Northeastern University

CodeX

Plagiarism Detection Tool by Team-102

Northeastern University
CS 5500 - Managing Software Development
Spring, 2018 - Professors Annunziato and Weintraub

CodeX

System Functionality

- Initial Use Cases count : 9
- Implemented Use cases count : 14
- System Actors
  - Admin
    - Professor Approval
    - Usage Statistics
  - Professor
    - Create course and homework
    - Check Plagiarism and detailed report
  - Student
    - Enroll new courses
    - Upload python assignments(folders and file)

CodeX

Additional Significant System Functionalities

- Implementation of **four** sophisticated plagiarism detection strategies :
  1. Identifier and function renaming
  2. Structure match
  3. Context match
  4. Weighted polynomial
- Application of machine learning techniques for scaling the overall score generated by stated strategies.
- Snippet Generation
- Simple and Intuitive User Interface

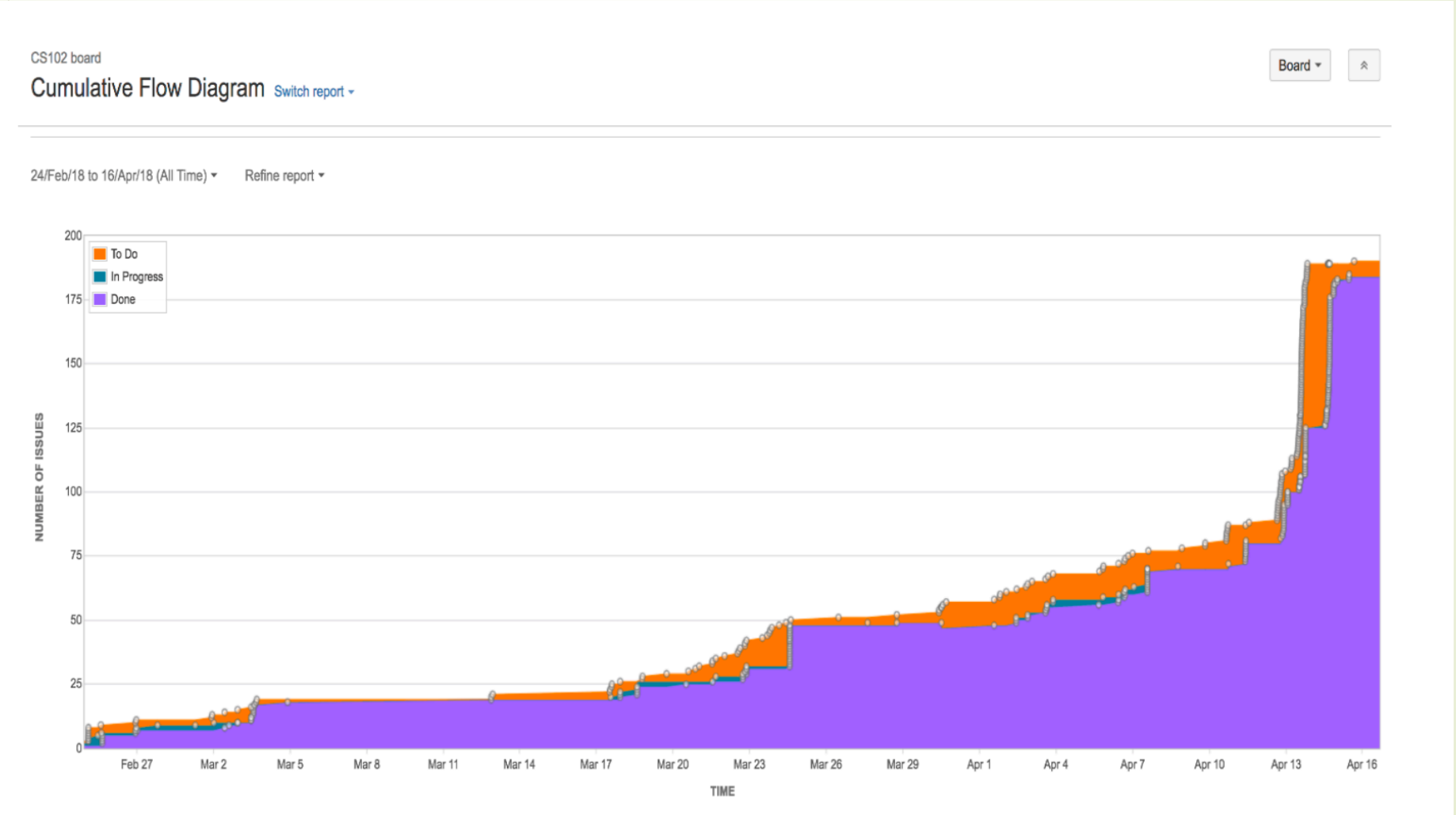
CodeX

Continuous Integration

- Integration of GitHub, Jenkins, Jira and Slack.
- Project repository master branch protection.
- Usage of smart commits.
- Continuous system quality check and maintenance via Jenkin+SonaQube pipeline.
- Code review by peers before merging to master.
- Automation of Slack messages and Email for each build status.

CodeX

Backlog Statistics



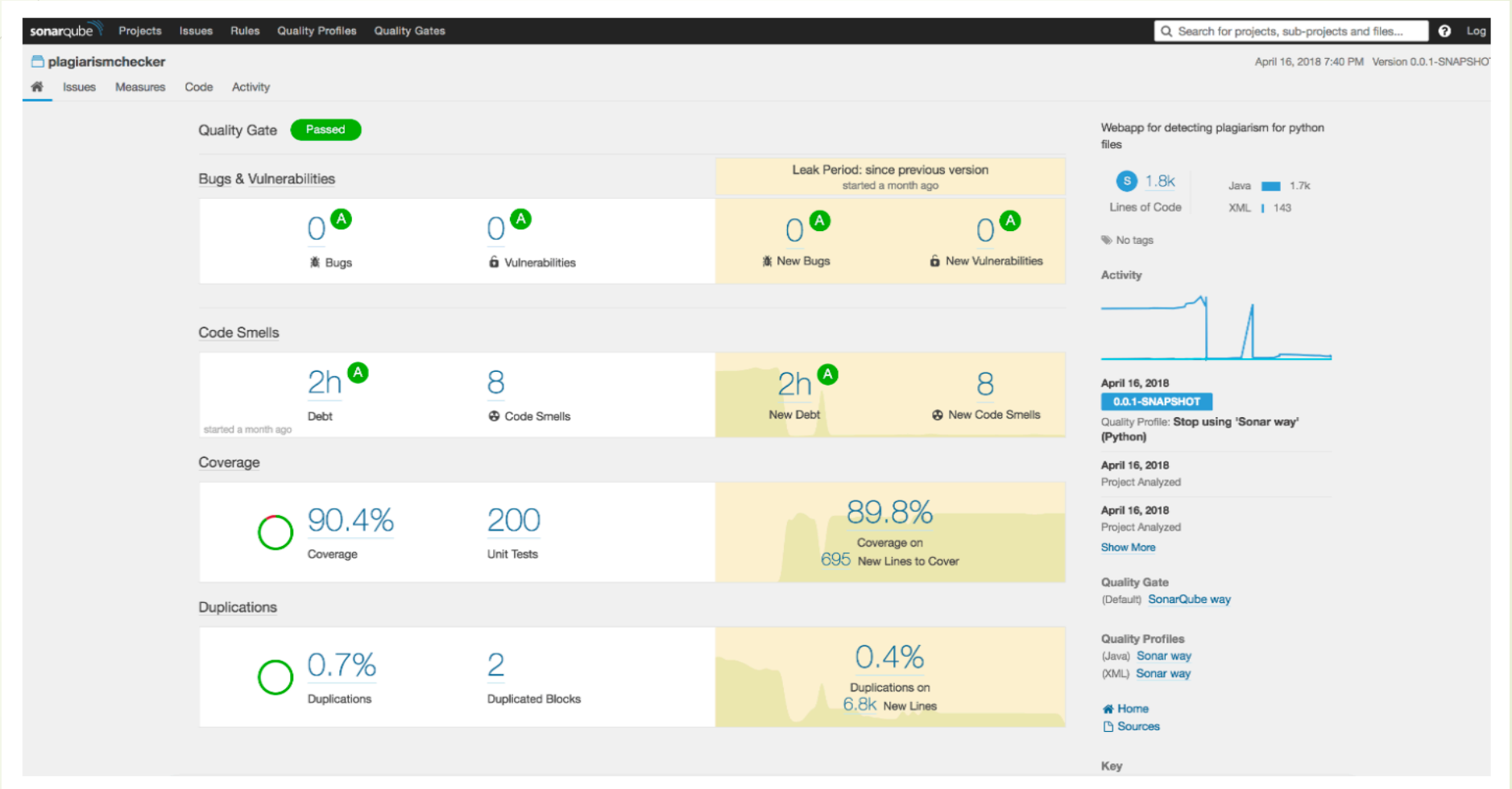
CodeX

Quality Job

- Test Driven Approach.
- Maintaining JIRA tickets and corresponding branches for each issue.
- Usage of Smart commits.
- Automating Jenkins pipeline to send message/emails to peers via slack.
- Enforcing peer review for pull request before merging to master.
- Code quality analysis through Sonar Lint and SonarQube.

CodeX

Quality Statistics



CodeX

Team Work

- Team co-ordination – well-structured and divided responsibility
- Flexibility – Team was able accommodate unexpected issues during development and deployment.
- Ownership -Reliable and responsible for individual tasks
- Timely Communication – weekly SCRUM meets to discuss status of the development and blockings if any
- Achieving goals of project working together.

CodeX

Agile Process

- Adopted agile methodology
- Development phase expanded over 3 sprints.
- New featured development discussed and assigned for each sprint
- Prioritized issues
- Thorough code review before merging into master.
- Focused on integration and deployment towards the end of each sprint.
- Embedded Design patterns to follow best coding practices.

Shortcomings

- Missed submitting first peer review on teammates – Set up calendar reminder for above for all other sprint reviews.
- Jenkins docker issues - Unresponsive Jenkins docker due to low system memory on AWS t2.micro server instance. Migrated from t2.micro to t2.small AWS instance.

Automating process

For every pull request the application builds on Jenkins, runs all test cases and checks for code quality on SonarQube before merging the branch into master.

Technology transfer

- Fork or clone this project using
  git clone *https://github.ccs.neu.edu/cs5500/team-102.git*
- Open the Project in any IDE of your choice
- Run *mvn clean install*.
- Run
  /plagiarismchecker/src/main/java/com/northeastern/msd/team102/plagiarismchecker/PlagiarismcheckerApplication.java
- To run on localhost: http://localhost:8080/#/login
- AWS link: http://codex.us-east-1.elasticbeanstalk.com/#/

Future Scope

- Plagiarism check for multiple languages.
- New plagiarism check strategies can be embedded easily.
- Email/Download plagiarism report for each student.

Thank you!