TEAM

Scope

Socio-economic developments around the world have among others seriously impacted the software engineering industry. Encountered with globalisation of the software process, and thus the geographical separation of personnel, companies are constantly seeking novel ways to increase the effectiveness and efficiency of their software development process. Software engineering is no longer the preserve of individuals but is essentially a team-based activity involving a wide variety of stakeholders and thus making the need for communication and co-operation an inherent characteristic.

The Team project addresses the need for a knowledge sharing environment with advanced capabilities suitable for the distributed development of software systems.

The Team project lays foundation for realizing a self-adapting, knowledgeable software development environment that captures developers' knowledge and preferences from their behaviour, discovers their information needs proactively and delivers the appropriate information support in the right context automatically. As a long-term vision, Team aims to contribute to the idea of an "intelligent and context-aware Integrated Development Environment".

Advances

Team addresses some of the most fundamental challenges of the new software engineering environment:

- The lack of time in searching for and creating new knowledge.
- The elusive nature of software itself resulting in less reuse than expected.
- The possessive attitude of the developers with respect to their knowledge and their reluctance in sharing it by fear that they will become expendable.

Team also challenges existing Integrated Development Environments (IDE) as most of these tools still cannot reflect the particularities of distributed development work.

The primary goal of the Team project is to support knowledge sharing in distributed software communities by applying semantic technologies. Team will develop a decentralized, personalized, context-sensitive and semantic-based framework for sharing knowledge about software implementation that is seamlessly integrated into a software development environment (IDE). Team will develop a toolkit in the form of an Open Source prototype ready to be deployed through the Eclipse IDE.

Positioning in global context

Team project is addressing five emerging research areas, including: (a) personalization & contextualization; (b) metadata management; (c) semantic search and recommendation, (d) knowledge desktops; and (e) semantic-based P2P systems. For all the above, Team has performed a thorough analysis in order to define the overall research landscape that is related to its objectives and position itself against the state-of-the-art. For the purpose of the present document, we focus on the Knowledge Desktop, which provides the user interface for accessing the integrated Team system functionality (e.g. search, recommendation) and also handles all other user interactions (e.g. notification of the user). We observed that the Team Knowledge Desktop is comparable to Haystack and IRIS in offering features for semantic search, suggestions of related items and tasks, but that unlike Haystack and IRIS, Team also offers support for social collaboration.

Contribution to standardization and interoperability issues

Since the Team project is primarily located at the intersection of Distributed Software Engineering, Knowledge Management, and Semantic Technologies, our major contributions to standardization are also positioned in this intersection area:

- Team stimulates efforts regarding Semantic Technologies in Software Engineering - The Team Consortium is actively represented in the W3C Software Engineering Task Force (SET) being part of the Semantic Web Best Practices and Deployment Working Group (http://www.w3.org/2001/sw/BestPractices/SE/)
- Team consolidates ongoing ontology-based metadata standardization for Web Service description.
- Team challenges and extends existing approaches for ontology-based context representation.
- Team supports Eclipse as a de facto standard.

Target users / sectors in business and society

The Team platform facilitates the transfer of knowledge, experience and lessons learned among knowledgeable developers of distributed **software companies** (especially SMEs) or open source projects. Moreover, Team targets **participants** of distributed development teams in an attempt to equalize their capabilities in the development of software systems, advancing the overall software systems development competences of the distributed team as a whole.

Overall benefits for business and society

The Team software system provides a proper abstraction mechanism to facilitate efficient communication among software development partners, enables proactive software engineering knowledge sharing and delivery, supporting and improving a recent business model of the software industry, software "nearshoring". The Team system will support effectively the virtual software community and ensure the production of more robust, flexible, high-quality software systems. The open source orientation of Team shall provide smaller software enterprises with new opportunities to drive their growth, and create a level playing field in the market for software SMEs.

Also open source development projects can benefit from Team and develop their software more efficiently. One of the validation scenarios of Team will be carried out by a community of open source developers.

Examples of use

Testing and Debugging

This area describes the case of sharing knowledge among developers how to cope with errors while programming an application. It is crucial for all software development organizations and many resources are dedicated for testing and debugging.

Example Scenario – An error is identified: A developer experiences an error (e.g. null-pointer) when running a test. The Team platform takes into account the relative knowledge resources including the Class/Program that runs, the input parameters, the Log messages, and the Console error output, (Debugger info). Then the Team platform can find out, if other developers encountered a similar problem and if a solution was found.

Reuse of components

Reusability of components is positively evaluated by all, but difficulties usually arise, especially in the cases that the reusable component is developed by third parties. Documentation and guidelines can be combined with hands-on experience and indication of when it can / should be reused. A mechanism may support the cooperation among the different teams and the feedback on the usage of components. In all cases the "explicitation" of the know-how of the component usage is highly appreciated.

Example Scenario - Usage of a component: A developer has to use a library/component. The Team platform takes into account the relative knowledge sources including example code snippets, documentation, comments from experts, design rationale (if available for internal component), filters this knowledge and decides which knowledge is suitable to supply.

Achievements

Team project is still in its early stages; nevertheless important achievements could be reported:

- State-of-the-art analysis of all topics related to the Team's objectives. More specifically, a series of state-of-the-art analyses has been performed for the following areas: (a) Knowledge Management is Software Engineering, (b) Personalization and Contextualization in Software Engineering, (c) Metadata Management, (d) Search Mechanism and Context Similarity, (e) Semantic Wikis and Semantic Desktops, (f) Semantic Recommenders, and (g) Semantic-based P2P systems.
- Team toolkit architectural design (1st version). This will provide the basis for the development of the first version of the running prototype.
- 1st iteration prototypes for the Team components (i.e. Knowledge Desktop, Context Observer, Metadata Storage, Search and Recommendation, P2P).

All project public deliverables can be downloaded from the project website: http://www.team-project.eu/