Recent Trends in Software Technology

Seminar Presentation

Ayush Singh
163050014

Guide: Prof. Kameswari Chebrolu

Computer Science & Engineering
Indian Institute of Technology, Bombay
Introduction

- Need to improve the current teaching & learning scenario in classroom.
- Students loosing interest in traditional classroom studies.
- Difficult to track and manage large number of students.
- Aim to experiment, analyse and implement new methods to increase effectiveness of learning.
Flipped Courses

- Instructor uploads the video every week after each class.
- Students are required to watch video at their own pace before coming to class.
- Behaviour of the students watching video can be tracked and logged.
Methodology

“TrACE”[1] provided interface for video playing, discussion forum, instructor’s dashboard and keeping log of students. Main purpose of the study was to answer following question:

- “To what extent do students actually prepare for class by viewing the video in advance?"
- “To what extent student revisit the course content?"
- “Which viewing behavior co-relates with course performance metrics?"
- “How do student viewing behaviors vary with sociotechnical changes in class?"
Analysis

The graph shows students behaviour according to number of video lecture(x-axis) viewed.

(a) Average Punctuality in hours(y-axis) (b) Average student visits per video(y-axis)

Figure: Statistical analysis of students viewing behavior [1]
Lightweight Teams & Gamification

Team of 4-5 students were made to work together and healthy competition was organized among them.

**Teams**
- Moodle’s auto group function was used to form group ensuring every group has 2 girls.
- Seating plan was made for class so team can sit together.

**Gamification**
- Clicker quiz were organized for all teams.
- Stamps were given for achievements.
- Leader-board of teams were maintained.
- Tokens were given for asking question in doubt session.
Survey

<table>
<thead>
<tr>
<th>ID</th>
<th>Agreement Statement</th>
<th>Fall 2013 Early</th>
<th>Fall 2013 Late</th>
<th>Spring 2014 Early</th>
<th>Spring 2014 Late</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flipped</td>
<td>The flipped structure of this class is working well for me (I am learning a lot).</td>
<td>3.17</td>
<td>3.76</td>
<td>3.76</td>
<td>4.05</td>
<td>3.69</td>
</tr>
<tr>
<td>Team-1</td>
<td>Working with my team members to answer clicker questions helps me to learn.</td>
<td>4.54</td>
<td>4.38</td>
<td>4.52</td>
<td>4.60</td>
<td>4.51</td>
</tr>
<tr>
<td>Team-2</td>
<td>The team aspect of the course makes the course overall more engaging.</td>
<td>4.11</td>
<td>3.96</td>
<td>4.22</td>
<td>4.28</td>
<td>4.14</td>
</tr>
<tr>
<td>Team-3</td>
<td>I like being part of a team and sitting with my team each week.</td>
<td>4.28</td>
<td>4.44</td>
<td>4.26</td>
<td>4.48</td>
<td>4.37</td>
</tr>
<tr>
<td>Stamp-1</td>
<td>Stamps provide a good incentive to get me to put more effort into this course.</td>
<td>3.26</td>
<td>3.07</td>
<td>4.00</td>
<td>3.75</td>
<td>3.52</td>
</tr>
<tr>
<td>Stamp-2</td>
<td>The leaderboard on Moodle showing which team has the most stamps adds to my desire to earn more stamps.</td>
<td>3.47</td>
<td>3.04</td>
<td>3.75</td>
<td>3.37</td>
<td>3.41</td>
</tr>
<tr>
<td>Stamp-3</td>
<td>The addition of 1% to my final grade for every 10 stamps earned adds to my desire to earn more stamps.</td>
<td>3.53</td>
<td></td>
<td></td>
<td>3.91</td>
<td>3.72</td>
</tr>
<tr>
<td>Token-1</td>
<td>The use of tokens makes me try hard to figure out the lab with my partner.</td>
<td>3.54</td>
<td>3.07</td>
<td>3.72</td>
<td>3.37</td>
<td>3.38</td>
</tr>
<tr>
<td>Token-2</td>
<td>The tokens are fun and cool.</td>
<td>2.65</td>
<td>2.80</td>
<td>3.56</td>
<td>3.36</td>
<td>3.02</td>
</tr>
<tr>
<td>Token-3</td>
<td>Because of the tokens I am becoming a more self-reliant learner.</td>
<td>2.85</td>
<td>2.86</td>
<td>3.47</td>
<td>3.09</td>
<td>3.03</td>
</tr>
<tr>
<td>Token-4</td>
<td>The potential loss of a token has discouraged me from asking for help, even when I really need it.</td>
<td></td>
<td></td>
<td></td>
<td>3.93</td>
<td>3.72</td>
</tr>
<tr>
<td>Overall</td>
<td>Because of this course, I am likely to take another programming course.</td>
<td>3.89</td>
<td></td>
<td></td>
<td>3.95</td>
<td>3.92</td>
</tr>
</tbody>
</table>

Figure: Survey Results Average scores [2]
Results

Student from this course who took other programming course performed better than who didn’t took this course.

Figure: Post Course Performance [2]
Pop Quizzes

- Surprise quiz taken by instructor to get feedback from student.
- Helps student to keep up with the updated course content.
- No need to factor attendance in final grading.
- Generate self awareness in student to study.
Methodology

In basic CS1 Course out of 9 section, pop quiz was taken for 6 and no quiz for 3 section. Course comprises of students from different academic year and majors.

<table>
<thead>
<tr>
<th>Major</th>
<th>No Quiz</th>
<th>Pop Quiz</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
</tr>
<tr>
<td>Freshmen</td>
<td>23</td>
<td>32.9%</td>
<td>60</td>
</tr>
<tr>
<td>Sophomores</td>
<td>22</td>
<td>31.4%</td>
<td>40</td>
</tr>
<tr>
<td>Juniors</td>
<td>17</td>
<td>24.3%</td>
<td>38</td>
</tr>
<tr>
<td>Seniors</td>
<td>8</td>
<td>11.4%</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>100.0%</td>
<td>149</td>
</tr>
</tbody>
</table>

**Figure:** Distribution of student by academic year [3]
These are the percentage of students getting their final grades [3].

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>≥B</th>
<th>≥C</th>
<th>≥D</th>
<th>F</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pop Quiz</td>
<td>33.6%</td>
<td>60.4%</td>
<td>79.2%</td>
<td>87.3%</td>
<td>5.4%</td>
<td>7.4%</td>
</tr>
<tr>
<td>No Quiz</td>
<td>25.7%</td>
<td>58.6%</td>
<td>78.6%</td>
<td>84.3%</td>
<td>4.3%</td>
<td>11.4%</td>
</tr>
</tbody>
</table>

**Figure:** Grade earned by all students

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>≥B</th>
<th>≥C</th>
<th>≥D</th>
<th>F</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FR &amp; SO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pop Quiz</td>
<td>30.0%</td>
<td>53.0%</td>
<td>72.0%</td>
<td>84.0%</td>
<td>8.0%</td>
<td>8.0%</td>
</tr>
<tr>
<td>No Quiz</td>
<td>24.4%</td>
<td>57.8%</td>
<td>77.8%</td>
<td>82.2%</td>
<td>4.4%</td>
<td>13.3%</td>
</tr>
</tbody>
</table>

**Figure:** Grade earned by underclassmen

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>≥B</th>
<th>≥C</th>
<th>≥D</th>
<th>F</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JR &amp; SR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pop Quiz</td>
<td>40.8%</td>
<td>75.5%</td>
<td>93.9%</td>
<td>93.9%</td>
<td>0.0%</td>
<td>6.1%</td>
</tr>
<tr>
<td>No Quiz</td>
<td>28.0%</td>
<td>60.0%</td>
<td>80.0%</td>
<td>88.0%</td>
<td>4.0%</td>
<td>8.0%</td>
</tr>
</tbody>
</table>

**Figure:** Grade earned by upperclassmen
Development Techniques & Tools

- Proper implementation of method as software application.
- Development process should be done as per the requirement, with optimal resource and time.
- Tools should be used if present, instead of doing everything from scratch.
Android Performance

- Androids performance should be kept in mind when working on android app development.
- Even a small issue can cause a laggy animation which can be noticed by anyone.

Rendering
Process of conversion of code to an image (or UI in this case) is called rendering. CPU and GPU work together to draw and display image on screen.
Android Performance

Figure: Android’s Rendering pipeline with common problem & solution [4]
Android Performance

Computation

- Profiling methods which are taking time.
- ‘Trace View’ built tool in android studio.
- Reduce cost by using proper technique like batching, caching or choosing proper data structure.

Memory management

- Android has its own garbage collector which is invoked when app uses all its memory quota.
- Higher frequency of garbage collector call reduce performance.
- Avoid memory leak and memory churn.
Google Analytics for Android

- Google Analytics track our app data.
- Shows real time information about app usage.
- Does screen tracking and provide response time of activity.
- Can dynamically update app without the need for redeployment.

**Figure:** Asynchronous Programming Model for Requesting Services
A/B Testing

- Process of comparison of two versions of an application to get which is more liked by users.

![A/B Testing Diagram](image)

**Figure: A/B Testing of an application [5]**

- Both versions should be slightly different but noticeable.
Steps to be followed during A/B testing:

Policy & Ethics: We have to see how experiment will effect our user.

Choosing metric: The metric should be properly defined and all mathematical process should be clear that one needs to do.

Designing Experiments: Proper speculation should be done.

Analysing result: Check result and decide whether version should be implemented or not.
Tooling & Automation

- Use of proper environment to save development time.
- Effective use of plug-ins and API instead of making it from scratch.
- **Build Tools** Program that automates intermediate process of developing and execution.
- Feature Build tool should have:
  - **Fast**: Changes should be seen quickly.
  - **Extensible**: Supports custom functionality
  - **Community Driven**: Should be used by more no. of people.
Conclusion & Future Work

- We found from these research papers that these methods do work and gives significant result on implementation.
- ‘SAFE’ is also a similar project on which I have to work on my MTP.
- My part will be related to android development, so I can apply all the learning which I have done in my seminar duration to ‘SAFE’ app.
References


Thank You...