Crowdsourcing Software Development: Silver Bullet or Lead Balloon

Brian Fitzgerald
Lero
The basic promise of crowdsourcing software development is that high quality software can be produced quickly and at low cost by a large pool of self-selecting experts.
Overview

1. Introducing CSD
2. In-Depth Case Study of CSD (from customer perspective)
3. Theoretical Model of CSD tested with large-scale sample data
4. Conclusions
Introducing CSD

No matter who you are, most of the smartest people work for someone else.

—Bill Joy
Crowdsourcing: Leveraging Wisdom of the Crowd

- Longitude Problem (1714)
- *Vox Populi* (Galton 1907)
- Amazon Mechanical Turk
- InnoCentive
## Positioning Crowdsourcing vs. Outsourcing vs. Opensourcing*

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Outsourcing</th>
<th>Opensourcing</th>
<th>Crowdsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locus of Control</td>
<td>Company</td>
<td>Community</td>
<td>Company</td>
</tr>
<tr>
<td></td>
<td>IP protected</td>
<td>IP open</td>
<td>IP protected</td>
</tr>
<tr>
<td>Nature of Workforce</td>
<td>Known</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Narrow &amp; deep</td>
<td>Broad &amp; deep</td>
<td>Broad &amp; deep</td>
</tr>
<tr>
<td>Crowd Motivation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Company Motivation</td>
<td>Resource saving</td>
<td>Innovation</td>
<td>Resource saving</td>
</tr>
<tr>
<td></td>
<td>Market growth</td>
<td></td>
<td>Innovation</td>
</tr>
<tr>
<td></td>
<td>Commodification</td>
<td></td>
<td>Market growth</td>
</tr>
</tbody>
</table>

Expected Benefits from Crowdsourcing

Cost Reduction
- Lower labour costs in different regions
- Eliminates recruiting overhead

Faster Time-to-Market
- ‘Follow-the-sun’ 24/7
- Parallel decomposition of tasks

High Quality
- Self-selecting experts with broad and deep knowledge
- Linus’ Law: Given enough eyeballs, every bug is shallow

Creativity and Open Innovation
- Go beyond internal fixed mindset
2

Case study*

Case: “Tech Platform Inc. (TPI)”

TPI: global player in cloud solutions

- 400 sales offices in 75 countries
- 50K employees

Crowdsourced project: “Titan”

**Task:** Porting a migration utility used by field engineers from a stand-alone tool to a web application (128 panels)
Testing the Wisdom of this Crowd

https://goo.gl/IKpgYi

CROWDSOURCING SOFTWARE DEVELOPMENT SURVEY

Survey

Question 1 - Cost for 128 HTML5 panels in US dollars?

Question 2 - Duration for development of 128 HTML5 panels in days?

Question 3 - Number of defects reported for 128 HTML5 panels?

Submit
TopCoder.com

>1 million members \textit{from} < 50K \textit{in} 2004

but < 0.5% active developers
>1 million members *from* <50K in 2004

but < 0.5% active developers
TopCoder Roles

Platform Specialist, Co-Pilot, Crowd Contestants

TopCoder mantra

TopCoder does heavy lifting/process management
Customer is “conductor of world-wide talent pool”

“Software development cost reduction of 62%”
(TopCoder, Tech Crunch 2013)
TopCoder Contest Interface

Contest Name

EMP Panels Phase 3B UI Prototype

Prizes/Cost

1st Place: $1,000
2nd Place: $500
Reliability Bonus: $200
DR Points: 450

Detailed description

Contest Overview

Detailed Requirements

The primary goal of this contest is to design the look and feel of a web application which has very defined goals. The look and the look is dashboard oriented, for business purposes. We are trying to provide a simple interface to a complicated tool. Our users need to be gently pushed to particular flows through the application so they don't get overwhelmed by the total number of choices available to them.

EMP is a migration planning application that's used to streamline the planning process for data migrations onto storage arrays. It's currently implemented as a stand-alone single user desktop installed Windows application, but in the process of being ported to a web application. The goal of this project is to replace the existing EMP UI with a modern, web-based application which will integrate with the existing EMP repository.

Contest info
What software parts to crowdsourcer?

- **Least domain knowledge required**
- **Self-contained**
- **Scarce internal resources**
Coordination: Communication

Multiple interaction layers

**TopCoder (TC)**
- Account Manager
- Platform Specialist
- Co-Pilot
- Contestants

**TPI**
- TC Program Manager
- Titan Program Manager
- TC Architect
- Titan Product Architect
- Tactical Scrum Team
- Normal Scrum Teams

TopCoder waterfall process ➔ TPI agile process
Challenge to integrate TC deliverables into Sprints
**Coordination: Communication**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Panels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Dashboards</td>
<td>40</td>
</tr>
<tr>
<td>2 Flagship product I</td>
<td>18</td>
</tr>
<tr>
<td>3 Flagship product II</td>
<td>33</td>
</tr>
<tr>
<td>4 Network devices</td>
<td>14</td>
</tr>
<tr>
<td>5 Legacy and third-party</td>
<td>23</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>128</strong></td>
</tr>
</tbody>
</table>

It feels like we’ve produced a million specification documents, but obviously we haven’t. The way we do specifications for TopCoder is entirely different to how we do them internally. - TPI Architect
Contest failure due to lack of submissions

53 contests but just 84 submissions

<table>
<thead>
<tr>
<th>Type</th>
<th>Registrants</th>
<th>Submissions</th>
<th>%Sub/Reg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copilot</td>
<td>13</td>
<td>6</td>
<td>46%</td>
</tr>
<tr>
<td>Studio</td>
<td>34</td>
<td>7</td>
<td>21%</td>
</tr>
<tr>
<td>Architecture</td>
<td>90</td>
<td>12</td>
<td>13%</td>
</tr>
<tr>
<td>Assembly</td>
<td>476</td>
<td>36</td>
<td>8%</td>
</tr>
<tr>
<td>Test Suite</td>
<td>8</td>
<td>1</td>
<td>13%</td>
</tr>
<tr>
<td>UI Prototype</td>
<td>99</td>
<td>22</td>
<td>22%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>720</strong></td>
<td><strong>84</strong></td>
<td><strong>12%</strong></td>
</tr>
</tbody>
</table>

Two’s company, 1.6 is a crowd...

IP Loss: Unknown workforce - 720 registrants saw specifications
Quality Assurance

- TC Waterfall approach pushes error identification later in life-cycle

- "Fleeting relationship"
  - Lack of developer continuity across contests – recurrence of same bugs
  - No domain knowledge built up by developers
TopCoder warranty periods unsuitable

5 days to accept/reject deliverable
But cannot accept/reject part of deliverable
Tendency to accept to not deter contestants

Additional 30-day warranty period
But fast changing code base – not useful to integrate new fixes after 30 days
Counting the Cost!
1st Prize

-Suggested by Co-Pilot
-Varied from $600 to $2,400

$1,000
Total Cost
1st $1,000
2nd $500
------------
$1,500

2nd Prize

50% of first prize:

$500
Total Cost
1st $1,000
2nd $500
R. Bo. $200
------------------
$1,700

Reliability Bonus
Up to 20% of first prize:
$200
Total Cost

1st $1,000
2nd $500
R.Bo. $200
DR $450

$2,150

Digital Run

45% of first prize

1 Point = $1.00

$450
<table>
<thead>
<tr>
<th>Description</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>$1,000</td>
</tr>
<tr>
<td>2nd</td>
<td>$500</td>
</tr>
<tr>
<td>R. Bo.</td>
<td>$200</td>
</tr>
<tr>
<td>DR</td>
<td>$450</td>
</tr>
<tr>
<td>Spec. R</td>
<td>$50</td>
</tr>
</tbody>
</table>

**Total Cost**: $2,200

**Spec. Review**: $50
Total Cost
1<sup>st</sup> $1,000
2<sup>nd</sup> $500
R.Bo. $200
DR $450
Spec.R $50
Rev.B. $800

--------------
$3,000

Review Board

$800
Co-Pilot Fees:

$600
Total Cost

1st $1,000
2nd $500
R.bo. $200
DR $450
Spec.R $50
Rel.B. $800
CP $600

---------------------
Subtotal $3,600

TC multiplier x 2

---------------------
Price of 1 contest:
$7,200

TC Commission = total of above
Total Cost
1st $1,000
2nd $500
R.bo. $200
DR $450
Spec.R $50
Rev.B. $800
CP $600

---------------------
Subtotal $3,600
TC multiplier x 2
---------------------
Price of 1 contest: $7,200

Platform “Cockpit” Fees for TPI:

$30,000 per month*

* Varies per customer – as low as $3,000 per ‘cockpit seat’
Cost: $650,000

Plus extra internal overhead in preparing specs and coordination effort

Time: 215 calendar days

(695 contest days)

Quality: 506 bug issues
### Prior ‘Academic’ Crowd

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost (US$)</td>
<td>$211,000</td>
</tr>
<tr>
<td>Time</td>
<td>145 days</td>
</tr>
<tr>
<td>Quality (# bugs)</td>
<td>96</td>
</tr>
</tbody>
</table>

### Prior ‘Practitioner’ Crowd

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost (US$)</td>
<td>$378,000</td>
</tr>
<tr>
<td>Time</td>
<td>174 days</td>
</tr>
<tr>
<td>Quality (# bugs)</td>
<td>158</td>
</tr>
</tbody>
</table>
Theoretical Model for CSD*

OPEN ACCESS!
Data Source for Model Construction

- Case study
- Crowdsourcing literature
- Topcoder platform API
# Model Variables

<table>
<thead>
<tr>
<th>Construct variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competition Parallelism</td>
<td>The number of competitions that are run simultaneously within the same project.</td>
</tr>
<tr>
<td>Competition Reward</td>
<td>First Prize money offered for a competition.</td>
</tr>
<tr>
<td>Competition Duration</td>
<td>Number of days between the registration deadline and the submission deadline (included).</td>
</tr>
<tr>
<td>Crowd Killer Registrations</td>
<td>Developers whose average win count is $3 \times \sigma$ greater than the average.</td>
</tr>
<tr>
<td>Crowd Interest</td>
<td>Number of registrations for a competition.</td>
</tr>
<tr>
<td>Crowd Participation</td>
<td>Number of submissions. Only registered members are able to submit.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand for Workforce</td>
<td>At a given time, the number of competitions that are running at the time of a competition being advertised.</td>
</tr>
<tr>
<td>Supply of Workforce</td>
<td>The number of platform members at the time of a competition’s advertisement.</td>
</tr>
<tr>
<td>Number of Technologies</td>
<td>The number of technologies that are specified for a competition.</td>
</tr>
</tbody>
</table>
H1: Running competitions in parallel is negatively associated with crowd interest
H2: Competition reward is positively associated with increased crowd interest
H3: Competition duration is positively associated with crowd interest
H4: Interest from the crowd is positively associated with participation
H5: ‘Crowd killer’ registration is negatively associated with participation
13,602 (completed) competitions on the Topcoder platform (2007-2016)

20,747 Topcoder crowd members involved
Evaluating Model Fit (SEM)

Model Fit Indexes

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi^2$ Yuan-Bentler corrected</td>
<td>7.688</td>
</tr>
<tr>
<td>RMSEA</td>
<td>0.067</td>
</tr>
<tr>
<td>Comparative Fit Index (CFI)</td>
<td>0.993</td>
</tr>
</tbody>
</table>

* $p < 0.05$, ** $p < 0.001$
Conclusions
Conclusions

- Costly++
- Quality issues
  - Waterfall competitions – late detection of errors
  - No accretion of domain knowledge - fleeting relationship
- Crowd may be very small
  - Running too many contests in parallel reduces crowd size
  - Increasing price or duration makes no difference
- Beware of Crowdkillers
- Crowdsourcing platforms lack *transparency* and *recombination* (*Secret Sauce* in Open Source)
Thank You