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

Alessandra Bagnato[1], Walid Maalej[2]

[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 the software development company TXT e-solutions plans to evaluate and deploy innovative results 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 initiatives.

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 due to 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 continues to emerge [S308]. Correct component integration, effective work with powerful frameworks or successful usage of design patterns, all require significant background and experience about used artifacts.

Moreover, the globalization of software development enables the formation of 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 versions ages rapidly. The research project TEAM [TE08] aims 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 innovations. In a real life development environment inside the company TXT e-solutions [TXT08], a pilot project has been selected with teams located 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 were € 56,3 million. TXT e-solutions is based in Milan with offices in Turin, Vicenza, Perugia, Bari, Roma and Genoa. The company owns subsidiaries in Paris, Lyon, London, Chemnitz, Halle, Haarlem, Barcelona and New York (Figure 1). Most development activities take place in the Italian offices. The company employs 500 persons, 250 of whom are software developers.

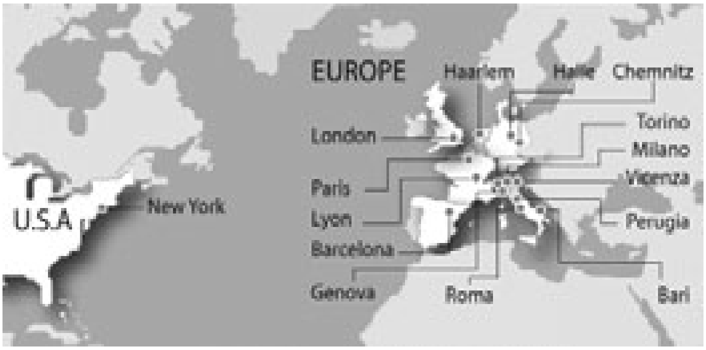

Figure 1: Distributed offices of TXT e-solutions

TXT e-solutions 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 offers specialized solutions dedicated to the primary needs of industry and finance.

At TXT e-solutions, the software development process is often distributed since the competencies are distributed among several offices. For a distributed project, there is a unique manager and one technical team leader for every office involved in the project. Technical team leader coordinates a team of 2-10 developers.

2.1 The Pilot Project

The usefulness of the TEAM conceptual models and knowledge sharing approaches for TXT e-solutions will be evaluated in the framework of a pilot project, where three distributed teams will develop a software application. Each team includes four developers working in three different locations in Italy: Genoa, Milan Bari. Genoa team presents a high expertise in developing graphical user interfaces, whereas Bari team is highly skilled in security matters and algorithms. Most customers are based in Milan.

The pilot project aims at creating a graphical application to assist the selection of persons that should be included in a team. It will comprise the following three main components:

- *Matching Engine*: This core component finds employees' profile that fits best to the 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 Identifier*: 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 planed to be reused in the pilot project. Figure 2 summarizes the main project organization, the components to be developed and the libraries to be reused.

Table 1: Libraries to be reused in the pilot project

LIBRARY	**DESCRIPTION**
JFREECHART 1.0.6 [JF08]	*JFreeChart* supports the creation of several charts types such as pie charts (2D and 3D), bar charts, line charts, time series charts, high-low-open-close charts, Gantt charts etc. *JFreeChart* can be used in applications, applets, servlets and JSP.
Google Widgets/Gadgets [Wi08]	*Web Widget* 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.

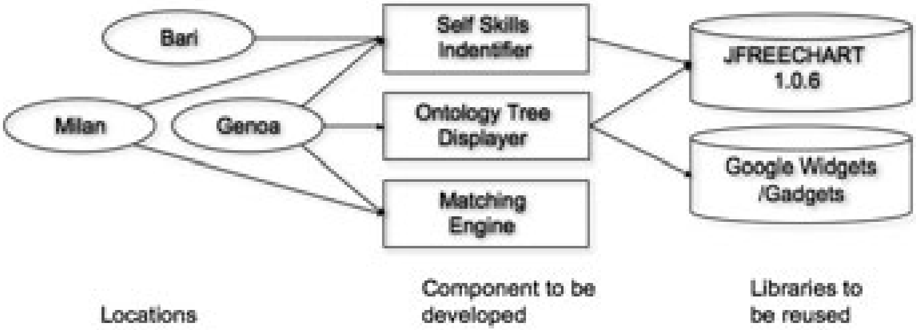

Figure 2: Organization, system architecture and libraries to be reused in the TEAM pilot project.

3 Current Practices of Collaboration and Knowledge Sharing

The various offices of TXT e-solutions make extensive use of a Virtual Private Network (VPN), enabling employees to share data, without realizing that the computer, with which they are working, is located in offices kilometers away. All employees including developers may view content shared in various company computers by means of shared folders. In distributed teams, developers tend to share folders on their own computers with 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 computers 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 held due to the experienced connection difficulties. On a weekly basis, a phone conference is organized with 4-5 participants. However calls are not sufficient in order to communicate and discuss complex models that are often used in software development. 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 from all offices via VPN or via SVN. Details on involved developers, the technical leader and the manager are always included together with the component, however in a non-systematic way. For instance, such information can be 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. In turn, the manager forwards the request to the technical leader or the most competent developer, who has the knowledge and expertise to answer the question.

If they need assistance, developers involved in the project send emails to a special address *dummyProject@dummyDomain*. A component guru replies to messages sent to this address. This workaround enables the separation of personal e-mails and e-mails related to the component, 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 a 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 a question taking into account more than one concrete situation he experienced during development.

4 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 e-solutions strive for improving knowledge sharing among its distributed software development teams. The TEAM system should make development activities more efficient and foster the collaboration among the developers. Developers should get recommendations and suggestions from the TEAM system, which save them time from email communication with other developers or with components gurus.

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

Known bugs as well as ways to solve them are often kept solely in the mind of the developers that discovered them. Usually, this knowledge 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 situations where specific knowledge is required, developers remember: a) who was the person involved in a particular project; b) why he had that particular issue with the library to be reused; and c) how he 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 recommendation or history analysis have not been tried in TXT software development environment yet. The developer's perception of this new means of assistance is crucial, since the acceptance of the new system might influence the whole development process. The acceptance of the TEAM system cannot be guaranteed due to the monitoring functionality. Thus, the implemented privacy protection mechanism as well as the perceived added value for the developers should be clearly convincing. For example, developers should have control over the monitoring functionality and just switch it 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 the minimum requirements that a pilot project should give to the TEAM system, which is crucial for its further rollout. 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 provided to development teams?

5 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. Meeting the success indicators described below will mean that developers working with the TEAM system perform better than working without it.

Table 2: Evaluation criteria for deploying the TEAM system in the pilot project

Evaluation criteria	**Technique to be used for measuring the performance indicator**	**Target value**
Quality of knowledge models (i.e. TEAM Ontologies)	Adequacy of these models as perceived by developers in the pilot project.	> 75% of pilot project developers
	Assessment of ontology quality according to [Go04]: Consistency, Completeness, Conciseness, Expandability and Sensitiveness.	
Usefulness of elicited context	Perceived retrieval quality (controlled experiments with users, with and without context usage)	> 90% pilot project developers acceptance
Quality of semantic search	Efficiency of retrieval algorithm (runtime efficiency)	Answer time < 5 seconds
	Perceived usefulness of full search functionality (user questionnaires)	> 90% pilot project developers acceptance
Quality of semantic recommendation	Efficiency of recommendation algorithm (runtime efficiency)	Answer time < 8 seconds
	Perceived usefulness of recommendations (user questionnaires)	> 80% accepted recommendations
	Distraction factor (user questionnaires)	< 10% of notifications rejected by user
Usefulness of history analysis	Perceived retrieval quality (controlled experiments with users, with and without history usage)	> 80% user acceptance
Usability	User satisfaction with GUI concepts and design for annotation and knowledge manipulation	> 90% user acceptance

6 Conclusion

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

Acknowledgement

We thank Maria Legal for her valuable comments on consistency and wording. This work was partly supported by the European Commission (IST-35111-TEAM).

References

[BU08] Bugzilla Homepage http://www.bugzilla.org/
[Ca99] Carmel, E. 1999, Global Software Teams: Collaborating Across Borders and Time
[Go04] A. Gómez-Pérez: Ontology Evaluation. In: R. Studer & S. Staab (eds.): Handbook on Ontologies, pp. 251-274. Springer-Verlag, 2004
[JF08] JFreeChart Homepage: http://sourceforge.net/project/showfiles.php?group_id=15494
[MR05] Minevich M. and F.-J. Richter, Global Outsourcing Report 2005, March 2005
[SK08] Skype Homepage http://www.skype.com/
[S308] The S3 initiative, A European Vision for Software, Services and System ftp://ftp.cordis.lu/pub/ist/docs/directorate_d/st-ds/visdoc.pdf
Zones, Prentice-Hall, Upper Saddle River, N.J., 1999
[TE08] TEAM Project Homepage http://www.team-project.eu
[TXT] TXT Group Homepage http://www.txtgroup.com/
[Wi08] Google Desktop API: http://desktop.google.com/dev/index.html
[YU08] A Skype collaboration tool http://www.yugma.com/