CS314 Software Engineering
Continuous Integration

Dave Matthews

Sprint 1 Submissions versus Reviews

- Only 1 team with perfect score
  - wrong aspect ratio, orientation
  - wrong tour or incomplete
  - argument handling
  - geographic coordinate handling
  - id column problems
  - submission configuration
  - program failures/exceptions

<table>
<thead>
<tr>
<th>Stories done</th>
<th>Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>
Sprint 2 rule change

- You can work on any of the classes you did not work on in Sprint 1.
- You can still consult on any code you wrote for Sprint 1, you just can’t change it.
- This should allow the team more flexibility to eliminate bottlenecks in their work.

Sprint 2

- simple CSV posted on piazza using Haversine formula
  - verify and fix before doing anything else
  - then rerun all four test problems to make sure it runs
- shortened Sprint 2 due to Spring Break
  - prioritize your stories
  - get as many stories completely done as possible
  - note anything not done in the Sprint 2 review or be penalized
Sprint 2 Domain Model for TripCo

Sprint 2 Planning from Retrospectives

• Finish 5 days or a week early
  – Buffer
  – Time to do more testing and fixing
• But is this Scrum?
## Using a storyboard

<table>
<thead>
<tr>
<th>Sprint Backlog</th>
<th>Ready</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>• User Stories</td>
<td>• Tasks</td>
<td>• Tasks</td>
<td>• User Stories, etc.</td>
</tr>
<tr>
<td>• Non-functional requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Defects</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Refine</td>
<td>• Well understood, no questions</td>
<td>• Assigned</td>
<td></td>
</tr>
<tr>
<td>• Prioritize</td>
<td>• Accurate estimates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Split user stories</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Ballpark estimates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Pull from Product Backlog during Spring Planning</td>
<td>• Pull from backlog after refinement</td>
<td>• Pull from ready when previous task completed</td>
<td>• Pull when tests/code merged or acceptance criteria met</td>
</tr>
</tbody>
</table>

## Midterm Review

- User Stories, Acceptance Criteria
- Domain Models
- Unit Test, JUnit
- Git/GitHub
- Scrum
## CMMI for Development Model

<table>
<thead>
<tr>
<th>Maturity</th>
<th>Process</th>
<th>Project</th>
<th>Engineering</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>• Organizational Performance</td>
<td>• Quantitative Project Management</td>
<td>• Causal Analysis and Resolution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>• Organizational Process</td>
<td>• Integrated Project Management</td>
<td>• Requirements Development</td>
<td>• Decision Analysis and</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>• Risk Management</td>
<td>• Technical Solution</td>
<td>Resolution</td>
</tr>
<tr>
<td>3</td>
<td>• Organizational Process</td>
<td></td>
<td>• Product Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Definition</td>
<td>• Requirements Management</td>
<td>• Verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Organizational Process</td>
<td>• Project Planning</td>
<td>• Validation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Focus</td>
<td>• Project Monitoring and Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Organizational Training</td>
<td>• Supplier Agreement Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>• Requirements Management</td>
<td>• Sprint Planning</td>
<td>• Configuration Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Project Planning</td>
<td>• Sprint Retrospective</td>
<td>• Measurement and Analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Project Monitoring and</td>
<td></td>
<td>• Process and Product Quality Assurance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>• Supplier Agreement Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Supplier Agreement</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Scrum

- **Product Backlog**
- **Sprint Planning**
- **Sprint Review**
- **Daily Scrum**
- **Sprint Retrospective**


For Scrum, visit [https://www.scrum.org/](https://www.scrum.org/)
Continuous Integration

- Software development practice where members of a team integrate their work frequently.
- Integrations are verified by automated build to detect errors quickly.
- Leads to significantly reduced integration problems.

Continuous Integration practices

- Maintain a single source repository.
- Automate the build.
- Make your build self testing.
- Everyone commits at least daily.
- Every commit builds/tests changes on integration machines.
- Fix broken builds immediately.
- Keep the build fast.
- Test is a clone of the production environment.
- Make it easy for everyone to get the latest executable.
- Everyone can see what’s happening.
- Automate deployment.
Continuous Integration benefits

- Immediate feedback on changes
- Find defects earlier, easier to fix while fresh
- Defect compounding mitigated
- Fewer bugs overall
- Work stays current, less technical debt
- Collaboration improved
- Progress can be measured
- Reduced Risk

Continuous Integration

- Slack
  - Waffle
  - GitHub
  - TravisCI
  - pom.xml
  - .travis.yml
**pom.xml**

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>edu.csu2017sp314.dtr99</groupId>
  <artifactId>tripco</artifactId>
  <packaging>jar</packaging>
  <version>0.2-sprint</version>
  <name>TripCo</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

**.travis.yml**

```yaml
language: java
jdk:
  - oraclejdk8
```
Repository file structure (preliminary)

pom.xml
.travis.yml

src/main/java/edu/csu2017sp314/dtr99/TripCo.java
src/main/java/edu/csu2017sp314/dtr99/Model/Model.java
src/main/java/edu/csu2017sp314/dtr99/View/View.java
src/main/java/edu/csu2017sp314/dtr99/Presenter/Presenter.java

src/test/java/edu/csu2017sp314/dtr99/TestTripCo.java
src/test/java/edu/csu2017sp314/dtr99/Model/TestModel.java
src/test/java/edu/csu2017sp314/dtr99/View/TestView.java
src/test/java/edu/csu2017sp314/dtr99/Presenter/TestPresenter.java

Instructions forthcoming