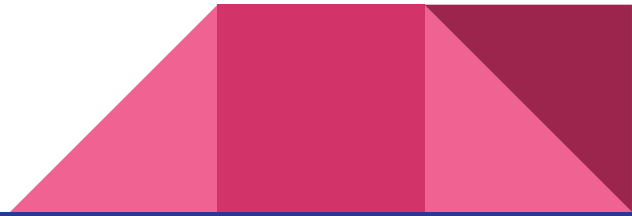
Communication System

Northeastern University
Managing Software Development
Instructor : Mike Weintraub
Team : Varad Choudhari
Saurabh Singh
Rutva Rajdev
Rohan Chitnis

System Functionality Achievement

The system supports below mentioned functionality -

- Modifying legacy codebase to extend the functionality. This includes -
  - Writing unit tests for achieving 85% coverage.
  - Removing code smells.
  - Making code testable.
- User Operations :
  - Create user.
  - View Details for a user.
  - Change user name.
  - Delete a user.
  - User login.



System Functionality Achievement (Contd.)

- Group Operations :
  - Create group.
  - View Details for a group.
  - Add user to group.
  - Remove user from group.
  - Change group name.
  - Delete a group.
- Send messages to User.
- Messages can be of any type : Text, Image, Video, Audio, Emoji.
- Send messages to Group. Included Reply-All function.

System Functionality Achievement (Contd.)

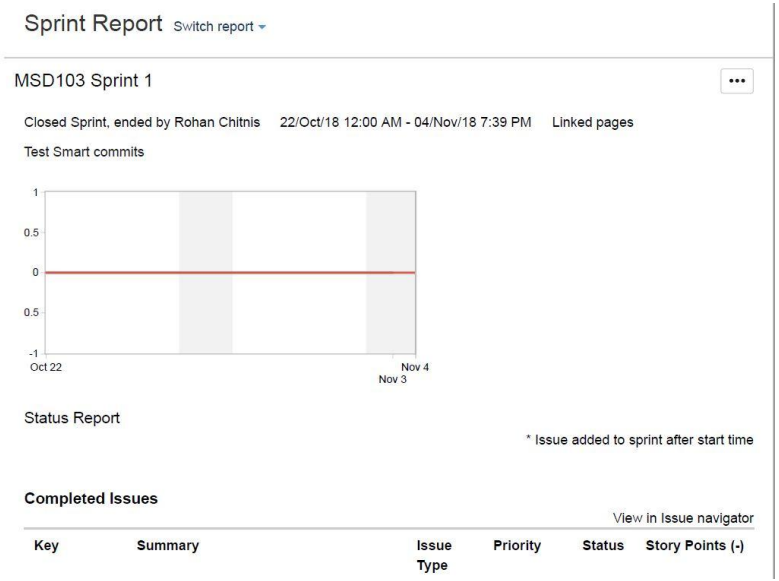
- User’s password is secured through AES encryption.
- Message Persistence using MongoDB.
- Implemented message Queue :
  - Offline users get all messages when they come online.
  - Same applies for group communication.
- Search messages :
  - Messages can be searched based on username, group name and timestamp.
- Subpoena :
  - Allows agencies to monitor live as well as archived chat between users and group.

System Functionality Achievement (Contd.)

- Recall messages :
  - Undelivered messages can be recalled by sender.
- CALEA compliance :
  - Messages are wrapped with IP address of the sender and receiver.
- Parental control feature :
  - Message server flags inappropriate content for users who have opted for parental control.
  - For communication in a group, inappropriate messages are flagged only for those participants who have parental control turned on.

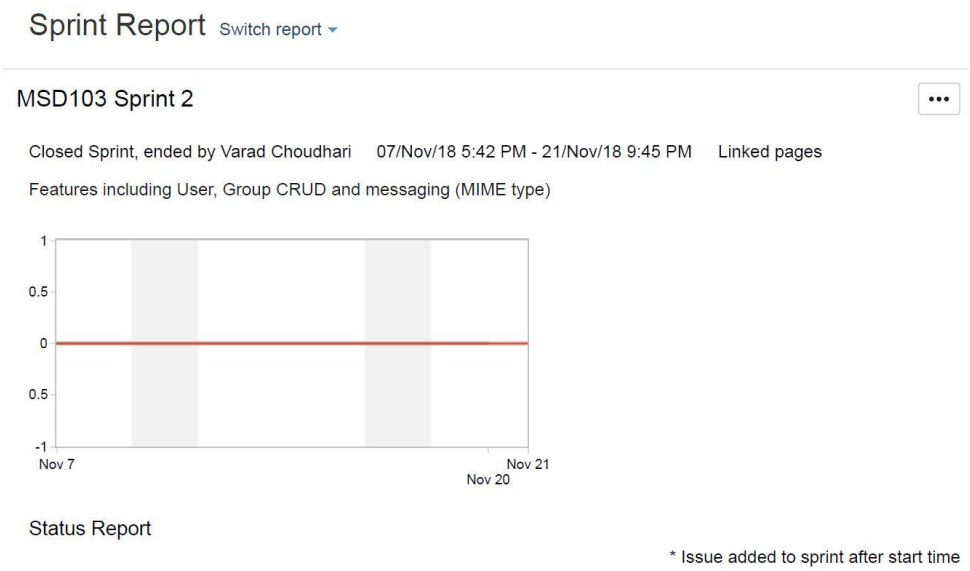
Sprint Reports

Sprint 1 :



Sprint Reports (Contd.)

Sprint 2 :



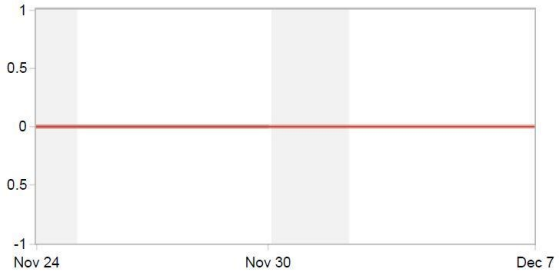
Sprint Reports (Contd.)

Sprint 3 :

Sprint Report Switch report

MSD103 Sprint 3 •••

Closed Sprint, ended by Varad Choudhari 24/Nov/18 10:22 PM - 07/Dec/18 6:49 PM Linked pages



Status Report

* Issue added to sprint after start time

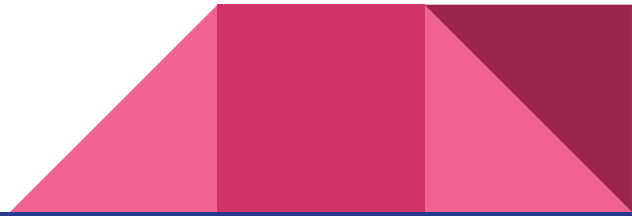
Development Tools

- Team used JIRA for tracking project.
  - Team worked in Agile sprints of 2 weeks duration.
  - Team followed Scrum events such as Sprint Planning, Sprint Review and Sprint Retrospective.
- Git was used for version control.
- Maven framework is used for managing project dependencies.
- Versions -
  - Java - 1.8
  - Maven - 3.1
  - Junit - 5
  - Jacoco - 0.8
  - Mockito - 1.9.5
  - Jackson - 2.9.7

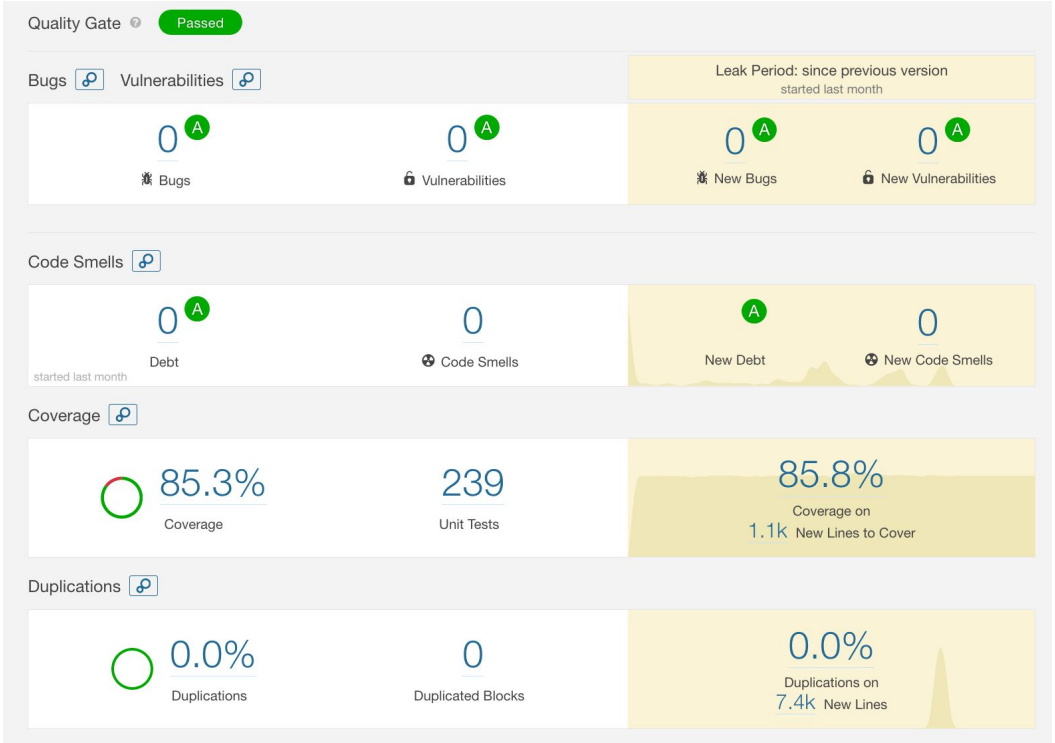


Development Tools (Contd.)

- Git smart commits.
- Jenkins for continuous integration.
- Slack :
  - For team communication.
  - Slackbot for informing build failures.
  - Slackbot for PRs in github.
  - Slackbot for logging system failures.
  - Slackbot for logging security concerns.
- Dynamic logging
  - Admin can turn on/off logging at any point of time.



Quality Assurance

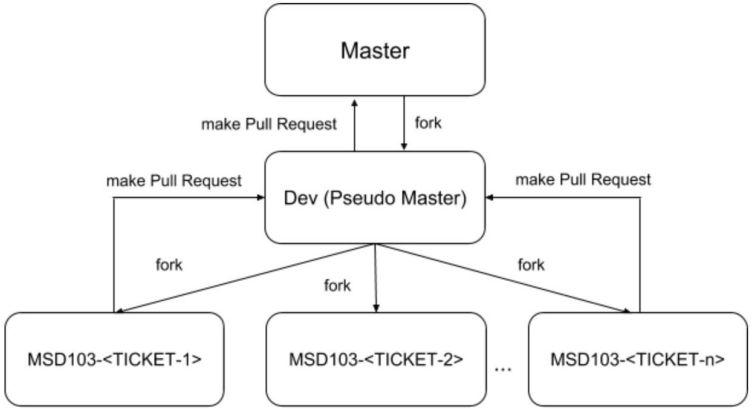


Team Performance

- Sprint 1 :
  - Team faced issues getting the legacy code base running and with git workflow which can be seen in JIRA and Git network diagram.
- Sprint 2 :
  - We understood team dynamics and working style.
  - Designed our custom git workflow.
  - Decided on what best practices we are going to follow.
  - Team understood the legacy code base well and could extend it with additional functionalities.
- Sprint 3 :
  - Last sprint was well coordinated and team achieved its maximum efficiency.

Process

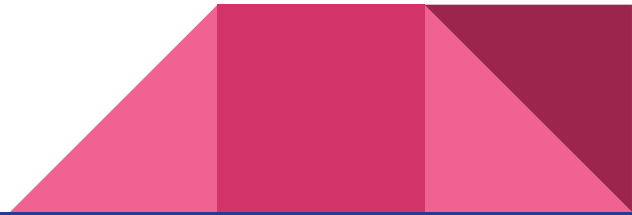
- Team followed coding best practices.
- Created custom git workflow -



Team 103 - Git Workflow

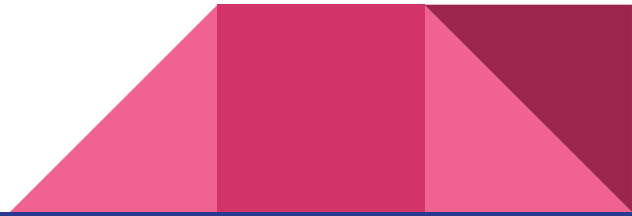
Process (Contd.)

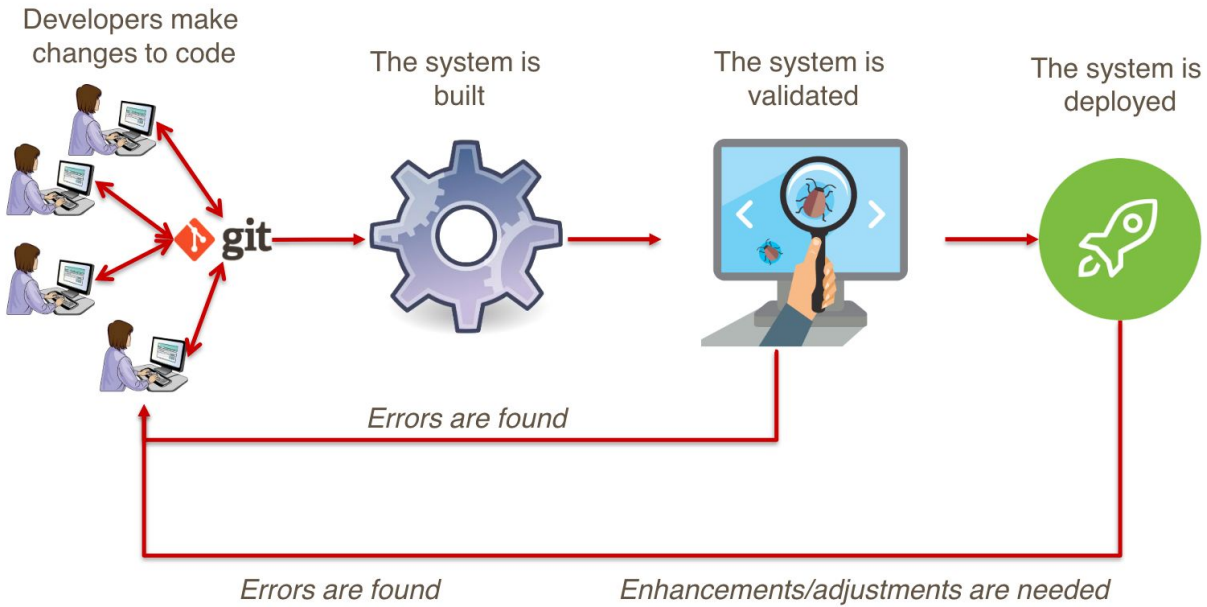
- Team created stories with sufficient description.
- Team used smart commits.
- Team communicated issues and blockers well ahead of time so that we had time to overcome those.
- Team had a good skill diversity.
- Team worked as self-organizing team to take up tasks during sprint and never over-committed.
- Team followed test-first approach and wrote tests for almost all testable functions and achieved 86% code coverage.



Process (Contd.)

- System admin made sure master branch is always up and running.
- This was achieved using automated jenkins build that ran after every “git push” and every “pull request”.
- The code is merged into master only if all unit tests and the quality gate passes certain threshold (85%).





Added from Alex Grob's presentation

Stakeholder Satisfaction

- Delivered all the promised functionality.
- Application is easy to use as it has menu options.

Future Development

- Upload data files to S3.
- Performance testing.
- Integration testing.