Incorporating Social Software into Agile Distributed Development Environments

Fabio Abbattista, Fabio Calefato, Domenico Gendarmi, Filippo Lanubile

Dipartimento di Informatica
University of Bari, Italy

Outline

- Distributed Agile Development
- Social Software overview
- Tool Support for Collaborative Sw Development
- Discussion on Social Software in DAD
Distributed Agile Development

- Agile development
  - Time-boxed short releases of working software
  - High level of collaboration and communication (collocated)
  - Self-organization and regular adaptation to changing circumstances
- Distributed Agile Development - DAD
  - Aims at gaining the benefit of agile development also in distributed settings
- So why DAD? What’s special?
  - Blending together agility and distribution is not as easy as just relaxing the collocation constraint from the Agile Manifesto…

Agile vs. Distributed Sw Development

<table>
<thead>
<tr>
<th>Some Agile practices</th>
<th>Some DSD characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular informal communication</td>
<td>Reduction of formal &amp; informal communication</td>
</tr>
<tr>
<td>Flexible requirements</td>
<td>Stable requirements</td>
</tr>
<tr>
<td>Informal agreement</td>
<td>Controlled processes</td>
</tr>
</tbody>
</table>
Social Software

- General definition encompassing a set of tools and applications that enable group interaction and computer-mediated communication
  - Has a focus on the creation of online communities (not only business-oriented as for Groupware)
  - Includes everything from classic email to 3D virtual worlds
- Sometimes expressed as Web 2.0
  - Can be misleading
  - Comprises only the latest form of social software excluding old Web 1.0 collaborative apps

Web 2.0

**Principles**
- Interaction
  - (Web-based) Rich user interfaces
- Participation
  - User generated content
- Collaboration
  - Social network effects

**Applications**
- Get better the more people use it
- Consume and remix data from multiple sources while providing their own data and services
- Create network effects through an "architecture of participation"
Popular Web 2.0 Software: Blogs

- More interaction among writers and readers
- Blogosphere

SOSEA 2008, Sept. 15, L’Aquila, Italy

A fresh take on the browser
9/01/2008 02:10:00 PM
At Google, we have a saying: "launch early and iterate." While this approach is usually limited to our engineers, it apparently also applies to our masturblogers! As you can now read in the blogosphere, we hit "launch" on a comic book introducing our new open-source browser, Google Chrome. We are excited to announce the next major innovation from the Google Chrome browser, now more the mainstream powerhouse — you can’t miss it! We will be launching the beta version of Google Chrome tomorrow in more than 100 countries.

Why are we launching Google Chrome? Because we believe it can still be done faster and at a lower cost, help drive innovation on the web.

All of us at Google spend much of our time working inside a browser. We search, chat, email, and collaborate in a browser. And in the same time, we shop, bank, read news and keep in touch with friends — all using a browser. Muslims can also spend hours on browsers, viewing websites, and reading content.

Wikis

- Collaborative authoring
- No distinction between writers and readers

MoinMoinWiki

Ubuntu Team Wiki

Welcome to the Ubuntu Team Wiki, a place for the Ubuntu community to discuss ideas and store team-related information. The documentation for Ubuntu users can be found on the HelpWiki.

You can edit all of the pages on this site. For more information on editing, see Help:editing. Please observe the Ubuntu Code of Conduct at all times. Please use the Help wiki instead of the Team wiki for user documentation.

Resources and Teams

1. Resources and Teams
   1. Software
   2. Quality Assurance and Packaging
   3. Marketing and Advocacy
   4. Support and Documentation
   5. Legal
   6. Accessibility
   7. Acceptance
   8. Documentation and Internationalization
2. Community
   1. Code
   2. Development, Quality, and Standards
   3. Events
   4. Meetings

Movable Type

Collaborative Development Group
Collaborative Tagging

- Personal way of organizing information
- Tight feedback loop

Social Networking

- Digital identity through public profiles
Mashups

- Single access point for different data and services

Tool support for Collaborative Sw Development

- Tools are the key to enable distributed team work
- Most used tools for building software include:
  - Software Configuration Management (e.g., CVS, SVN)
  - Bug & Issue tracking systems (e.g., Bugzilla, Jira)
  - Build and release management (e.g., make, CruiseControl)
  - Knowledge Center / Document Repository (FAQ, guidelines, howto)
  - Communication tools (email, IM, tele-/video-conference)
Collaborative Development Environments (CDE)

- Integrate tools needed by teams to manage software development projects
- Initially born to manage OSS projects
- Nowadays have become a critical infrastructure for distributed development projects in general
- More and more used also to support agile development

Prof. Filippo Lanubile

Tools available in CDE

- Asynchronous communication
  - Mailing list, forum, news, blog
- Synchronous communication
  - IRC, IM
- Software Configuration Management (SCM)
  - CVS, SVN
- Web-based documents repository
  - Howtos, guidelines, FAQ
  - Wiki
- Web-/FTP-server for release download
- Task management (to do list)
- Bug/Issue tracking system

Prof. Filippo Lanubile
SourceForge
http://sourceforge.net/

Tigris.org
http://www.tigris.org/
**GForge**  [http://gforge.org/](http://gforge.org/)

GForge helps you manage the entire development life cycle.

- **Major Feature List**
  - [Diagram of features]
- Additional Tools:
  - [List of additional tools]

**Collab CDE**  [http://cde.di.uniba.it/](http://cde.di.uniba.it/)

Collab CDE is the collaborative development environment (CDE) developed by Prof. Filippo Lanubile. It is the software group of a university and provides various tools and services for collaborative development.

- **Collaborative Development Environment**
  - **Forum**
  - **Tracker**
  - **Release Management**
  - **Source Code Management**
  - **Statistics**
    - Projects: Total 17, Top 3 Projects: 013, 035, 012

**Collab CDE**

- **Participate**
  - [List of roles and responsibilities]

---

**Contact Information**

- **Prof. Filippo Lanubile**
  - [Email and contact details]
Collaborative Development Group

Trac
http://trac.edgewall.org/

Welcome to the Trac Project.

Trac is an enhanced wiki and issue tracking system for software development projects. Trac uses a minimalistic approach to web-based software project management. Our mission is to help developers write great software while staying out of the way. Trac should impose as little as possible on a team’s established development process and policies.

It provides an interface to Subversion, an integrated Wiki and convenient reporting facilities.

Trac allows wiki markup in issue descriptions and commit messages, creating links and seamless references between bugs, tasks, changesets, files and wiki pages. A timeline shows all project events in order, making the acquisition of an overview of the project and tracking progress very easy.

- Live Demo

If you want to test out some of the features in Trac, please use the demo site kindly provided by...

Collaborative Development Group

Jazz
http://jazz.net/

- Collaborative platform that aim at integrating team activities within a software lifecycle

- Characteristics
  - Extensible (plugin)
  - Scalable (team of teams)
  - Interoperable (API)

- Built upon the experience of the Eclipse dev team
  - Built-in dev process: Eclipse Way
  - Supports the definition of custom processes

Prof. Filippo Lanubile
Challenges of DAD

- Surveyed most recent paper on DAD topic from
  1. XP 2008
     - A keynote on using social computing to organize (collocated) sw development
  2. Agile 2008
     - A number of experience reports from practitioners on challenges encountered
     - No or limited focus on generalizing lessons learned
  3. ICGSE 2008
     - Paasivaara et al., “Distributed Agile Development: Using Scrum in a Large Project”
     - Sureshandra et al., “Adopting Agile in Distributed Development”
     - Cristal et al., “Usage of SCRUM Practices within a Global Company”

<table>
<thead>
<tr>
<th>Study description</th>
<th>Project Measures</th>
<th>Agile Methodology</th>
<th>Challenges</th>
<th>Lessons learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paasivaara et al.,</td>
<td>1 project 2 sites (Norway, Malaysia) 1.5 year</td>
<td>Distributed SCRUM of SCURMS</td>
<td>Videoconference issues, both technical (low bandwidth) and conversational (slowness and silence) Rqmts often misunderstood</td>
<td>Multiple communication mode are needed</td>
</tr>
<tr>
<td>Sureshandra et al.,</td>
<td>90 projects 7 Sites (India, …) 3 years</td>
<td>Distributed SCRUM</td>
<td>N.A.</td>
<td>Dedicated videoconferencing equipment Photo-chart of team members at sites Transition to distributed agile should be stepwise</td>
</tr>
<tr>
<td>Cristal et al.</td>
<td>2 projects 2/3 sites (US, S. America, Asia) 1+ year</td>
<td>Distributed SCRUM of SCRUMS</td>
<td>Communication issues Inappropriate team organization and task assignment Lack rqmts documentation</td>
<td>Document and share SCRUM meetings minutes Less document does not mean less valuable Global taskboard</td>
</tr>
</tbody>
</table>
Our goal

- Investigate how the incorporation of Web 2.0 apps (i.e., more recent social software other than IM, conferencing, email, etc.) into CDEs can improve communication & knowledge sharing for DAD teams.

Balancing Practices for Distributed Agile Development

- Improve communication
  - Blogs to enhance informal communication
- Facilitate knowledge sharing
  - Wikis to ease up documentation efforts
  - Tagging of code & artifacts to facilitate tracing and retrieval
- Build team trust and culture
  - Social networks profiles as an incentive to preserve digital identities

E.g. Twitter for developers?

- Yammer.com
  - A private Twitter for co-workers
  - “What Are You Working On?”
- Twitterclipse
  - Eclipse plugin under development
  - Could automatically tell others developers what you are working on (much like Mylin does)

Social Software in DAD: Challenges

- Fun factor
  - User participation in general scenarios is non-compulsory
- Critical mass
  - Social benefits come only when a large base of user is reached
- Collaboration as side effect
  - Active participation mainly rely on individual benefits
Thanks for your attention!

Fabio Calefato
calefato@di.uniba.it
Collaborative Development Group
http://cdg.di.uniba.it