From Research to Practice: How Does TXT Plan to Deploy Innovations in Sharing Development Knowledge

Alessandra Bagnato¹, Walid Maalej²

¹ TXT e-solutions
Salita San Barborino 23r Genoa, Italy, 16100
alessandra.bagnato@txt.it

² Technische Universität München
Bolzmannstr. 3, 85748 München, Germany
maalejw@in.tum.de

Abstract: The research project TEAM aims at developing lightweight knowledge sharing mechanisms, optimized for distributed software teams. This paper describes how does TXT e-solutions plan to evaluate and deploy innovations from this project. First we introduce the as-is situation for collaboration and knowledge sharing inside the company. Then we discuss expectations as well as evaluation environments for TEAM and similar research incentives.

1 Introduction

Working in distributed teams rapidly gains increasing importance for professional software development [Ca99, MR05]. These teams face not only the emergence of communication bottlenecks through far-distance collaboration, but also a growing need for extensive communication and knowledge sharing. For example, nowadays reuse happens at a higher level of abstraction and complexity than earlier, as the “construction by configuration” approach continue to emerge [S308]. Correct component integration, effective work with powerful frameworks or successful usage of design patterns, all require significant background and experience knowledge about used artifacts.

Moreover, the globalization of software development enables new alliances and more effective software creation value chains, resulting in shorter turn-around time for knowledge. New versions are released more frequent and the experience with old version ages rapidly. The research project TEAM [TE08] aim at developing lightweight knowledge sharing mechanisms optimized for distributed teams to deal with such agile environments. Key enablers are semantic annotations of development artifacts, search and recommendation features as well as automatic capture and sharing of developers’ contexts.

This paper discusses an evaluation plan for the TEAM research intentions. In a real life development environment inside the company TXT e-solutions [TXT08], a pilot project has been selected with teams in three Italian cities. We introduce the as-is situation and discuss expectations from the TEAM system as well as the evaluation framework.
2 The Context

TXT e-solutions is a software development company, listed on the Italian Stock Exchange since July 2000. Its revenues in 2006 are € 56.3 million. TXT is headquartered in Milan with offices in Torino, Vicenza, Perugia, Bari, Roma and Genova. The company owns subsidiaries in Paris, Lyon, London, Chemnitz, Halle, Haarlem, Barcelona and New York (Figure 2). Most development activities take place in the Italian offices. The company includes about 500 employees and 250 software developers.

Figure 1: Distributed offices of TXT e-solutions

TXT operates in three main areas:

- In Demand & Supply Chain Management, the company offers products and consulting services based on the TXTPERFORM suite. Target markets are fashion, retail, aerospace, defense and automotive.

- In Multi-channel Content Management, the company provides solutions based on the TXT Polymedia product suite that supports various media formats, e.g. TV and SMS. Products target the media, telecommunication, industry and finance markets.

- In Business Intelligence and Business Process Modeling the company operates with specialized solutions dedicated to the primary needs of industry and finance.

At TXT, development is often distributed since the competencies are distributed through various offices. For on distributed project, there is a unique manager and one technical team leader for every involved office. Technical team leader coordinate an average of 2-10 developers.

2.1 The Pilot Project

The usefulness of the TEAM conceptual models and knowledge sharing approaches for TXT will be evaluated by a pilot project, where three distributed teams will develop a software application. Each team includes four developers working in three different locations in Italy: Genova, Milano Bari. While Genova team presents a high expertise in developing graphical user interfaces, Bari team is highly skilled in security matters and algorithms. Most customers are based in Milan.
The pilot aims at creating graphical application to assist the selection of persons that should be included in a team. It will have the following three main components:

- **Matching Engine**: This core component finds employees’ profile that better fit for required skills. The engine will be developed in Milan and Bari.

- **Ontology Tree Displayer**: This component displays the employee’s skill-tree, allowing the easy selection of different nodes. It will be developed in Milan and Genoa.

- **Self Skills Chooser**: This component manages the automatic selection of own skills. It takes input from Ontology Tree Displayer (selected nodes) and displays them to the user, allowing the selection of related skill levels. This component will be developed in the Milan and Bari with support from Genoa developers in graphical issues.

Table 1 introduces libraries already deployed in past projects and planned to be reused in the pilot. Figure 2 summarizes the main project organization, the components to be developed and the libraries to be reused.

<table>
<thead>
<tr>
<th>LIBRARY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>JFreeChart 1.0.6 [JF08]</td>
<td><em>JFreeChart</em> supports the creation of several charts types as pie charts (2D and 3D), bar charts, line charts, time series charts, high-low-open-close charts, Gantt charts etc. <em>JFreeChart</em> can be used in applications, applets, servlets and JSP.</td>
</tr>
<tr>
<td>Google Widgets/Gadgets [Wi08]</td>
<td><em>Web Widget</em> is a portable chunk of code that can be installed and executed within any separate HTML-based web page by an end user without requiring additional compilation. Google supplies its own developer SDK to create widgets for its platforms.</td>
</tr>
</tbody>
</table>

Figure 2: Organization, system architecture and libraries to be reused in the TEAM pilot project.
3 Current Practices of Collaboration and Knowledge Sharing

The various TXT offices extensively use a Virtual Private Network VPN, enabling employees to share data, without perceiving that the machine, with which they are working, is on offices kilometers away. All employees including developers can see content shared in various company machines by means of shared folders. In distributed teams, developers use to share folders on their own machine to the members involved in the project with temporary and non-temporary data. Source code and documentation such as installation manuals, user guides, system requirements, and component descriptions are kept and maintained centrally by means of SVN repositories.

In addition, TXT developers working in a distributed team have machines equipped with headphones to facilitate skype calls [SK08]. They use to have almost daily skype calls with team members from other offices, and exchange emails with more technical details. They also make use of the yugma tool [YU08] to share desktop content during the calls. Multi users conferences via skype are seldom used due to the experienced connection difficulties. In a weekly base, a phone conference is organized with 4-5 participants. To communicate and discuss complex models that often used in software development, calls are not sufficient. Although, screen sharing helps communicating the discussion context, offline knowledge articulations are required to clarify and structure ideas. Bugzilla [BU08] is rarely used to internally report bugs. Developers often prefer mail exchange and direct phone calls for collaboration and organize their email folders as it better suits their needs.

Developed components are kept in a separate machine that is accessible by all offices via VPN or via SVN. Details on involved developers, the technical leader and the manager are always included with component, however in a non-systematic way. Such information can be for instance included in the source code or in a component description document. The phone number and the contact details of all the involved people are to be found in the intranet-based address book. If developers have specific question on reusing the component, the manager is contacted. This latter forwards the request to the technical leader or to the more proper developer, who have the knowledge to answer the question.

If they need assistance, developers involved in the pilot write emails to a special address dummyProject@dummyDomain. A component guru replies to messages sent to this address. This workaround enables the separation of personal mails and component related mails, in this case JFreeChart. Documents describing special issues about the component, which were found by the guru are not communicated to the team. Knowledge discovered about particular component that should be reused is often kept on the mind of the component guru. This is due to the limited time of the guru and the reduction of information overload for developers. Experience showed that even written documents do not guarantee that all needed information is provided. The guru might be able to answer to a question taking into account more than one concrete situation he experienced during development.
3 Knowledge Sharing Approach and Expected Benefits

TEAM project aims to develop an open-source software system, seamlessly integrated in a software development environment for enabling decentralized, personalized and context-aware knowledge sharing. This will be achieved through:

- Lightweight annotation of development artifacts as well as knowledge manipulation features directly integrated in the software development environment.
- Capturing of the user’s behavior in the desktop environment and automatically triggering the knowledge manipulation.
- Analysis of interaction history and elicitation of a user profiles.
- Context-aware and ontology-based proximity search for relevant knowledge items.
- Semantic recommendation, through a proactive knowledge delivery depending on the actual working and personal context of the user.
- Ontology-based efficient structuring and persistent storage of acquired knowledge, as well as reasoning about its completeness and consistency.
- P2P-based decentralized communication.

By using the TEAM system TXT strive for improving knowledge sharing between its distributed software development teams. The TEAM system should make development activities more efficient and foster the collaboration. Developers should get recommendations and suggestion from the TEAM system, which make them save time in email communication with other developers or with components gurus.

Using TEAM, solutions to problems with potentially reusable library features should be shared only if this is required by other developers.

Known bugs as well as known ways to solve them are often just kept in the mind of developers that discovered them. Usually, this is not made explicit anywhere. With TEAM it should be possible to keep track of developers that are expert in certain features or certain parts of the library. Recommendations should be similar to normal developers brain. In knowledge need situations, developers remembers a) who was the person involved in a particular project, b) why he had that particular issues with the library to be reused and c) how he did solved this issue.

The TEAM system should gather information from existent repositories, in this case the mail archive. The TEAM system should help involved developers with recommendations without a need for consulting the guru. This will speed up the development process since the guru will not be a bottleneck any more. Developers are not blocked if the guru is, e.g. on holiday, ill or away for work.
Features similar to semantic recommendations or history analysis have not been tried in TXT environment yet. The developer perception of this new means of help is crucial, since the acceptance of the new system might influence the whole development processes. The acceptance of the TEAM system cannot be guaranteed due to the monitoring functionality. Thus realized privacy mechanism as well as the perceived added value for the developers should be clearly convincing. For example developers should have the control about the monitoring and just switch this off if desired.

The developers will use the TEAM system to facilitate knowledge exchange in reusing software artifacts. An example recommendation might be: ‘this interface of the component is the one you need to solve your problem’. It is intended to better understand what are minimum requirements a pilot project should give to the TEAM system, which is crucial for its further rollouts. Important functionality that can be only evaluated in such concrete situation includes: Does the TEAM system learn automatically? How much extra effort do developers require for running the TEAM system? What functionality and recommendation will be given to development teams?

4 Evaluation Criteria for Lightweight Knowledge Sharing

Table 2 shows the evaluation criteria as well as techniques that will be used to evaluate the perceived usefulness of TEAM functionalities within the pilot project. Reaching the success indicators described below will mean that developers working with the TEAM system perform better than working without it.

<table>
<thead>
<tr>
<th>Evaluation criteria</th>
<th>Technique to be used for measuring the performance indicator</th>
<th>Target value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of knowledge models (i.e. TEAM Ontologies)</td>
<td>Adequacy of these models as perceived by developers in the pilot project.</td>
<td>&gt; 75% of pilot project developers</td>
</tr>
<tr>
<td>Usefulness of elicited context</td>
<td>Perceived retrieval quality (controlled experiments with users, with and without context usage)</td>
<td>&gt; 90% pilot project developers acceptance</td>
</tr>
<tr>
<td>Quality of semantic search</td>
<td>Efficiency of retrieval algorithm (runtime efficiency)</td>
<td>Answer time &lt; 5 seconds</td>
</tr>
<tr>
<td></td>
<td>Perceived usefulness of full search functionality (user questionnaires)</td>
<td>&gt; 90% pilot project developers acceptance</td>
</tr>
<tr>
<td>Quality of semantic recommendation</td>
<td>Efficiency of recommendation algorithm (runtime efficiency)</td>
<td>Answer time &lt; 8 seconds</td>
</tr>
<tr>
<td></td>
<td>Perceived usefulness of recommendations (user questionnaires)</td>
<td>&gt; 80% accepted recommendations</td>
</tr>
<tr>
<td>Evaluation criteria</td>
<td>Technique to be used for measuring the performance indicator</td>
<td>Target value</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Distraction factor (user questionnaires)</td>
<td>&lt; 10% of notifications rejected by user</td>
<td></td>
</tr>
<tr>
<td>Perceived retrieval quality (controlled experiments with users, with and without history usage)</td>
<td>&gt; 80% user acceptance</td>
<td></td>
</tr>
<tr>
<td>User satisfaction with GUI concepts and design for annotation and knowledge manipulation</td>
<td>&gt; 90% user acceptance</td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper we described the as-is situation for collaboration and knowledge sharing in a distributed software development company. In a particular reuse-intensive project problems and needs with regard to current sharing practices—mainly file based—was derived. The research project TEAM aims at coping with these problems using semantic search and recommendation technologies as well as context awareness, i.e. observing and analysing developer’s interactions. Based on a developed evaluation framework, we described developers’ as well as company’s expectations from the TEAM approach. Next steps include conducting the pilot project as well as assessing the benefits and the consequences of the TEAM approach on the development practices.

6 Acknowledgement

This work was partly supported by the European Commission (IST-35111-TEAM).

References

[JF08] [http://sourceforge.net/project/showfiles.php?group_id=15494](http://sourceforge.net/project/showfiles.php?group_id=15494)
[Wi08] [http://desktop.google.com/dev/index.html](http://desktop.google.com/dev/index.html)
[BU08] [http://www.bugzilla.org/](http://www.bugzilla.org/)
[TE08] TEAM Project Homepage [http://www.team-project.eu](http://www.team-project.eu)