MIMANSA: Process Mining Software Repositories from Student Projects in an Undergraduate Software Engineering Course

Megha Mittal and Ashish Sureka

ICSE SEET 2014 Presentation
Presentation Outline

• Research Motivation & Aim
• Related work & Research Contributions
• Research Framework and Experimental Dataset
• Team Wiki
• Version Control System
• Issue Tracking System
• Conclusion
Presentation Outline

• Research Motivation & Aim
  • Related work & Research Contributions
  • Research Framework and Experimental Dataset
• Team Wiki
• Version Control System
• Issue Tracking System
• Conclusion
Software Engineering

• Software Engineering is a practice oriented and applied discipline.

• Its learning objective includes:
  - Learn fundamental technical concepts in SE courses.
  - Learn software development processes, teamwork and project management.
  - Exposure to popular SE tools.
In an educational domain, SE courses contain team-based semester long projects.

Artefacts like software requirement specification document (SRS, design document, project plans, test plans and source code) are produced at various milestones.
Research Motivation

• Course Instructor can easily assess and provide feedback on product and deliverables produced.

• Providing feedback on process and team-work is not straightforward.
• **Effective mechanism** is required for SE course instructors to:
  
  - gain visibility and insights on software development processes followed by student teams.
  
  - provide appropriate feedback on process improvement.

• Our motivation is to **develop tools and techniques** for solving problems encountered by SE course instructors.
Research Aim

• To investigate the application of process mining event log data generated from Wiki, VCS and ITS.

• To define new process quality metrics and visualization.

• Examine the application of existing process quality metrics.
Presentation Outline

☑ Research Motivation & Aim
• Related work & Research Contributions
• Research Methodology
• Experimental dataset
• Team Wiki
• Version Control System
• Issue Tracking System
• Conclusion
## Related Work

<table>
<thead>
<tr>
<th>Study</th>
<th>University</th>
<th>Repository</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glassy 2006 [2]</td>
<td>University of Montana</td>
<td>SVN</td>
<td>Mining problematic patterns, steadiness of progress and quality of commit messages</td>
</tr>
<tr>
<td>Jones 2010 [4]</td>
<td>Bloomsburg University</td>
<td>SVN</td>
<td>Aid in determining the accomplishments of each individual in a group programming project.</td>
</tr>
<tr>
<td>Robles 2013 [14]</td>
<td>Universidad Rey Juan Carlos</td>
<td>GIT VCS</td>
<td>Gathering software analytics data from programming assignments</td>
</tr>
</tbody>
</table>
Research Contributions

- Analyzing Wiki, VCS and ITS activity event logs from a process mining perspective.

- Visualizations and metrics providing feedback on various aspects such as:
  - workload distribution
  - consistency in contributions
  - quality of commit messages
  - efficiency of bug fixing process
Presentation Outline

✓ Research Motivation & Aim
✓ Related work & Research Contributions

• Research Framework and Experimental Dataset
• Team Wiki
• Version Control System
• Issue Tracking System
• Conclusion
- Research Framework and Experimental Dataset

**Research Framework**

- **ISSUE TRACKING**
  - BUGZILLA
  - MERCUIRAL OR GIT
  - VERSION CONTROL
  - BITBUCKET WIKI

- **REQUIREMENTS**
- **XML-RPC JSON-RPC**
- **MySQL RDBMS**

- **DATA EXTRACTION, PRE-PROCESSING**

- **ANALYTICS**

- **PROCESS MINING SOFTWARE REPOSITORIES**

1. Process Verification & Conformance
2. Development Activity
3. Actual Process Discovery
4. Team-Work Collaboration
5. Product - Process Correlation
Experimental Dataset

- Number of Students: 99
- Number of Teams: 19

- Total Commits:
  - Wiki: 1167
  - VCS: 925

- ITS:
  - Bugs Reported: 482
  - Bugs Resolved: 404
Presentation Outline

✓ Research Motivation & Aim
✓ Related work & Research Contributions
✓ Research Framework and Experimental Dataset

• Team Wiki
• Version Control System
• Issue Tracking System
• Conclusion
Team Wiki

• Collaborative tool that supports software development and maintenance process of a team project.

• Facilitates inter and intra team information exchange.

• Used by students for documentation of user stories.
User Story

- Primary development artefact of Extreme Programming Methodology.
- Defines functions required to provide and facilitate requirements management.
- Focuses on efficient interaction between the developers and customers.
- Captures the 'who', 'what' and 'why' of a requirement.
## Wiki Event Log

### File History for Bitbucket Repository:

<table>
<thead>
<tr>
<th>Author</th>
<th>Commit</th>
<th>Message</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c0e6f42</td>
<td>Placement of image in 7.</td>
<td>2013-09-01</td>
</tr>
<tr>
<td></td>
<td>0a65163</td>
<td>Added additional details for 21-23 and 25</td>
<td>2013-09-01</td>
</tr>
<tr>
<td></td>
<td>f526307</td>
<td>Added &quot;Additional details&quot; for User stories 17,18,19</td>
<td>2013-09-01</td>
</tr>
<tr>
<td></td>
<td>fa64147</td>
<td>Updated placing of text &quot;corresponding sketch&quot; in user story 3</td>
<td>2013-09-01</td>
</tr>
<tr>
<td></td>
<td>f7d6eb7</td>
<td>Added corresponding sketch for User story 3,6</td>
<td>2013-09-01</td>
</tr>
<tr>
<td></td>
<td>e623666</td>
<td>Added additional details to user stories (13-16) and 24</td>
<td>2013-09-01</td>
</tr>
</tbody>
</table>

- **Timestamp:** Date of the event
- **Commit ID:** Identifier of the commit
- **Message:** Description of the event
- **Author:** Person who committed the change

### Wiki Log:

- Bitbucket Repository: `https://bitbucket.org/iiit_deli_se/.../wiki/history/Home.md`
- Repository Details: `Repositories → Create`
RQ1: Study quality of commit messages.

RQ2: Study process followed by students during requirement documentation.

RQ3: Study uniformity of work load distribution in a team.
RQ1: Study quality of commit messages

<table>
<thead>
<tr>
<th>Quality</th>
<th>Example</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>1. Edited online.</td>
<td>Vague, not specific, only 2-3 terms, too general.</td>
</tr>
<tr>
<td></td>
<td>2. Wrote 4 user stories.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Minor fixes here and there.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. User Story-32</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1. Changes to story 8,9,10,14.</td>
<td>Specific but still not covering all (rationale, why and what) relevant</td>
</tr>
<tr>
<td></td>
<td>2. Edited some user stories (Priority heading removed).</td>
<td>details.</td>
</tr>
<tr>
<td></td>
<td>3. Edited 2nd user story.</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>1. Re added costs for 9-12, got removed due to commit conflict.</td>
<td>Short, concise, specific and explains the rationale, why and what.</td>
</tr>
<tr>
<td></td>
<td>2. FAQ page addition for easier usage of the application.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Added Title, Priority and Cost for User Stories 17,19.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Release of user story 1 and 2 added.</td>
<td></td>
</tr>
</tbody>
</table>
**RQ2: Study process followed by students during requirement documentation**

- Course project should evolve in an incremental and iterative fashion.

- Analysing the frequency of commits over time helps in understanding:
  - Consistency in the development activity of a project.
  - Near the deadline behaviour.
An incremental pattern.

A non-incremental pattern.
Graph Illustrating Significant Increase in Number of Commits Close to the Deadline

RQ2 - Near Deadline Behaviour

- Started early and finished well before the deadline.
- Though work in start and middle but sudden increase in last 2 days.
- Contributes 35% of work on the last day of deadline.
RQ3: Study uniformity of work load distribution in a team

- Software Development Process inherently requires teamwork with an equal distribution of work load.
- Uniform team work is required to develop the quality of team spirit amongst students.
RQ3 – Work Load Distribution

Contribution of Each Member of a Team in a Wiki
Presentation Outline

✓ Research Motivation & Aim
✓ Related work & Research Contributions
✓ Research Framework and Experimental Dataset
✓ Team Wiki
  • Version Control System
  • Issue Tracking System
  • Conclusion
Version Control System

- Records changes to a file or set of files over time.
- Benefits students by preparing them for the real life situations.
- Acts as a tool for instructor to monitor the development process and visualize team and individual contribution.
RQ4: Study uniformity of work of team developers across various components.

RQ5: Study effects of milestones on development activity.
RQ4: Study uniformity of work of team developers across various components.

- Analysing the contribution of individual developers in a team is very important.
- Helps the instructor in understanding the quality of team work for fair grading process.
- A project contains a modular structure that requires a collaborative effort from all members of a team.
RQ4 – Uniformity in Team Work

A peak at one axis shows expertise in that component but shows lack of uniform participation.

Member3 and Member4 show significant contribution in components “Authentication” and “Systems” respectively.

“Member1” worked least in the entire team. Equal work.

Participation of member2 equally spread.
RQ4 - Component and Developer Entropy

- We define 2 metrics to quantify two types of distribution (developer wise component and component wise developer).

- Proposed metrics are:

  **Developer-Component Entropy**

  \[
  H(c|d) = - \sum_{i=1}^{n} \sum_{j=1}^{m} p(c=j|d=i) \cdot \log_{m}(p(c=j|d=i))
  \]

  **Component-Developer Entropy**

  \[
  H(d|c) = - \sum_{j=1}^{m} \sum_{i=1}^{n} p(d=i|c=j) \cdot \log_{n}(p(d=i|c=j))
  \]
The entropy value can vary between 0 to 1.

1 signifies a perfectly uniform distribution of work in all components and equal contribution by all developers.

0 signifies no co-development in a component or collaboration by developers.

It serves as an indicator of extent of co-development at component level.
The developer entropy for member2 is 0.99.

Component entropy for component “Feedback” as 0.64.

Component entropy for component “System” is 0.88.
“Leave“ has component and developer entropy value in range 0-0.5

RQ5: Study effects of milestones on development activity.

• Development process of a software is an incremental process.

• Analyzing development activity of projects helps in understanding the change in behaviour of students near release dates.
RQ5 – Behaviour near Release Dates

Colour represents a developer. Member2 shows consistent behaviour. An increase in number of commits can be noticed. We infer that developers become more active near to the release of a deliverable.

Scatter Plot Representing the Commit Activity with Release Dates
Presentation Outline

✓ Research Motivation & Aim
✓ Related work & Research Contributions
✓ Research Framework and Experimental Dataset
✓ Team Wiki
✓ Version Control System
• Issue Tracking System
• Conclusion
Issue Tracking System

• Used during *software development and maintenance phase*.

• Software Maintenance and defect-fixing process requires a collaborative effort from testing and resolving team.

• Produces an event log that contains the information about the steps followed during bug-fixing process.
## ITS Event Log

### Bugzilla – Activity log for bug 292: User is able to sign up twice with the same data

<table>
<thead>
<tr>
<th>Who</th>
<th>When</th>
<th>What</th>
<th>Removed</th>
<th>Added</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2013-10-12 13:58:38 IST</td>
<td>Status</td>
<td>NEW</td>
<td>ASSIGNED</td>
</tr>
<tr>
<td></td>
<td>2013-11-25 15:34:51 IST</td>
<td>Assignee</td>
<td><a href="mailto:richa@iiitd.ac.in">richa@iiitd.ac.in</a></td>
<td><a href="mailto:richa@iiitd.ac.in">richa@iiitd.ac.in</a></td>
</tr>
<tr>
<td></td>
<td>2013-11-25 15:35:54 IST</td>
<td>Status</td>
<td>ASSIGNED</td>
<td>RESOLVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resolution</td>
<td></td>
<td>FIXED</td>
</tr>
</tbody>
</table>
Research Questions Answered

RQ6: Study efficiency of bug fixing process.
RQ7: Study variations in the time devoted to the repairing activity of bugs.
RQ8: Study weak and less functional components of a project.
RQ9: Study runtime process models of projects.
RQ10: Study inconsistencies between design time process model and runtime event log.
RQ6: Study efficiency of bug fixing process.

- Bug fixing performance efficiency can be measured using 2 performance indicators namely:

  - *Continuity* – that represents whether closing trend curve is smooth and without peaks or steps.

  - *Efficiency* – that represents whether the closing trend curve stays near to the opening trend curve.
RQ6 – Bugs Opening & Closing Trend

- Bug opening trend - cumulated number of bugs opened over time.
- Bug closing trend - cumulated number of bugs resolved over time.

An average behaviour can be noticed.

Growth of unresolved bugs increase with time.

Curve for the closing trend grows nearly as fast as curve for opening trend.
RQ6 - Bug Fixing Score

\[
\text{BugFixingScore} = \left( \frac{1}{n - i} \right) \sum_{i=n}^{i=n} \log(BO_i - BC_i) / \log(BO_i)
\]

- **BO** = Bugs opened till time instant \(i\)
- **BC** = Bugs closed till time instant \(i\)
- “\(i\)” → First time instant when \(BO > 0\)
- “\(n\)” → Maximum recorded time instant.
- Score of a project can vary between 0 and 1.
- 0 indicates a perfectly efficient behaviour where the bug fixing process is as fast as the bug opening process.
- 1 indicates an inefficient behaviour where there is no bug fixing though number of bugs opened continue to increase over time.
50% of the projects showed average bug fixing performance with **Bug Fixing Score** = [4.1-7.0].

16% of the teams showed an efficient bug fixing performance with **Bug Fixing Score** = [0.0-4.0].

<table>
<thead>
<tr>
<th>Score</th>
<th>Leave</th>
<th>Canteen, Grading, PhDContingency, Placement</th>
<th>Cafex, Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1-1.0</td>
<td>BTP, CourseReview, IFM</td>
<td>Alumni, Faculty, Hostel, Mess, Sports, PhDAdmission</td>
<td></td>
</tr>
<tr>
<td>4.1-7.0</td>
<td>Canteen, Grading, PhDContingency, Placement</td>
<td>Canteen, Grading, PhDContingency, Placement</td>
<td>Cafex, Issue</td>
</tr>
<tr>
<td>0.0-4.0</td>
<td>Coolefieds</td>
<td>Polling</td>
<td>Facility</td>
</tr>
</tbody>
</table>

**Bugs Reported**

<table>
<thead>
<tr>
<th>Bugs Reported</th>
<th>0-19</th>
<th>20-29</th>
<th>30-40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leave</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canteen, Grading, PhDContingency, Placement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cafex, Issue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTP, CourseReview, IFM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alumni, Faculty, Hostel, Mess, Sports, PhDAdmission</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coolefieds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facility</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RQ7: Study variations in the time devoted to the repairing activity of bugs.

- *Time taken to repair a bug (TTR)* reflects efficiency of developers during testing phase.

\[
TTR(b)(\text{inHours}) = date_{closed}b - date_{opened}b
\]

- TTR reflects the importance given by the development team to the testing and maintenance phase of their project.
High value of “Q3” represents irresponsible behaviour of the development team towards testing process.

Most of the teams shows an efficient behaviour in bug resolving.

Team “cafex” indicates an inefficient behaviour towards the bug resolving process with high Q3 value.

Box Plot Illustrating the Statistics for Time to Repair of Bugs Reported
RQ8: Study weak and less functional components of a project.

- Rigorous testing and inspection is required in each component.

- A component with majority of high priority bugs in a region indicates:
  - Its importance for a user and urgency to correct them.
  - Unsuccessful attempt of the development team in serving the main functionality of this component.
More bugs in “System” shows it is important but lacks in serving the main functionality of the product.

Less bugs in “Security” shows expertise and successful attempt of the project team in serving the need of the user.

Histogram Showing the Distribution of Bugs Priority Across Various Components of a Project
RQ9: Study runtime process models of projects.

- Analysed ITS event logs to study process models followed.
- A tool, *Disco* is used to obtain process map and other statistical information.

  - Pre-processed data is imported into Disco to discover the runtime process from the event log (generated during the progression of a bug).
A node represents an activity. Reported bugs = 476 with 10 different stages.

29% of bugs exhibited direct transition from fixed \(\rightarrow\) closed without getting "verified".

Shade and Thickness corresponds to the frequency with more frequent being dark and less frequent as light.

New \(\rightarrow\) Assigned, Assigned \(\rightarrow\) Fixed and New \(\rightarrow\) Fixed are the most frequent transitions.
RQ10: Study inconsistencies between design time process-model and runtime event log.

• We implemented 2 algorithms.

• First algorithm evaluates fitness metric to find the ratio of valid traces with total cases.
  – Fitness metric: depicts the compliance of observed process with the defined design time.

• Other algorithm indicates most frequent inconsistent transition and its frequency.
### Fitness Evaluation and Compliance Verification

<table>
<thead>
<tr>
<th></th>
<th>Alumni Mgmt</th>
<th>CourseReview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness Metric</td>
<td>0.384</td>
<td>0.793</td>
</tr>
<tr>
<td>Total unique Transitions</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>Total Inconsistent Transitions</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>Most Inconsistent Transition</td>
<td><strong>Resolved → Closed</strong></td>
<td><strong>New → Resolved</strong></td>
</tr>
<tr>
<td>Frequency of Most Inconsistent Transition</td>
<td>15</td>
<td>6</td>
</tr>
</tbody>
</table>

Value of Fitness Metric for “Alumni Mgmt" is low as compared to “CourseReview“.  

“For CourseReview“, only 8% of total unique transitions belongs to New → Resolved.  

“Alumni Mgmt“ shows 30% of inconsistent transitions.  
15/22 inconsistent transitions belongs to Resolved → Closed.
Presentation Outline

✓ Research Motivation & Aim
✓ Related work & Research Contributions
✓ Research Framework and Experimental Dataset
✓ Team Wiki
✓ Version Control System
✓ Issue Tracking System

• Conclusion
Conclusions

• Mining activity logs generated by Wiki, VCS and ITS presents useful insights to the course instructor.

• Visualizations and metrics obtained through activity log helps in characterizing:
  - variation in quality of commit messages
  - patterns indicating consistency in activity
  - distribution of workload.
  - variations in degree of process compliance and bug fixing quality.

• Proposed framework provides effective visibility to the instructor for improving academic outcome and learning methodology.


References


Thank You
Questions