Exploring the Use of Crowdsourcing to Support Empirical Studies in Software Engineering

Kathryn T. Stolee & Sebastian Elbaum

September 16, 2010
Recruiting the right type and number of users for empirical studies in software engineering is hard.
Known Issue

Recruiting the right type and number of users for empirical studies in software engineering is hard.

Possible Solutions:

- Use fewer participants of the right type
  - Limits generalizability to larger groups
Introduction

Known Issue

Recruiting the right type and number of users for empirical studies in software engineering is hard.

Possible Solutions:

- Use fewer participants of the right type
  - Limits generalizability to larger groups
- Relax requirements for participation
  - Limits generalizability to target population
Introduction

Known Issue

Recruiting the right type and number of users for empirical studies in software engineering is hard.

Possible Solutions:

- Use fewer participants of the right type
  - Limits generalizability to larger groups
- Relax requirements for participation
  - Limits generalizability to target population
- Crowdsource the study
Crowdsourcing

Leveraging a global community of users with different talents and backgrounds to help perform a task that would not be feasible without a mass of people behind it.
Crowdsourcing Services (examples)

Companies with hard problems connect with people interested in solving. 1,000+ problems, 200,000+ solvers
Crowdsourcing Services (examples)

- Companies with hard problems connect with people interested in solving. 1,000+ problems, 200,000+ solvers
- Photographers collect with people who need stock photography. 3,000,000+ members
Crowdsourcing Services (examples)

Companies with hard problems connect with people interested in solving. 1,000+ problems, 200,000+ solvers

Photographers collect with people who need stock photography. 3,000,000+ members

Companies with scientific problems connect with retired scientists. 1,000+ companies, 5,000+ scientists
Crowdsourcing Services (examples)

- Companies with hard problems connect with people interested in solving. 1,000+ problems, 200,000+ solvers
- Photographers collect with people who need stock photography. 3,000,000+ members
- Companies with scientific problems connect with retired scientists. 1,000+ companies, 5,000+ scientists
- People with many small tasks connect with scalable workforce. 100,000+ tasks, 100,000+ workers
Workflow in Mechanical Turk

Requestors:

Create Tasks → Describe Tasks → Upload Tasks → Accept or Reject

ESQuaReD

Kathryn T. Stolee & Sebastian Elbaum
Crowdsourcing Empirical Studies in Software Engineering
Workflow in Mechanical Turk

Requestors:
- Create Tasks
- Describe Tasks
- Upload Tasks
- Accept or Reject

Types of tasks:
- Short duration (60s. or less)
- Require human intelligence (handwritting analysis, image tagging)
- Specialized (requires certain knowledge) or generic
Workflow in Mechanical Turk

Workers:

- Search for Tasks
- Select Task
- Complete Task
- Submit Task
Workflow in Mechanical Turk

Workers:

1. Search for Tasks
2. Select Task
3. Complete Task
4. Submit Task

Answer Two Short Questions about Yahoo! Pipes - Easy!

Requester: Katie Stolee

HIT Expiration Date: May 13, 2010 (3 days 8 hours)

Reward: $0.20

Time Allotted: 60 minutes

HITs Available: 8

Description: The task is to answer two short questions, comparing two versions of Yahoo! Pipes programs that have the same output.

Keywords: programming, Yahoo, Pipes, survey, mashup, questionnaire, coding, easy

Qualifications Required:

Your Value

Qualification Quiz for UNL Study on Yahoo! Pipes is greater than 90

100 You meet this qualification requirement

HIT approval rate (%) is greater than 90

100 You meet this qualification requirement

Contact the Requester of this HIT
Workflow in Mechanical Turk

1. Create Tasks
2. Describe Tasks
3. Upload Tasks
4. Accept or Reject Tasks
5. Search for Tasks
6. Select Task
7. Complete Task
8. Submit Task
Goal of This Work

Conjecture
Crowdsourcing can be a good solution for recruiting the right type and quantity of participants for an empirical study in software engineering.
Goal of This Work

Conjecture

Crowdsourcing can be a good solution for recruiting the right type and quantity of participants for an empirical study in software engineering.

In this work, we crowdsource a software engineering experiment using Amazon’s Mechanical Turk service, and reflect on our experiences.
Study Definition

**Purpose:** Evaluate the impact of coding practices (e.g., code smells) on end user’s preferences and understanding of web mashups built in Yahoo! Pipes.
Task Description: Given two pipes with the same behavior, one with a smell and one without, select the preferable one.
Task Description: Given two pipes with the same behavior, one with a smell and one without, select the preferable one.
### Experimental Design

#### Experiment Definition

#### Design

#### Selection

#### Instrumentation

#### Operation

#### Analysis

<table>
<thead>
<tr>
<th>Task</th>
<th>Subjects</th>
<th>Pretest</th>
<th>Object</th>
<th>Treatment</th>
<th>Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_1$</td>
<td>$Smell_5$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_2$</td>
<td>$Smell_4$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_3$</td>
<td>$Smell_5$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_4$</td>
<td>$Smell_8$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_5$</td>
<td>$Smell_7$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_6$</td>
<td>$Smell_1$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_7$</td>
<td>$Smell_{5,10}$</td>
<td>$O_3$, $O_4$</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td>$O_1$, $O_2$</td>
<td>$Pipe_8$</td>
<td>$Smell_{2,9}$</td>
<td>$O_3$, $O_4$</td>
</tr>
</tbody>
</table>

$O_1 = \text{Education}$  
$O_2 = \text{Pipes test score}$  
$O_3 = \text{Preference}$  
$O_4 = \text{Time to completion}$
**Lessons Learned:**

- Experimental tasks must be modular and independent, but can be longer (ours took 3-4 minutes, on average).
- Qualification tests can be used to capture pretest measures.
- Cannot control which tasks are completed by which participants.
- Self-selection of tasks may introduce bias that needs to be accounted for in the analysis.
Desired Participant Characteristics:
- Limited computer science education (end users)
- Familiar with Yahoo! Pipes

Mechanical Turk:
- Facilitates recruitment by hosting tasks
- Allows for qualification tests to be administered prior to participation (pretest measures)
Selection and Recruitment

**Lessons Learned**

- 50 qualification tests submitted in two weeks, 38 passed
- 22 participants in total, 14 were considered “end users”
- More variation and unknowns in participants (e.g., age, gender, education, experimental context)
Experimental Task in Mechanical Turk

To perform this HIT, answer the questions below based on Pipes A and B, shown in the images. You may assume that the output of Pipes A and B is equivalent. You will be paid based on response completion.

1. Select the pipe that is easiest to understand
   - A
   - B
   - Same

2. Justify your answer (you must use at least 10 words in your explanation):

   [Space for justification]

Click each image to open a larger view.

Take some time to understand the behavior of each pipe. To answer questions 1 and 2 below, consider the following context:

Pipes with different structures can generate the same output, as is the case with Pipes A and B.
Instrumentation

Experiment Definition

Design

Selection

Instrumentation

Operation

Analysis

Lessons Learned

- Need to learn how to use a new tool and/or API
- Need to adjust presentation of tasks to fit the Mechanical Turk interface
- All tasks are in competition with other tasks for participants, so the task description must be enticing.
Experiment Operation

Mechanical Turk:
- Hosts tasks for a custom time period (2 weeks)
- Administers qualification tests (50 requests)
- Maintains user anonymity
- Collects results and metrics (188 tasks submitted)
Lessons Learned:

- Hand-grading qualification tests introduce delay, and may discourage further participation
- Time to completion is reported, but is suspicious
Response Quality:
- Qualitative responses were detailed and demonstrated understanding (Average length was 31 words, only 10 were required)
- Did not need to reject any responses
Lessons Learned:

- We were able to validate our hypotheses (for only $42)
- May need to throw away some data due to learning (we threw away 28 responses)
- Too many responses from a small group of participants could skew results
Crowdsourcing allowed us to:

- Obtain a sufficient number of participants with the desired characteristics
- Evaluate our research questions using an empirical study for low cost

However...

- Requires careful experimental design to work within the Mechanical Turk infrastructure
- Due to the “unknowns” about the subjects and environment, crowdsourcing may not be appropriate for all studies