Collaboration Spaces for Virtual Software Teams

Geetanjali Takane

Department of Computer Science and Engineering, SVERI's College of Engineering, Pandharpur

Third Year Engineering Student

Software engineering is a field in which distributed development through virtual teams is a fact of life. Thus, environments for supporting virtual software teams should place collaboration at the forefront. A set of eight core requirements for support environments derived from, and validated in, industrial settings address how to provide virtual software teams with a sufficient level of awareness for their work activities.

OVERVIEW:

This article is based on our work on boosting the advantages of global software engineering by neutralizing certain disadvantages via technological solutions. We performed this research iteratively, meaning we first studied one aspect, including an empirical evaluation, and then directly applied findings and lessons learned. We began by investigating the effects of enabling the overhearing of conversations in virtual software teams.

THE DAYS of software engineering taking place in a single office building are long gone; in this dynamic field, virtual software teams are a fact of life. However, bringing distance into software engineering has had an enormous impact on teams and the work itself: distance matters.

Nearly everyone with experience in virtual software teams has encountered some challenges. If software engineering were a purely technical, one-man job, we could cope with these challenges much more easily. However, this is a strongly collaborative profession. Social activities represent a considerable portion of the average day of software engineers, who mainly perform four sorts of daily activities:

- coding,
- organizing workspaces and processes,
- representing and communicating design decisions and ideas, and
- communicating and negotiating with various stakeholders.

Collaboration is downright essential to software teams. Unsurprisingly, virtual software teams have started to develop tools to address these challenges. Today, many environments exist to support virtual software teams, most of which include some support for collaboration. However, these tools have different origins, with different initial goals in mind—compare, for example, integrated development environments such as Eclipse and Microsoft

Visual Studio to project hosting sites such as Source Forge and Google Code and centralized build systems such as Apache Continuum and Cruise Control. All of these tools have been extended with collaborative capabilities;

Because collaboration is so essential to virtual software teams, we claim that their support environments should place collaboration at the forefront to be successful. Companies such as Atlassian are starting to do this, but we think mobile phone support and related features are only initial steps, and much more is to be gained.

Refocusing Support Environments

In other words, existing support environments for virtual software teams focus mainly on individual programming tasks; any collaborative functionalities are available mostly in isolated (special- purpose) environments, such as a code editor that supports a chat function or a separate debugger that shows which colleagues are using that tool at that point in time. Distance can’t be changed through tools, but the perception of it can be greatly reduced.

For instance, virtual teams that have a Google Hangout open all day might be physically distributed, but they won’t perceive a large distance within the team. In our research (see the sidebar for an overview), we worked on increasing awareness among virtual software teams through tools.

Ideally, tools that support virtual teams need to bring engineers to the same (or a higher) degree of awareness as when they’re collocated. When this happens, teams can truly collaborate on a global field, perhaps even making the term *virtual* superfluous.

Requirements for Virtual Team Support Environments

What we need is a single platform that both supports all the awareness needs of virtual softwareteams and integrates awareness information from different information sources. But how can we make such an environment a reality? Over the past four years, we worked with real-life virtual software teams to identify their needs and validate possible solutions. We condensed our results into a set of eight core requirements that address how to provide virtual software teams with a higher level of awareness, thereby helping them function as a team more effectively.

Requirement 1: Enable Unobtrusive Awareness Information Exchange

In a traditional collocated setting, awareness is achieved without much effort, but in a distributed one, engineers must manually analyze, filter, and combine available information.

Tools to support virtual software teams should enable a comparable level of awareness, a goal that’s feasible because a system can “know” what task you’re working on, who previously edited your file, and whether colleagues can be interrupted. Your system could ideally provide you with contextualized information without you actively searching for it or disturbing you.

Requirement 2: Make Basic Work-Related Data Available

Obviously, support environments need to make necessary information available, but the biggest challenge

- To prevent workers from becoming overloaded, it’s important to filter the data they get, is to ensure that it’s available at the appropriate time. Therefore, such systems should be built on a deep knowledge of the software engineering profession. Our results showed that virtual software team members consider a large and diverse set of information to be important as long as it’s directly related to their current project.

Requirement 3: Provide Multisource Data Combinations

To collaborate effectively, team members need to combine information from different sources; through automation, support environments can relieve team members from taking on this burden personally. that encourage creative, healthy, and high-bandwidth modes of communication among a project’s stakeholders.

Requirement 4: Filter Irrelevant Information

To prevent workers from becoming overloaded, it’s important to filter the data they get. Environments for virtual software teams should recognize automatically what information is relevant to the current activity. They should also contain a functionality that automatically recognizes when individuals can be interrupted to provide this information.

We investigated the concept of how virtual office walls can be used to contextualize information based on someone’s current activity and provided empirical evidence that this approach was indeed an effective method for supporting people in performing their tasks. We also investigated the prioritization of different types of information and how it changes based on current activity.

Requirement 5: Represent and Recognize Current Contexts of Team Members

Support environments must be able to recognize a team member’s currentcontext so that they can filter and provide only the information that’s valuable to that person at a specific time. Automatically recognizing a team member’s current task can be done by assuming that he or
she is working on the task personally selected from the digital task board.

Requirement 6: Support the Overhearing of Conversations

We discovered various benefits and challenges to overhearing conversations in virtual software teams—specifically, the information and useful actions in those conversations. One of the most important benefits of overhearing conversations is having access to your colleagues’ technical knowledge. At the same time, the greatest challenge is that overhearing your colleagues talk can be a distraction, and conversational context can be unclear.

The most important information about a conversation is its topic, so a support environment should help identify or track it, even if it changes midstream. Furthermore, it should be possible to initiate, participate in, discover, watch, and _ nish conversations, invite others to join them, and make conversations private (not “overhearable”).

- Distance can’t be changed through tools, but the perception of it can be greatly reduced

OVERVIEW OF UNDERLYING

Requirement 7: Support Mood Sharing

Being able to have an idea of your teammates' mood and overall happiness is important to virtual software teams. We found evidence for this when we investigated the use of a system that supports microblogging with mood indicators. In particular, we found that being able to express a small amount of information together with their associated mood made people feel more connected with one other on a social level.

This evidence was reaffirmed when we evaluated a system that we developed and designed with collaboration at its core. In this later evaluation, knowing the happiness of colleagues was identified as one of the system's two most valuable aspects.

Requirement 8: Provide for Interruption Support

Virtual software team members need a variety of information about the context in which they're working to collaborate effectively with their colleagues. Support environments have the potential to regulate that information based on both its importance and the current degree of "interruptibility" of team members. To strike this balance, support systems must consider the noises and distractions that software engineers face, even when they're collocated.

Challenges remain in providing virtual software teams with a support environment that enables the same or higher degree of awareness found in a collocated setting. When the support environment reaches that point, distance will no longer be an issue. We expect a major step in this direction when tool suppliers change their focus from a coding orientation to a collaboration one.

We imagine a future in which team members automatically receive the information they need when they need it and without disrupting them in their core activity. When this future is attained, the degree of awareness will be lifted far beyond being collocated.