System Perspective and Lean Coordination

The case of Open Source Software

Joao Castro

Committee: Warren Seering
Eric Rebentisch
Chris Magee
Coordination and lean

What they mean to me:
- Lean as being effective, efficient, minimal waste
- Coordination as being effective in managing interactions in complex environments

Can we get efficient coordination?
- In high complexity environments?

Product development is the study context
Coordination

- Coordination is consistently cited as one of the most important factors of competitive advantage.

  "The primary task of management is to get people to **work together in a systematic way**"  

- Literature describes many methods:
  - Centralize people
  - Centralize information
  - Facilitate Communication
  - Structure Communication
  - Structure Processes
Field exploration

- Visited and collected data, interviews from different PD companies:
  - Industrial machinery (2)
  - Aerospace (2)
  - Medical equipment
  - Services (2)
  - Food packaging
  - PD consultancy

- Factors affecting coordination are varied:
  - Team size
  - Schedule
  - Reputation for prob. solving
  - Product complexity
Where to focus?

“Ohno thought that assembly workers could probably do most of the functions of the specialists and do them much better because of their direct acquaintance with conditions on the line.”

Womack, Jones, and Roos, (1990)

*The Machine That Changed The World*
Where to focus?

- Brook’s Law implies that the **ideal size for a programming team is one** - a single developer who never has to stop to communicate with a colleague.

- This approach streamlines everything, and it also provides insurance that the project will retain “**conceptual integrity**”.

Where to focus?

“I also believed there was a certain critical complexity above which a more centralized, a priori approach was required.”

“Linus Torvalds's style of development came as a surprise. (...)The fact that this bazaar style seemed to work, and work well, came as a distinct shock.”

Eric Raymond, (1999) *The Cathedral and the Bazaar*
Where to focus?

But, as Galileo is said to have murmured after officially recanting his statement that the earth moves around the sun:

“And yet it moves!”

What is going on here?”

von Krogh and von Hippel (2006)

The Promise of Research on Open Source Software
Driving Hypothesis

Individuals in a collaborative environment and behaving autonomously are able to *efficiently* solve complex problems.

In other words:

Coordination is possible without heavy supervisory and overhead methods.
Is there a driver behind what connections are made?

- Connections in a complex project will happen
- Given two similar projects, will the connections also be similar?

Test scenario:
- Multiple concurrent engineering sessions
  - No barriers to communication in each session
- Sessions have different objectives and different function areas are selected and staffed
Are connections made on purpose?
Are connections made on purpose?

Data for real projects provided by Mark Avnet
Are connections made on purpose?

![Graph showing communication link similarity vs. function similarity for real and simulated projects.](image)

- Real projects
- Simulated projects

Data for real projects provided by Mark Avnet
Hypotheses and questions

• Hx: Different levels of product complexity require different patterns of people communication. Number, focus and attention span vary.

• Hx: Level of oversight by a person is extremely limited (they only see what they do) when compared to the whole system.

• Hx: People's attention (as verified by their footprint) varies through time. Once something is done, they move on and do not return to it.

• Hx: Systems that operate under the freedom of participants have high redundancy communication channels.

• Hx: System critical components are verified by several people at different times.
Finding more data

- Product complexity, component interaction and specialty interaction is context specific and varies across industries

- But, at a low-level, we can abstract to:
  - Component A <- logical interaction - > Component B

- This allows us to study the same problem in different industries and try to understand and generalize

- But to understand different behaviors, a lot of very detailed data from several projects is required
  - Focus on one area: software
Why software? Fits the topic

- Software code is also made of a set of logical interactions:
  - procedures, functions, variables and objects

- Complex software is developed collaboratively by teams.
  - Each member works on a sub-part of the system that interacts
  - Members often work on code written by others
  - Teams can typically be geographically dispersed

- Software is key part in *almost* all modern complex products
Why software? Good data for research

• Software engineering practices have excellent book-keeping methods which give us:
  - Fine grained information
  - Complete information. Long history on past projects is available
  - Uniform data over time
  - Even small projects generate large volumes of changes making it possible to detect even small effects statistically

• The data collection is nonintrusive / non-disturbing
  - doesn’t require resources from project to help with the data collection

• The data collection is cheap
  - no impact on the project as this data collection is already performed

Adapted from (Mockus, Weiss et al. 2003)

• Data is ripe for processing
  - Using computer to process and analyze a lot of information
Why software? Even better data

- The information-based nature of software products brings another benefit in that **we can track the evolution of a design over time**. (…) For a researcher, this presents an opportunity to follow the “living history” of a design, a technique that is typically not possible for physical products.

Why software? Available literature

- Academic Research
  - Software development process
  - Coordination in software projects
  - Measures of software complexity
  - Visualization of software and team participation
  - Social settings, motivations and behaviors of participants
  - Social network analysis of software projects
  - System evolution
  - Case studies

- Ethnographies
Software Data. What it looks like

- Change in the code
- Bug correction activity

<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>2007-11-20 10:33:10 PST</td>
<td>Severity</td>
<td>major</td>
<td>norma</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Component</td>
<td>Application</td>
<td>Calendar UI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Priority</td>
<td>P1</td>
<td>P2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Product</td>
<td>Chandler</td>
<td>Cesmo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Summary</td>
<td>Events belonging to several calendars are only displayed correctly when one of the calendars is unchecked</td>
<td>EVENTS BelONGING TO seVERAL CALENDARS ARE ONLY SYNC'D ON THE CALENDAR THEY WERE CREATED IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Target Milestone</td>
<td>0.7.3</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version</td>
<td>0.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<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>2007-11-20 11:41:10 PST</td>
<td>AssignedTo</td>
<td><a href="mailto:pbossesf@osafoundation.org">pbossesf@osafoundation.org</a></td>
<td><a href="mailto:mde@osafoundation.org">mde@osafoundation.org</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC</td>
<td><a href="mailto:adamm@osafoundation.org">adamm@osafoundation.org</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2007-11-30 15:02:07 PST</td>
<td>Target Milestone</td>
<td>---</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2005-02-28 12:53:54 PST</td>
<td>AssignedTo</td>
<td><a href="mailto:mde@osafoundation.org">mde@osafoundation.org</a></td>
<td><a href="mailto:travis@osafoundation.org">travis@osafoundation.org</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Target Milestone</td>
<td>1.0</td>
<td>Future</td>
</tr>
</tbody>
</table>

Revision: 4380
Author: morgen
Date: 7:11:38 PM, Tuesday, February 01, 2005
Message:
Added a dialog to prompt the user for new webdav account info, and fixed some typos

---
Added: /trunk/chandler/parcels/osaf/framework/sharing(AccountInfoPrompt.py
Added: /trunk/chandler/parcels/osaf/framework/sharing/AccountInfoPrompt.wdr
Added: /trunk/chandler/parcels/osaf/framework/sharing/AccountInfoPrompt_wdr.xrc
Modified: /trunk/chandler/parcels/osaf/framework/sharing/Sharing.py
Projects of interest - Open Source

- Open Source Software is a type of software project that relies on a loose articulation between developers.

- Open source software projects are based on voluntary contributions and involve only very light coordination activities by a central project team. 

  Kogut and Metiu (2001) Open-Source Software Development and Distributed Innovation

- “What is perhaps most surprising about the process is that it lacks many of the traditional mechanisms used to coordinate software development, such as plans, system-level design, schedules, and defined processes.”

  Mockus, Fielding et al. (2002) Two Case Studies of OSS Development: Apache and Mozilla

- “everyone, under this type of project management, is self-determining”

  Mockus and Herbsleb (2002) Why Not Improve Coordination in Distributed Software Development by Stealing Good Ideas from Open Source?
Open Source Software

• Because of their open policies, project data is already public
  - Time to gain access to projects is cut to almost zero
  - Data is available online
  - No need to travel
  - No need to navigate NDAs
Rewriting our Hypotheses and questions

- Hx: Different levels of product complexity require different patterns of people behavior. Number, focus and attention span vary.

- Hx: Level of oversight in code by a person is limited (the files they edit mostly reference themselves)

- Hx: Does people's attention (as verified by their footprint) varies through time. Do they come back to their older files while editing new ones?

- Hx: How well does the ensemble of perspectives cover the whole code?

- Hx: A developer engages in coordination only with those who are part of his system view

- Hx: Most time is spent on the boundary components than on independent components.
# Projects collected (so far)

<table>
<thead>
<tr>
<th>Project name</th>
<th>Description</th>
<th>Number of developers</th>
<th>Got code</th>
<th>Source code files</th>
<th>Lines of code</th>
<th>Number of functions</th>
<th>Number of edits in DB</th>
<th>Got Bugs</th>
<th>Number of bugs</th>
<th>Got ML</th>
<th>Total # of Msgs</th>
<th>Got IRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandler</td>
<td>PIM</td>
<td>44</td>
<td>Yes</td>
<td>5374</td>
<td>1195429</td>
<td>2560</td>
<td>14835</td>
<td>Yes</td>
<td>12427</td>
<td>Yes</td>
<td>10405</td>
<td>Can get</td>
</tr>
<tr>
<td>Audacity</td>
<td>Audio editing</td>
<td>Can get</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache</td>
<td>Web Server</td>
<td>95</td>
<td>Yes</td>
<td>8645</td>
<td>1382475</td>
<td>694</td>
<td>18020</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireshark</td>
<td></td>
<td>40</td>
<td>Yes</td>
<td>12560</td>
<td>5800442</td>
<td>7374</td>
<td>26421</td>
<td>Can get</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MythTV</td>
<td>Media center</td>
<td>41</td>
<td>Yes</td>
<td>20792</td>
<td>3941132</td>
<td>18091</td>
<td>17221</td>
<td>Can get</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rsync</td>
<td>Backup</td>
<td></td>
<td>Yes</td>
<td>382</td>
<td>73392</td>
<td>218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Git</td>
<td>Repository mgmt</td>
<td>41</td>
<td>Yes</td>
<td>1796</td>
<td>322768</td>
<td>1042</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>Mobile OS</td>
<td>Very big</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GNUmeric</td>
<td>Spreadsheet</td>
<td>217</td>
<td>Yes</td>
<td>6583</td>
<td>4818253</td>
<td>2366</td>
<td>15827</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gimp</td>
<td>Image processing</td>
<td>260</td>
<td>Yes</td>
<td>14455</td>
<td>6648419</td>
<td>7610</td>
<td>24772</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Juice</td>
<td>Podcast</td>
<td>No code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Songbird</td>
<td></td>
<td>30</td>
<td>Yes</td>
<td>14433</td>
<td>1924184</td>
<td>1626</td>
<td>7954</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

825,859 records in database so far
Example of data from an open source project

Chandler
- Tracked from Aug’02 to Nov’08
- 43 developers
- Source Code:
  - 14 835 changes (commits)
  - 5 347 files
  - 2 560 functions, 1 195 429 loc
- Bugs:
  - 12 427
- Mailing list:
  - 10 405 emails exchanged
- Internet chat:
  - ?
Data analysis

- Demographics of projects

- Analysis of individuals
  - Footprint - what parts of the product are focused
  - Change over time

- Analysis of team
  - Communication network
  - Visibility overlap
  - System hand-offs

- Analysis of product
  - Function call graph
  - Bug duration, origin, severity

- Analysis of product, team
  - Overlap in communication and objects
Some results

• Scale of review and rework
  - How many times a file is edited

• System visibility
  - How much of the product does each one see

• System overlap
  - Who worked on whose files

• Evolution of personal footprint
Scale of review and rework

Histogram of number of edits on a file

Average 9 edits/file
### System visibility

#### How much do they work on?

Top ten developers and average on each project

<table>
<thead>
<tr>
<th>Chandler</th>
<th>Apache</th>
<th>Gimp</th>
<th>Gnumeric</th>
<th>MythTV</th>
<th>Wireshark</th>
<th>Songbird</th>
</tr>
</thead>
<tbody>
<tr>
<td>36%</td>
<td>67%</td>
<td>61%</td>
<td>71%</td>
<td>35%</td>
<td>44%</td>
<td>39%</td>
</tr>
<tr>
<td>32%</td>
<td>35%</td>
<td>57%</td>
<td>29%</td>
<td>15%</td>
<td>36%</td>
<td>25%</td>
</tr>
<tr>
<td>29%</td>
<td>22%</td>
<td>31%</td>
<td>14%</td>
<td>15%</td>
<td>34%</td>
<td>23%</td>
</tr>
<tr>
<td>24%</td>
<td>21%</td>
<td>18%</td>
<td>13%</td>
<td>14%</td>
<td>32%</td>
<td>22%</td>
</tr>
<tr>
<td>14%</td>
<td>21%</td>
<td>8%</td>
<td>12%</td>
<td>13%</td>
<td>25%</td>
<td>20%</td>
</tr>
<tr>
<td>12%</td>
<td>20%</td>
<td>8%</td>
<td>10%</td>
<td>12%</td>
<td>24%</td>
<td>13%</td>
</tr>
<tr>
<td>5%</td>
<td>14%</td>
<td>7%</td>
<td>9%</td>
<td>10%</td>
<td>23%</td>
<td>12%</td>
</tr>
<tr>
<td>3%</td>
<td>13%</td>
<td>7%</td>
<td>8%</td>
<td>10%</td>
<td>13%</td>
<td>8%</td>
</tr>
<tr>
<td>3%</td>
<td>12%</td>
<td>7%</td>
<td>7%</td>
<td>9%</td>
<td>12%</td>
<td>6%</td>
</tr>
<tr>
<td>3%</td>
<td>12%</td>
<td>5%</td>
<td>7%</td>
<td>8%</td>
<td>12%</td>
<td>4%</td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0%</td>
<td>4.6%</td>
<td>1.0%</td>
<td>1.2%</td>
<td>4.7%</td>
<td>8.1%</td>
<td>6.5%</td>
</tr>
</tbody>
</table>

% of files in the project that have at least one edit by the member
Chandler: 43 developers
System overlap

Who worked on whose files (1st order)

Apache: 94 developers

© 2009 Joao Castro, Engineering Systems Division, Massachusetts Institute of Technology
<table>
<thead>
<tr>
<th>eikki</th>
<th>davids</th>
<th>bear</th>
<th>markie</th>
<th>john</th>
<th>morgen</th>
<th>grant</th>
<th>bkirsch</th>
<th>pje</th>
<th>stearns</th>
<th>jeffrey</th>
<th>alecf</th>
<th>donn</th>
<th>jecho</th>
<th>pavlov</th>
<th>d a n</th>
<th>bind</th>
<th>andy</th>
<th>chao</th>
<th>jbotz</th>
<th>sheila</th>
<th>anthony</th>
<th>chaolam</th>
<th>atotic</th>
<th>y s</th>
<th>michael</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>192</td>
<td>130</td>
<td>56</td>
<td>79</td>
<td>37</td>
<td>52</td>
<td>222</td>
<td>119</td>
<td>91</td>
<td>52</td>
<td>119</td>
<td>174</td>
<td>25</td>
<td>81</td>
<td>171</td>
<td>279</td>
<td>59</td>
<td>102</td>
<td>159</td>
<td>53</td>
<td>171</td>
<td>119</td>
<td>174</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>90</td>
<td>182</td>
<td>50</td>
<td>45</td>
<td>49</td>
<td>7</td>
<td>25</td>
<td>95</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
</tr>
<tr>
<td>83</td>
<td>176</td>
<td>50</td>
<td>45</td>
<td>49</td>
<td>7</td>
<td>25</td>
<td>95</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
</tr>
<tr>
<td>123</td>
<td>216</td>
<td>50</td>
<td>45</td>
<td>49</td>
<td>7</td>
<td>25</td>
<td>95</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
</tr>
<tr>
<td>107</td>
<td>200</td>
<td>50</td>
<td>45</td>
<td>49</td>
<td>7</td>
<td>25</td>
<td>95</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
<td>111</td>
<td>103</td>
</tr>
</tbody>
</table>
Evolution of personal footprint

$\Delta t \text{ v Distance}$
Evolution of personal footprint

$\Delta t \text{ v Distance}$
Evolution of personal footprint

$\Delta t \, v \, \text{Distance}$
Next steps

• Integrate in analysis the *bug* and *mailing list* data

• Analyze relationship between
  - Product and communication
  - Analyze problem solving over time and product structure

• Analyze using FCG instead of folder hierarchy for product structure
Thank you