Development Projects: Lifecycle and Lessons Learnt

Kameswari Chebrolu
Department of CSE, IIT Bombay

Collaborators: Bhaskar, Sriram and many UG/PG students over the years
Lifecycle

1. Idea
2. Execution
3. Field-trial
4. Refinement
5. Wide-spread adoption
Lifecycle

• Idea:
  – Pain point: Poor engineering education in India
  – Can technology help address the same?

• Execution
• Field-trial
• Refinement
• Wide-spread adoption
Background

State of Engineering education in India

• 3,345 engineering colleges producing 1.5 million engineering students per year
• Majority are unemployed or under-employed
• Huge retraining cost in terms of money and time
• Poor workforce to conduct research or start ventures
Germ of an Idea

• Quality material in the form of videos hosted on server
  – Inspiration: Khan academy; do something similar for engineering education
  – Easier to follow than textbooks; more readily available
  – Final avatar: Multimedia Book

• Teachers use multimedia books to conduct flipped classes
  – Reduces their teaching load

• End Result: BodhiTree
  – Three Interfaces: Content-developers, Instructors, students
The flipped classroom inverts traditional teaching methods, delivering instruction online outside of class and moving “homework” into the classroom.

**THE INVERSION**

**The Traditional Classroom**
Teacher’s Role: Sage on the Stage

**LECTURE TODAY**
Reading and questions due tomorrow

**Homework**

**The Flipped Classroom**
Teacher’s Role: Guide on the Side

**ACTIVITY TODAY**
WATCH lecture online tonight!

---

Fig. From http://www.knewton.com/flipped-classroom/
Multimedia Book

- Content-developers create multi-media books
  - Organization: chapters/sections
  - Section explains a concept
    - Made up of videos, quizzes, animations, slides, virtual labs, reference docs etc
- Objective: Mimick in-class room setting
  - Interactive, short, exciting videos
- Misc: Discussion forum, Progress tracking, Chat
  - Across students following the book
Flipped Class

• Traditional textbooks replaced by multi-media books

• Instructors provide a flipped class offering in which students enrol (via BodhiTree)
  – Pull content from multiple multimedia books and assemble a course specific book
  – Particularly important where syllabus varies across colleges
  – Track students progress (who watched videos, quiz performance etc)
Flipped Class

• BodhiTree supports Moodle + Piazza style features: Share midsem/final marks, email support, facilitate discussion, chat support

• Lab Report Grading: Form based exam which eases manual grading for TAs
Related Work

• NPTEL/CDEEP: Studio/classroom video recordings
  – Focus: Local dissemination, mostly students (CDs, youtube, IITB server)

• MOOCs like Coursera, edX
  – Focus: World-wide student enrolments

• Our Focus:
  – Multimedia “Interactive” Books for use in flipped classes
  – Focus: local instructors in addition to students
Lifecycle

• Idea:

• **Execution:**

• Field-trial

• Refinement

• Wide-spread adoption
Framework

• Django/python/react.js (web development framework tools); android (app version)
• Scalability: memcached/Redis, code-optimization, database query optimization; affinity scheduling
• Network and server performance logs: ngnix logs, Munin, collectl, htop etc
• ML based user models derived from logs
  – For scalability tests via jmeter
Demo
Lifecycle

- Idea:
- Execution:
- **Field-trials**
- Refinement
- Wide-spread adoption
Usage So Far

• Touched 13000+ teachers/students so far
• Within IITB: Conducted 9 flipped classes in CSE
  – 6 UG courses and 3 PG courses (70-100 students per course) by 3 faculty
    • Subjects: Computer Networks (UG), Computer Architecture (UG); Wireless Networks (PG) and Software Lab (PG)
• BodhiTree being used by two other faculty from other departments (CHEM and IEOR)
• T10kt (teach 10k teachers) Workshop on “Computer Networks”
  – Covered 9000 engineering college teachers
  – One month online activity on BodhiTree followed by 5 day face-to-face interaction

• Two QEEE (Quality Enhancement in Engineering Education) Workshops on “Computer Networks” and “Computer Architecture”
  – Covered a total of 3000 students

• Expressed Interest (future use):
  – Faculty in Math department for Maharashtra wide math teacher training
  – Faculty in IDC for an India wide design course offerings
  – Few start-ups (life skills development)
Sample Feedback

- CS224 Spring 2015 and Teacher Training Workshop
- Survey Sample Size: 63 students; 225 teacher
60% better understanding
12% decreases understanding

Increase in Load:
IO: 61% No increase
TW: 25% No increase

IP: 70% found it helps exam preparedness
Sample Feedback

- CS224 (2nd year UG) Spring 2015 Offering
- Survey Sample Size: 63 students

- 83% - 93% preferred flipped mode of teaching to traditional teaching
- Average class marks increased from 56 (traditional teaching) to 69 (flipped teaching) (similar exam difficulty)
## Traditional vs Flipped

<table>
<thead>
<tr>
<th>Offering</th>
<th>Size</th>
<th>Max</th>
<th>Avg</th>
<th>AAs</th>
<th>ABs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring-2009 (Traditional)</td>
<td>56</td>
<td>79</td>
<td>50</td>
<td>[16%, 88]</td>
<td>[11%,76]</td>
</tr>
<tr>
<td>Spring-2012 (Traditional)</td>
<td>72</td>
<td>85</td>
<td>56</td>
<td>[10%,90]</td>
<td>[14%,81]</td>
</tr>
<tr>
<td>Autumn-2013 (Flipped)</td>
<td>94</td>
<td>93</td>
<td>64</td>
<td>[13%,86]</td>
<td>[29%,75]</td>
</tr>
<tr>
<td>Spring-2015 (Flipped)</td>
<td>96</td>
<td>92</td>
<td>69</td>
<td>[11%, 92]</td>
<td>[24%, 84]</td>
</tr>
</tbody>
</table>
Lifecycle

• Idea:
• Execution:
• Field-trials
• Refinement (personal experience, surveys, suggestion box)
• Wide-spread adoption
**Drawback**

- What motivates an **average** student to learn?
  - High quality instruction?
  - Good job prospects?

- **Students care about “certification” that fetch jobs**

- **Three Requirements:**
  - Quality instruction should be part of syllabus for certification
  - Certification process (exams) should test understanding not rote memorization
  - Both should scale to lakhs of students
SAFE: Smart, Authenticated, Fast Exams

- Idea Origins: Ease conduction of tutorial quizzes

1. Download SAFE app
2. Authenticate
3. Download exam
4. Attempt exam
5. Submit
Features

• **Smart, Fast**: use of smart devices + WiFi
  – In most situations: use candidates’ own devices (BYOD model)
  – Quick results (auto-grading)

• **Secure, Authenticated** (cheating free): key to any meaningful exam
  – Achieving ‘scalable’, and ‘secure’ non-trivial
  – Extensive logging, Use of visual/manual password, Bluetooth maps, per-exam app, random seating with phone swap etc
  – Need to address WiFi bottleneck
Challenges

• **Self Cheating:** Student alone cheats
  – Resources could be Internet/friends via phone, chat, bluetooth etc

• **Impersonation:** Student’s friend takes the exam on her behalf.
  – Student/Friend, both in class, give exam for each other.
  – A Friend in class gives exam for both.
  – Outside Friend with access to resources gives exam for student.
Challenges

• Means of cheating:
  – Compromise app (reverse engineer)
  – Compromise OS
  – Exploit bugs in app/OS
  – Pass on authentication/Quiz info to friend who uses a valid app and OS.
Features

- **Smart, Fast**: use of smart devices + WiFi
  - In most situations: use candidates’ own devices (BYOD model)
  - Quick results (auto-grading)
- **Secure, Authenticated** (cheating free): key to any meaningful exam
  - Achieving ‘scalable’, and ‘secure’ non-trivial
  - Extensive logging, Use of visual/manual password, Bluetooth maps, per-exam app, random seating with phone swap etc
  - Need to address WiFi bottleneck
Features

• Low cost; Low maintenance; Easy setup, Highly scalable

• Usage: Classroom quizzes, Company placement tests, Admission tests, Course certification
## Related Work

<table>
<thead>
<tr>
<th></th>
<th>safeexambrowser.org</th>
<th>respondus.com</th>
<th>42gears.com</th>
<th>Pearson VUE/TCS</th>
<th>SAFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique</td>
<td>Lock-down browser on PC</td>
<td>Lock-down browser on win/mac</td>
<td>Lock-down App</td>
<td>PCs, server, camera, in exam centre</td>
<td>App with lock-down++</td>
</tr>
<tr>
<td>Smart-device</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes ➔ portable</td>
</tr>
<tr>
<td>BYOD</td>
<td>Yes</td>
<td>Only win/mac</td>
<td>No</td>
<td>No</td>
<td>Yes ➔ low mgt</td>
</tr>
<tr>
<td>Secure exam</td>
<td>Incomplete</td>
<td>Incomplete</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Test banks</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Not yet</td>
</tr>
<tr>
<td>High stake exams</td>
<td>No</td>
<td>Not yet</td>
<td>No</td>
<td>Yes</td>
<td>Yes, at low cost</td>
</tr>
</tbody>
</table>
Lifecycle

• Idea:
• Execution:
• Field-trials
• Refinement
• Wide-spread adoption
Execution and Field Trials

- Django/Python Server (older version: PHP), android app
  - Android coding was the toughest
    - Not easy to achieve required level of security
    - Need to tackle a variety of versions
- WiFi Scalability: crowdsourced testing framework (another android app)
Usage

• Used for “real” quizzes in 5 courses so far (4 in CSE, 1 in CHEM)

• Expressed Interest: Companies and other faculty in IITB
Original Problem: Solution Adoption

• Two Tools: BodhiTree and SAFE
• Adoption seems the toughest hurdle to cross; but we are on it
• Current Plan (seem ever-changing :-)
  – Three players in our ecosystem: Industry, Content-developers and academia (colleges and students)
  – Players interface via a section-8 non-for profit (or) for-profit company
Steps

1. Syllabus and Content Creation:
   – Focus on 5-6 core subjects in CS/EE (to begin with)
   – Content developers create multi-media books with input from Industry (that meets industry needs)

2. Outreach:
   – Company will work on adoption of these multi-media books
   – Targets both colleges and self-learning students
3. Examination:

- Company will conduct high-quality exams pan India on specific dates (university model)
  - Exams set by content developers and other competent teachers
  - Use SAFE, local colleges and teachers to scale
  - Objective exam performance as filter for subjective exam eligibility
4. Certification: Company provides certification
   – Per course or Per module (series of courses)
   – Grade based
   – Can work with colleges to include student performance in college transcript
     • Example: Company 50%: Local-evaluation: 50%
     • Avoids students having to give two exams

5. Placement Cell: Company runs a placement cell
   – Put qualified students in touch with industry
Lessons Learnt: What to expect?

Major Hurdles in our system:

- Project spans many years
- Moving student population (lot of churn)
  - Most work 1 sem; some work for a year, rarely more than a year
  - Limited success with project staff
• Inexperienced coders
  – Inefficient code; buggy code, poor documentation, no coding standards
• Limited work hours
  • RnD: 6 hrs a week; BTP: 6-12hrs a week, MTP: 20-30 hrs a week
• End Users: Unforgiving or Clueless
  – Polished/bug-free product, User-friendly
  – Heterogeneity: Different devices, OS, browsers, Internet speeds
Lessons Learnt: Take Aways

1. You (Faculty i.e.) can do it without knowing to write or read a single line of code 😊
   But need to ask the right questions and do detailed testing
   - Code-logic needs to be captured via flow-charts, pseudo code, state diagrams etc
   - Quantify code execution details
     • How many database queries?, Number of requests generated?
       Execution time? Response time?
   - Devise all corner cases and test
2. **Invest in a software management tool (cannot emphasize its importance)**
   - We use a tool from Facebook’s called Phabricator
   - Wrapper around version control (Git, Bitbucket etc)
   - Imposes a coding standard
   - Permits peer-code review; can push only after approval
   - Support for unit-testing (painful to write but useful in long run)
   - Tracks bugs, assign tasks, chatroom, wiki,
3. Attention to scalability and logging structure from beginning

– We are trying to use a variety of tools to quantify code performance during coding
  • Database queries, execution time, no-of-threads, memory consumption etc

– For debugging problems or improving system, server/client logs during real-use are essential
4. Expect to rewrite code from scratch after trials
5. Expect to abandon some modules
6. Publications are hard (not impossible)
   – Difficult to sell novelty (work is more application-oriented)
   – Need to find newer avenues; takes time to get to know the community and expectations
   – Tradeoff: Work for adoption or publication?
7. Wider Adoption: Plan from beginning; involve all players in the ecosystem
Other Education Themed Projects

• File Sharing App: Share files (pdfs, doc, code etc) among a group of smartphone users
  – Usage: Classroom setting, conference talk settings
  – Sharing based on multicast over WiFi
  – Tough problem to implement: WiFi scalability, smartphone/AP limitations
• **Mic App:** Use smartphone as Mic in a large room/auditorium (Bhaskar’s)
  – Speaker with mic; audience’s voice cannot be heard
  – Local skype like call between audience and speaker via WiFi; plays out on speaker’s mic
  – Echo cancellation, timing issues

• **Doubts Clear App:**
  – Audience use smartphone to pose questions
  – Rest audience upvote questions
  – Speaker takes up the most popular questions
  – Useful in large classes and conference talk settings
Summary

• Development projects use current technology to solve local pain-points
  – Can give lot of satisfaction: Feels great to see work being used in practice
  – Can be technically challenging as well
• Lengthy life-cycle full of hurdles
  – Technical hurdles easier to cross, many others out of your control
• Much grateful and big thanks to the amazing set of students who made all this possible.
Thank You

• Faculty: Please consider adopting the tools in your classes
  – Students: Please cooperate and help us build as well as experiment with new ideas 😊

• Faculty: Please consider developing multi-media books
  – Students: Please help us improve such books via feedback

• Rest assured, we will make the task easy and enjoyable