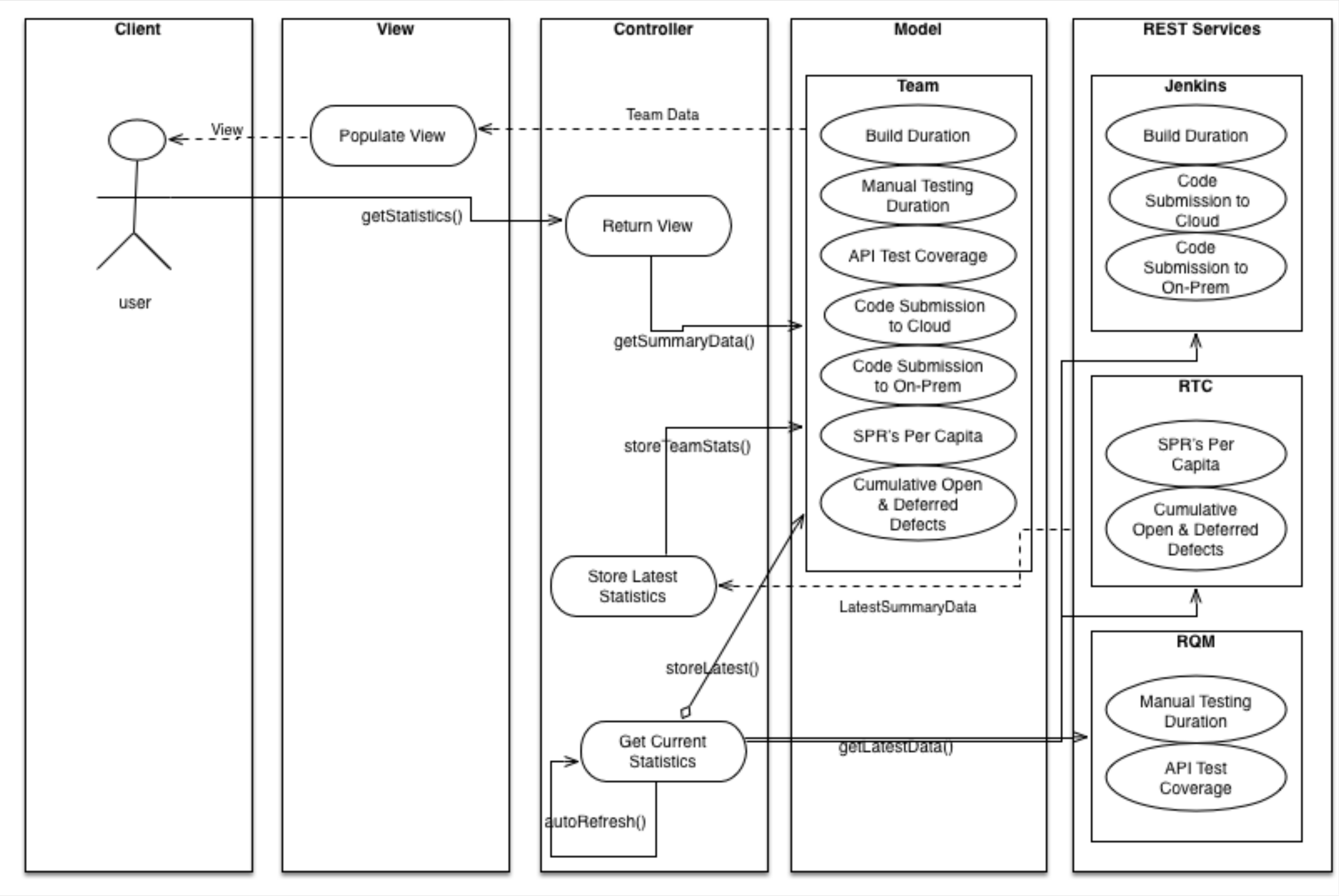
Continuous Delivery Commit Phase Dashboard

Shane Murphy

Challenge

- IBM Collaboration Solutions are to adopt CD for all development teams.
- ICS Director to report progress on set of CD Trnasformation Checkpoints and metrics regularly.
- CD is a pattern language used to automate and improve software delivery process.
  - Automated testing
  - Continuous Integration
  - Contiuous Deployment
- Provide information to team members, management and executives to compare against their goals.
- To display summary data relating to the continuous delivery.
  - Build Duration
  - Manual Testing Duration (automatable)
  - API Test Coverage
  - Manual Testing Duration (non-automatable)
  - Time from Final Code Submission to Production (Cloud)
  - Time from Final Code Submission to Production (On-Premises)
  - SPRs per capita
  - Cumulative Open & deferred defects

Design


Client
View
Controller
Model
REST Services
Team
Jenkins
Team Data
View
Populate View
Build Duration
Build Duration
Manual Testing Duration
Code Submission to Cloud
getStatistics()
Return View
API Test Coverage
Code Submission to On-Prem
user
Code Submission to Cloud
getSummaryData()
Code Submission to On-Prem
RTC
SPR's Per Capita
storeTeamStats()
SPR's Per Capita
Cumulative Open & Deferred Defects
Cumulative Open & Deferred Defects
Store Latest Statistics
LatestSummaryData
RQM
storeLatest()
Manual Testing Duration
Get Current Statistics
getLatestData()
API Test Coverage
autoRefresh()

Solution

- VirtualBox and Ubuntu used to host and run the RTC Server
- IntelliJ IDE
- MVC Architecture with Services Layer
- Implemented using Groovy & Grails
- Grails plugins used.
  - For example:
    - Spring Security
    - Google Charts
    - Quartz2
- Views use SiteMesh to Insert different data into one template used across the application

- D3.js and Google Charts API used to create the graphs
- MySQL Database for backend
- Cron Jobs update the local database to reflect RTC
- Database hosted on MAMP
- User Logins implemented with Spring Security and custom profiles
- Integrated with RTC, research into how to extract data from Jenkins/RQM undertaken. Sample data used.

Revision

- Spent more time trying to integrate with Jenkins/RQM than adding functionality

- Project entered a yellow status.

- Rescoped to focus on the commit phase in order to meet deadline.
  - Build Duration
  - Commit Phase Testing duration (automated)
  - SPRs opened per capita (general quality measure)
  - Cumulative open and deferred defects (technical debt)
  - User Login and personalisation
  - Use Mock data

Learning outcomes

- Groovy
- Grails
- Hibernate
- Quartz2
- Configuration
  - Servers
  - Databases
  - Virtual Machines
- RTC, RQM, Jenkins
- Insight into SDLC in a real world environment
- IBM Experience
  - IBM Culture
  - SDLC
  - Tools
  - Regular meetings with mentor