

Auto-Erecting Virtual Office Walls

A Controlled Experiment

Ben van Gameren
Delft University of Technology
IHomer
b.j.a.vangameren@tudelft.nl

Rini van Solingen
Delft University of Technology
Prowareness
d.m.vansolingen@tudelft.nl

Kevin Dullemond
Delft University of Technology
IHomer
k.dullemond@tudelft.nl

Abstract—A Virtual office wall is a mechanism which automatically regulates information to support distributed software engineers. These walls reduce the available information to only that information which is currently relevant. In this paper we present a controlled experiment with experienced software engineers as study participants. In this experiment we study whether there is a relation between the presence of virtual office walls and the actual and perceived speed and accuracy of the work carried out by the participants. Additionally, we measured the extent in which the participants experience the presence of virtual office walls as useful. The main findings include that virtual office walls appear to contribute to an improved awareness of co-worker synchronicity, an easier understanding of things to do and a more concise overview of the work performed. These improvements mostly seem to benefit the speed of coordination and the perception regarding overall performance.

I. INTRODUCTION

In a co-located setting, software engineers are confronted with all kinds of information, noises, distractions, etc. that are not relevant to their work at hand. These distractions can be so severe a complete workday becomes ineffective [1]. Dislocated software engineers have the advantage that tooling has the potential to filter the information delivered to them. Such tooling has access to information on what everyone is working on and can as such filter non-relevant information. This resembles the creation of 'moving' office walls in a co-located fashion that move around all the time, depending on the work an engineer is carrying out. Such 'auto-erecting virtual office walls' are unfeasible in real life but can be created in tooling for (globally) distributed software engineers. As such, these walls provide dislocated software engineers with the awareness level of a 'virtual office', undisturbed by information not relevant to their current activity.

The main contribution of this paper is the indication that virtual office walls are valuable to (dislocated) software engineers. In practice this comes down to the actual speed and accuracy of their work, as well as perceptions of speed, accuracy and usefulness. Furthermore, an important contribution of this work is the construction of a controlled experimental design with tasks, resources and materials that can be used for replicating and scaling of the experiment.

Research on virtual office walls is interesting because it enables dislocated software engineers to better focus on the actual work and prevents distractions and manually searching for required information.

This research, however, also brings an additional dimension to the research on distributed Software Engineering. Tooling for distributed Software Engineering is largely set up with the motivation to compensate for the negative consequences of distance [2]. However, such tools also provide opportunities to further build upon. A real-life physical wall is not able to be moved easily and has no understanding of the work an office occupant is working on. Tools for dislocated software engineers do possess such knowledge and can, as such, leverage this. So, the research addressed in this paper also looks into the additional benefits and added value that can be delivered by removing the physical boundaries of Software Engineering work offices.

This paper is structured as follows. First in section II we discuss related work regarding virtual office walls. Following this, in section III we discuss the research questions and hypotheses of the experiment. Subsequently, in section IV, we present the design of the experiment and discuss the variables and tooling used in this experiment. In section V we present the findings and evaluate the hypotheses based on these findings. Next, in section VI, we discuss the threats to the validity of this study. Finally, we present the conclusions of this research and discuss future research in section VII.

II. RELATED WORK

Software Engineering is a highly collaborative activity in which knowledge about the context in which you are working is essential to properly collaborate with others [3], [4]. In literature this knowledge is commonly referred to as 'awareness' [5], [3]. Examples of such information items are: information about the other members of the project team, their activities and information about the progress of the project. It is essential to have a sufficient level of awareness, since software engineers need to coordinate their efforts to be able to produce a functional system.

In the traditional co-located setting this information is exchanged relatively passively and unobtrusively [3], [6], so engineers are continuously aware of information related to their current activity [7]. In [8] we discussed this is probably caused by the design of the office building [9], since in general an office building consists of several rooms each with its own characteristics. By moving around in the building developers are able to select a room which characteristics match their needs, and as such are able to change the context of their activities.

But nowadays, both due to the globalization of business [10], [11], [12] and because people are starting to work from home more and more [13], people no longer share a physical work environment. In such a distributed setting exchanging awareness information without technological support becomes unfeasible [7]. So, in order to retrieve information related to their current activity developers need to use technological solutions. To fulfill this need, the (Global) Software Engineering community has developed and studied a wide variety of tools, for example: Instant Messaging solutions, issue management systems and configuration management systems [14], [15].

Most of these tools, however, only support a single aspect of the development process and as a consequence many diverse tools are needed to provide software engineers with the information they need [14]. When they have finally gathered all relevant information, this information needs to be analyzed, combined and filtered manually by each developer to acquire the information necessary to create a context for his current activity. This process can be quite time-consuming and may result in misunderstandings, inconsistencies, incompatibilities and duplicated information [14].

In [8] we discuss how best to provide distributed software engineers with the awareness information they need. We state that to be able to acquire awareness in a relatively passive and unobtrusive fashion, such as in the co-located setting, the analytical process of accessing, combining and filtering information needs to be automated. In essence we need a mechanism which automatically regulates information based on the current context of a software engineer: a *'virtual office wall'* [8].

III. RESEARCH QUESTIONS AND HYPOTHESES

The goal of this study is to find out how valuable virtual office walls are for real-life distributed software engineers during their day-to-day activities. To determine this value, we measure the extent in which this kind of technical support impacts the speed and accuracy of the work, the extent in which it impacts the perception of speed and accuracy of the work, and the extent in which experienced distributed software engineers consider this kind of support useful.

To reach this goal we have formulated the following research questions for this experiment:

- RQ1** How do virtual office walls influence the speed of work?
- RQ2** How do virtual office walls influence the perception of the speed of work?
- RQ3** How do virtual office walls influence the accuracy of the work carried out?
- RQ4** How do virtual office walls influence the perception of the accuracy of the work carried out?
- RQ5** How useful is the introduction of virtual office walls in a (distributed) Software Engineering project?
- RQ6** Do virtual office walls make it easier to understand what is going on in a Software Engineering project?

In order to answer these research questions we have chosen to perform a controlled experiment with experienced distributed software engineers as study participants. The reasons we chose to conduct a controlled experiment are:

- We intended to find evidence that the introduction of virtual office walls is valuable. In order to exclude any other influence we decided to set-up a controlled experiment in which the only variable is the availability of such functionality. All other variables are kept constant
- We intended to find evidence that the actual and perceived performance differs when the amount of information is limited (as already indicated by Solingen et al. [16] and based on the research of Prickladnicki [17])

Considering these two intentions a controlled experiment with real-life software engineers with distributed experience is the best approach. Before undertaking the experiment we formulated the following hypotheses regarding the above six research questions:

- H1** The introduction of virtual office walls has a positive impact on the speed of work carried out
- H2** The introduction of virtual office walls has a positive impact on the perception of the speed of work
- H3** The introduction of virtual office walls has no impact on the accuracy of the work carried out
- H4** The introduction of virtual office walls has a strong positive impact on the perception of accuracy of the work
- H5** Software engineers consider the introduction of virtual office walls a useful feature
- H6** The introduction of virtual office walls makes it easier to differentiate between information that is relevant to the current activity of an engineer and information that is not.

Hypotheses 1 and 2 express that the speed of work will go up, because the reduced amount of information makes it easier to find out what to do. Hypothesis 3 expresses that the accuracy of the work will not be different as this is largely determined by the specific task itself and the specific skills of each individual engineer. Hypothesis 4, however, expresses that the perception of accuracy will be influenced because the more difficult it is to see what is happening the larger the probability one might feel things are not going well. Hypotheses 5 and 6 express that providing distributed software engineers with the (awareness) information they need is considered useful and makes it easier to understand the current status of the project.

IV. CONTROLLED EXPERIMENT

We conduct a controlled experiment to study whether there is a relation between the presence of virtual office walls and the actual and perceived speed and accuracy of the work carried out by distributed software engineers. Additionally we measured the extent in which the participants experience the

presence of virtual office walls as useful. In a controlled experiment the results obtained from two samples are compared; the results obtained from a test group and the results obtained from a control group. These two groups are practically identical to each other, except for the one aspect of which the effect is being tested [18]. In this section we discuss the design of the experiment, the dependent, independent and control variables, the tooling environment used, and the context in which the experiment is conducted.

A. Design

In this controlled experiment we examine the impact virtual office walls have on actual and perceived speed and accuracy. Additionally, we measured the extent in which participants experience the presence of virtual office walls as useful. Therefore we split the total group of study participants into two subgroups; a test group which has access to an environment in which the concept of virtual office walls is implemented and a control group which has access to an environment in which this concept is disabled. Such an experiment is referred to as a 'One factor with two treatments experiment' [18]. The distribution of participants into these groups is random but takes into account their level of expertise. We included this dimension in the selection to ensure that the two groups would be as similar as possible.

During the experiment both groups have to successfully complete six projects, called Blue, Green, Orange, Purple, Red and Yellow. Each of these projects consists of: (i) three randomly assigned project members (participants), made anonymous by changing their names; each participant is only allowed to work on the projects he or she is assigned to, (ii) twelve project specific tasks, called Task 1 through 12, each task has a status; Open, In Progress or Resolved, a description and a corresponding resource, and (iii) two project specific resources¹, called after a city or country e.g. Limerick or Egypt, each resource consists of a status; Locked or Unlocked, and the location of the resource.

Participants of the experiment should use the following process to successfully complete each of the six projects (See figure 1 for an overview):

- 1) *Selecting a Task:* A participant should select an open task of one of the projects he or she is assigned to. Subsequently, the engineer should verify that all tasks on which this task depends are resolved.
- 2) *Selecting a Resource:* When the participant has selected a task, he or she has to verify the status of the corresponding resource. If the status of the resource is unlocked the participant, can lock the resource. However, when the status of the resource is locked, the participant should select another task.
- 3) *Locking a Resource:* Before a participant is able to work on a task he or she should first lock the corresponding resource. If the status of the resource is locked, the participant should select another task.
- 4) *Assigning to a Task:* When a participant has locked the resource corresponding to the selected task he or she should assign this task to himself or herself.

- 5) *Working on a Task:* When a participant has both assigned himself or herself to a task and locked the resource corresponding to this task, he or she is able to start working on the task. The participant first needs to download the resource from a central repository, subsequently, the participant can perform the task instruction, and finally the participant needs to save and upload the file to the central repository.
- 6) *Resolving a Task:* When a participant completed the task on which her or she was working, the participant should update the status of the task to resolved.
- 7) *Unlocking a Resource:* When a participant resolved the task to which he or she assigned himself or herself, the participant should also update the status of the corresponding resource to unlocked, so other participants have the ability to lock this resource.

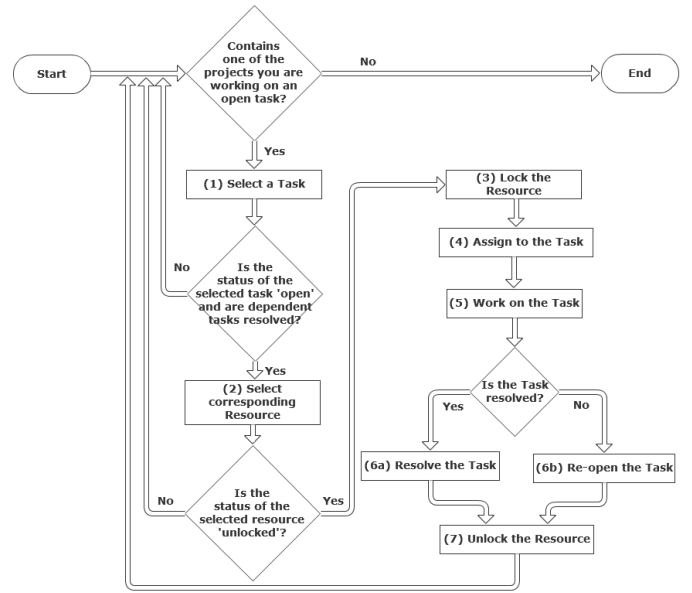


Fig. 1. Process to successfully complete each project

This process is repeated for each of the 72 tasks of the experiment. When all tasks are completed the experiment stops. Then a questionnaire is distributed to all participants with additional questions to gather quantitative and qualitative data on their perceptions of speed, accuracy and usefulness of virtual office walls.

We need an alternative to Software Engineering tasks, since completing a regular software project is unfeasible due to the implicit time constraints of a controlled experiment. We decided to use simple tasks because they minimize mistakes due to differences in software programming abilities of the participant, they require the type of knowledge that needs to be exchanged in software tasks [19], and they have been used in previous experimental studies [19], [16]. The simple tasks we use are the modified fictional map tasks adopted by Espinosa et al. [19]. This type of tasks mimic important aspects of global Software Engineering teams including [19]: (i) shared goals, (ii) interdependent activities and skills, (iii) the need for effective communication, and (iv) the need to articulate and interpret requirements correctly.

Furthermore, we decided to exclude inter-participant com-

¹Despite what is common in practice, we consider the term resource to be limited to the materials necessary to carry out the tasks

munication from the controlled experiment. We did this because communication is likely to have a too strong impact on the outcomes of the experiment. The only way to keep things as constant as possible in this experiment is to exclude inter-participant communication. When participants are unable to communicate directly with each other, all coordination actions take place in the environment in which the concept of virtual office walls is either implemented (test group) or not (control group). Therefore we modified the map tasks adopted by Espinosa et al. [19] to remove the need for communication and to decrease the ambiguity of the requirements.

B. Dependent, Independent and Control Variables

In this section we discuss the three types of variables used in this experiment. An independent variable (factor) is a variable that is manipulated in the experiment. The values or settings for an independent variable are the test conditions. The impact of the different test conditions will be measured by analyzing the dependent variables. Finally, we have circumstances that might influence a dependent variable but are not being investigated. These are called control variables and need to be controlled to limit the variability of the measures.

In this experiment only one variable is changed, the support environment and we defined two test conditions for this dependent variable: an environment in which (i) the concept of virtual office walls is enabled and (ii) the concept of virtual office walls is disabled.

Next to the independent variable we also defined 6 dependent variables:

- *Actual speed:* We measure the time (in seconds) it takes for participants to successfully complete each of the 72 tasks
- *Perceived speed:* We measure the perceived speed by asking participants how they would grade the overall speed of work on a 5 point likert scale (very low, low, normal, high, very high) with a no-opinion option
- *Actual accuracy:* We measure the accuracy of the performed work by dividing the number of correct elements by the total number of elements
- *Perceived accuracy:* We measure the perceived accuracy by asking participants how they would grade the overall quality of the work on a 5 point likert scale (very low, low, normal, high, very high) with a no-opinion option
- *Ease of use of the system:* We measure the ease of use of the system by asking participants how they would grade the use of the system on a 5 point likert scale (very difficult, difficult, normal, easy, very easy) with a no-opinion option
- *Usefulness of the system:* We measure the usefulness of the system by asking participants how they would grade the overall support on a 5 point likert scale (very low, low, normal, high, very high) with a no-opinion option

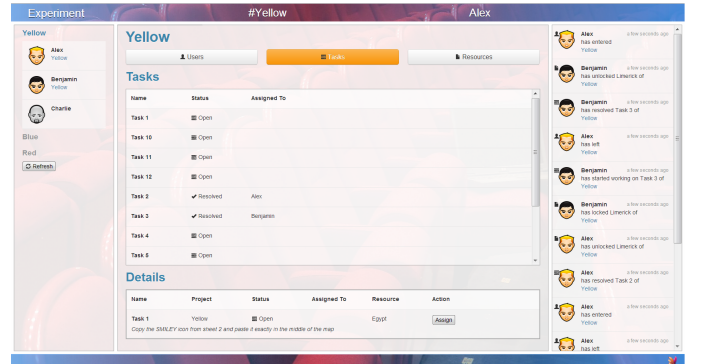
There also exist six variables which need to be kept constant between the two groups of the experiment: users, projects,

maps, tasks, resources and tools. Finally, one randomized variable exist: the assignment of tasks to a participant, as such each user can complete an unequal number of tasks.

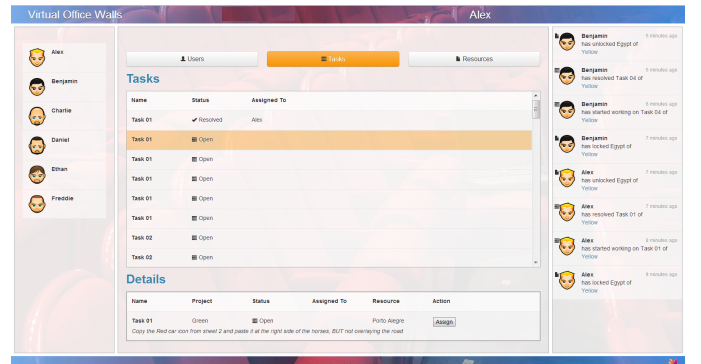
C. Tooling Environment

In this section, we discuss the environment we have used during the experiment to enable or disable the concept of virtual office walls. In this discussion we present an overview of this environment.

Because we use the same environment for both groups (we only change whether or not the concept of virtual office walls is enabled (See figure 2)), the influence differences in design could have on the outcome of the experiment is minimized. On the left section of the user interface information about team members is shown, in the middle section of the view both an overview and detailed information about the users, tasks and resources is shown, and on the right section information about the actions participants performed is shown. The only difference between the two groups is that the test group has access to only that information which is relevant to the project they currently work on while the control group has access to the information about all projects.



(a) Environment with the concept of virtual office walls enabled



(b) Environment with the concept of virtual office walls disabled

Fig. 2. Tooling environment

Having discussed the main difference between the two groups, we continue by illustrating the effects this has on quantity and relevance of the accessible information about users, projects, tasks, resources and actions. Firstly, we discuss differences in information shown about the fellow team members of an user (the left section of the view). In figure 3a the information provided to the participants of the test group

is shown and in figure 3b the information provided to the control group is shown. Comparing these two figures, it can be seen that participants in the control group have access to information about all other users, while participants in the test group only have access to information of the project they are currently working on (Project Yellow). This filter on active project makes it possible to enrich the visualization of the test group by showing which of the team members are currently working on the project. Because participants of the test group only have access to information related to their active project, a mechanism is needed to change this context. This can be done by selecting another project in the left section of the view (e.g. Blue). In this view only projects are shown in which you participate, all other projects are filtered out.

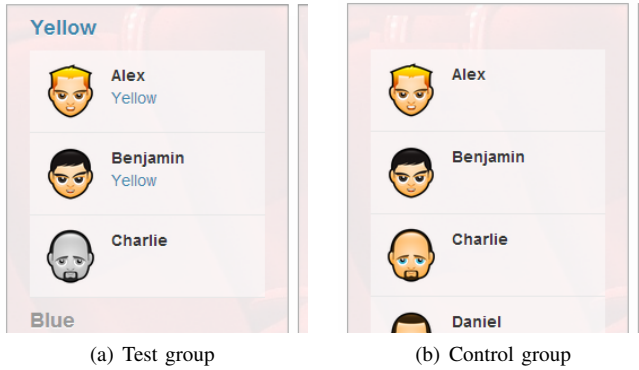
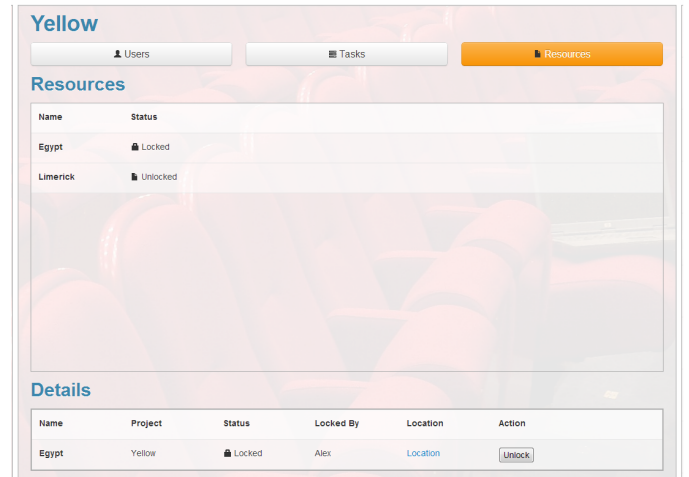


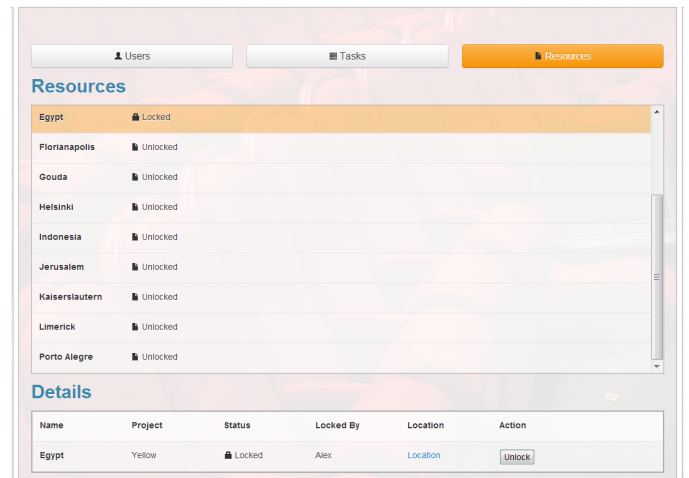
Fig. 3. Team members

Secondly, we discuss the middle section of the view in which an overview and detailed information about the users, tasks and resources is shown. Again, the main difference is that participants in the control group have access to information about all projects, while participants in the test group only have access to information about the project they are currently working on. We will illustrate the consequences of this approach by discussing the impact it has on the resources (materials necessary to carry out a task). However, the same argumentation holds for both the users and the tasks. Figure 4a and 4b both show information about the status of resources. However, in figure 4a only resources of project Yellow are shown.

Finally, we discuss the right section of the view in which information about the actions participants performed is shown (The time-line). Again, participants in the control group have access to all actions, while participants in the test group only have access to actions related to the project they are currently working on. During the experiment participants perform actions to successfully complete the projects, for example locking a file and resolving a task. The following actions are shown on the time-line of both groups: locking a resource, unlocking a resource, start working on a task, stop working on a task and resolving a task. As a consequence participants in the control group get to see actions from a wide variety of projects (See figure 5b), while participant in the test group only see actions related to the project they are currently working on (See figure 5a). Next to the actions which appear on both time-lines participants of the test group also get to see actions of project members who either enter or leave the project space (By



(a) Test group (With virtual office walls)



(b) Control group (Without virtual office walls)

Fig. 4. An overview and detailed information

changing their active project), providing additional information about their active project.

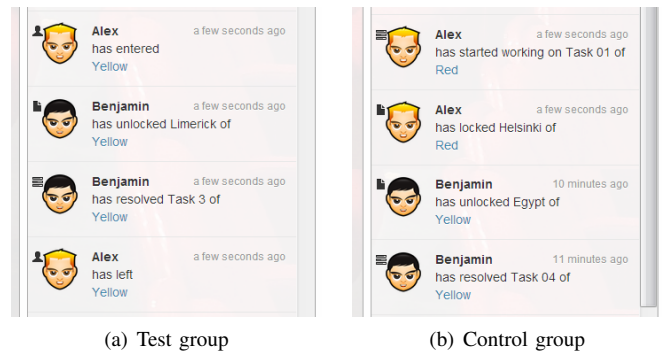


Fig. 5. Actions performed by team members

D. Context, execution and assignments

This controlled experiment is conducted at IHomer, a Dutch Software Engineering company founded in August of

2008. The company currently employs 21 employees and is fully distributed, since the default work location of the employees is their home. As a consequence, all employees are experienced with dealing with the difficulties of developing software when working physically separated from each other. Even though it is common practice to work from home, the employees try to get together at least once a week to stay connected. This distributed nature makes employees of this company particularly suitable as study participants for this experiment. Employees participated voluntarily in the experiment. In total 12 employees participated in the experiment, divided into two groups of 6 participants, one group which has access to an environment in which the concept of virtual office walls is enabled (Test group) and one group which has access to an environment in which this concept is disabled (Control group). The distribution of participants into these groups is random after accounting for their level of expertise. We included this dimension in the selection process to ensure that the two groups would be as similar as possible.

The experiment itself was conducted on a single day. During this day two runs were executed, one by the test group and one by the control group. The first run took place between 9:00 AM and 10:30 AM, in this run we used the environment in which the concept of virtual office walls was enabled (Test group). The second run took place in the afternoon, between 2:00 PM and 3:30 PM, in this run we used the environment in which this concept was disabled (Control group). To ensure participants of the control group were not influenced by participants of the test group, we asked them not to talk about the experiment until both runs were finished. As far as we know all participants did this. Because of the distributed nature of both the company and experiment we decided to execute both runs with participants working from their home. Before each run was executed, the participants were gathered in a Google Hangout² in which the objective of the experiment and the tasks were explained. Subsequently, a 15 minute demo was given to the participants by one of the researchers, in which the process of successfully completing a task was demonstrated. Finally, the participants were allowed to enter the environment and they were asked to leave the hangout (so they could not communicate with each other). When all projects were completed we again gathered all participants in a Google Hangout to thank them for their participation, and to distribute the questionnaire. During both runs, two of the authors were available to provide help and prevent undesired interactions to occur.

We conclude this section by giving an impression what actions are necessary to successfully complete each task, see figure 1. We show a scenario in which all actions of the process are demonstrated. In this scenario we discuss all available information types, all necessary actions, and the tooling used to complete each action in contrast to the more general discussion in section A.

1. Selecting a Task: Before a participant is able to work on a specific task, he³ first has to select an open task of one of the projects he is assigned to. To find out to which projects a participant is assigned, he should use the tooling environment

discussed in the previous section. In this environment he can access all information about a user, see figure 6: (i) the name of the user, we changed the names of the participants to Alex, Benjamin, Charlie, Daniel, Ethan and Freddie to enable anonymous participation, (ii) the projects he is assigned to, a user is always assigned to three of the following projects: Blue, Green, Orange, Purple, Red and Yellow, (iii) the project he is currently working on, (iv) the task he is currently working on, and (v) the resource he is currently locking.

Name	Projects	Current Project	Current Task	Current Resource
 Alex	Yellow, Red, Blue	I Homer		

Fig. 6. User details

When a participant knows to which projects he is assigned, he can select an open task of one of these projects. Again the participant should use the tooling environment to determine the current state of the selected task. Each task is in one of the following three states: Open, In Progress and Resolved. When the state of the task is Resolved, the task is successfully completed and no further actions are required. When the state of the task is In Progress, the task is currently being carried out by one of the participants. Finally, when the state of the task is Open, work has to be carried out to complete this task. Next to the current state of the task also (i) the name, ranging from Task 1 through 12, (ii) the corresponding project, (iii) the description, (iv) the participant who is currently working on the task or the participant that has resolved the task, and (v) the corresponding resource are shown, see figure 7.



Name	Project	Status	Assigned To	Resource	Action
Task 01	Green	 Open		Porto Alegre	
Copy the Red car icon from sheet 2 and paste it at the right side of the horses, BUT not overlaying the road					

Fig. 7. Task details

2. Selecting a Resource: When a participant has selected an open task of which all tasks on which it depends are resolved, he has to verify the status of the corresponding resource. A resource can only have two states: unlocked and locked. When the state of the resource is locked, another participant locked the resource and as a consequence the participant has to select another task. When the status of the resource is unlocked, the participant can lock the resource and start working on it. Next to the state of the resource also, (i) the name, either called after a city or country, (ii) the corresponding project, (iii) the participant who has currently locked the resource, and (iv) the location of the resource are shown, see figure 8.



Name	Project	Status	Locked By	Location	Action
Porto Alegre	Green	 Unlocked			

Fig. 8. Resource details

3. Locking a Resource: Now a participant has selected both an appropriate task and resource, he should use the environment to actually lock the resource indicating he is currently working on that resource.

4. Assigning to a Task: When a participant has locked the resource belonging to the selected task, he should use the

²<http://www.google.com/+/learnmore/hangouts>

³All participants of the experiment were male, so when talking about participants of the experiment we will use 'he'

environment to actually assign the selected task to himself indicating he is currently working on that task.

5. *Working on a Task*: When a participant has both assigned himself to a task and locked the resource corresponding to this task, he is able to start working on the task. He first needs to download the resource from a central repository to acquire the file on which he needs to work. In this experiment we use Dropbox⁴ as repository, because of its easy to use web interface and because it enables a history of all versions of a single file. Secondly, the participant has to edit the file (resource) he just downloaded either by using PowerPoint⁵ or Impress⁶. Each file consist of two parts, a map and a set of objects (See figure 9). By adding objects and arrows to the map, according to the description of the current task, a modified version of the file is created. In contrast to the map requirements of Espinosa et al. [19] we more explicitly defined the tasks and tried to remove ambiguity from them since inter-participant communication was not allowed to resolve such issues. Therefore we decided on the following categories of task instructions:

- 1) *Adding an object*: Tasks in this category include descriptions to copy and replace an object from the set of objects to the map. For example: *"Copy the telephone object from the set of objects to the map and paste it directly above the factory"*
- 2) *Drawing an arrow*: Tasks in this category include descriptions to add an arrow to the map. For each arrow multiple properties are specified like, the starting point, ending point, color, weight, and style of the arrow, either solid or dashed. An example of such an task description is: *"Draw a green dashed arrow of width 5pt from the telephone object to the most right horse"*.
- 3) *Copying and moving an arrow*: Tasks in this category include descriptions to copy and move an arrow already drawn on the map e.g. *"Copy and paste the green arrow and move it, starting at the most right horse to the left of the bulldozer"*.

Each task only includes a single task instruction. Finally, when the participant has completed the current task, see figure 10 for an example of a modified resource, he needs to save and upload the file to the repository (Dropbox).

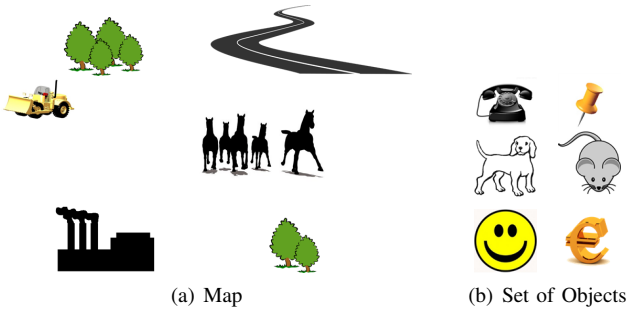


Fig. 9. Example of a resource

⁴<http://dropbox.com>

⁵<http://office.microsoft.com/powerpoint/>

⁶<http://libreoffice.org/impress/>

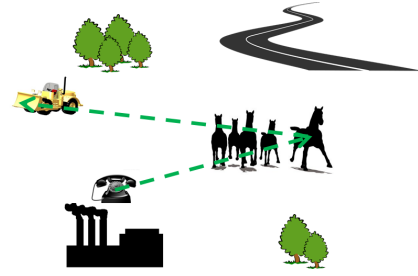


Fig. 10. Example of a modified resource

6. *Resolving or re-open a Task*: When a participant has completed the task to which he assigned himself, he should update the status of the task to resolved. However, if the participant has not completed the task and decides to work on another task, he should re-open the task.

7. *Unlocking a Resource*: When a participant has either resolved or re-opened the task to which he assigned himself, he should also update the status of the resource to unlocked, so other team members can lock this resource.

V. FINDINGS

In this section the main findings of this study are presented. We discuss the the six research questions regarding the accuracy, speed and usefulness of virtual office walls. For each of these domains we will present quantitative results from the experiment and questionnaire, and qualitative results from the questionnaire. The results of the data are visualized in different ways: we use (i) histograms to illustrate the distribution of the data, and (ii) stacked bars without the neutral data to illustrate the ratio between the positive and negative responses. Finally we also evaluate the hypotheses based on the findings of the experiment and questionnaire.

A. Speed of the work performed

We measured the influence of virtual office walls on the actual speed of work by measuring the time it took the participants to successfully complete all projects of the experiment. The control group needed 39 minutes to complete these projects, while the test group only needed 35 minutes to complete their work. By comparing these results, it can be seen that the introduction of virtual office increases the speed of work by almost 10%. Since we also measured the time participants needed to perform their tasks (so actually performing the instruction of the task), we can also calculate the time they needed to coordinate their work. The control group needed 19 minutes to coordinate their work while the test group only needed 15 minutes. This difference makes the impact of the introduction of virtual office walls even more obvious, since the time needed to coordinate work was decreased by approximately 20%.

The results of the questionnaire were used to analyze the perception of the speed of work. We asked each participant to grade the speed of the work performed by himself, the work performed by his team members, and the work performed by the whole team. The results of the test group were slightly more positive than the results of the control group. Since participants in the test group have not reported any negative

values at all and some of them indicated the perception of speed to be very high. Participants of the control group, however, reported several negative values and none of them rated the perception of speed very high, see figure 11. One of the participants of the control group gave as feedback: [Charlie] *"It seemed that I was searching a lot for a task that had an unlocked file associated to it. My team members where locking files that I needed to edit"*.

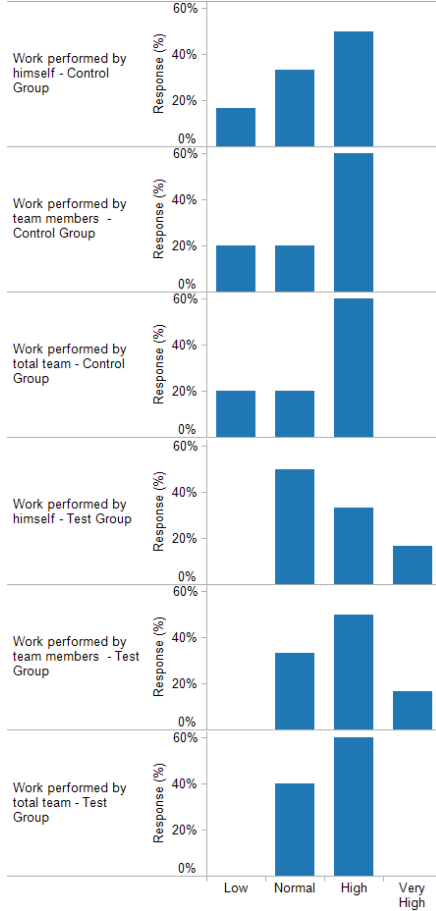


Fig. 11. Perception of the speed of work

However, we can not accept or reject our hypotheses regarding the speed of work because of the small sample size of the experiment. We have, however, some indications that the introduction of virtual office walls has decreased the time needed to coordinate work and as such has a positive impact on the total speed of work (**H1**). Additionally, we found some indicators corresponding to the hypothesis that virtual office walls have a strong positive impact on the perception of the speed of work (**H2**).

B. Accuracy of the work performed

Next to measuring the speed of work, we also measured the accuracy. The metric used to verify the accuracy of the work performed, is dividing the number of correct elements by the total number of elements. Both the test group and the control group have performed their tasks without making a mistake. As such, both groups have an accuracy of 100%. The removal of ambiguous requirements from the original experiment appears

to be the reason for this. Note that this was done to control communication by excluding it from the experiment.

The perception of the work accuracy, however, can not be derived by analyzing the results of the performed tasks. Therefore, we asked each participant to grade the accuracy of the work performed by himself, the work performed by his team members, and the work performed by the whole team. From these results it can be seen that in general the control group is more negative about the accuracy of the work performed than the test group, see figure 12.

These findings seem to correspond with both hypotheses regarding accuracy: (**H3**) the introduction of virtual office walls has no impact on the accuracy of the work carried out, and (**H4**) the introduction of virtual office walls has a strong positive impact on the perception of accuracy of the work carried out. Again, the sample size of the experiment is too small to either reject or accept these hypotheses.

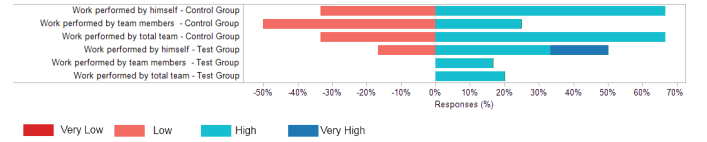


Fig. 12. Perception of the work accuracy

What is perhaps most striking about the data we gathered about the perception of work accuracy, is the difference in the number of *No-Opinions* between these groups. Five participants of the control group have indicated they were not able to grade the accuracy on a scale ranging from very low to very high, against only one participant of the test group. Several of the participants of the control group explain their choice, for example: [Ethan] *"I was unable to grade the work that my team had done, since I did not know what their assignments were, or what the goal was"* and [Freddie] *"I have not checked the accuracy of my team members, I opened the file to execute the task but I have not checked back at older tasks to see if the file reflects the right state"*. This is striking because it indicates participants of the control group have a low level of awareness about the accuracy of work carried out.

C. Usefulness of virtual office walls

Finally, we measured both the usefulness and the ease of use of virtual office walls. To analyze the usefulness of virtual office walls we asked the participants of the test group to grade the usefulness of (i) the integration of the information, (ii) being able to see the current context of your project members, the project they are currently working on, and (iii) the differentiation between information that was helpful to your current task and information that was not. We only considered the test group in this analysis, because this group has actually experienced working with virtual office walls and as such is able to give a well argued opinion of the usefulness of this concept. The results of these opinions (see figure 13), provide strong indications that context based filtering of awareness information is useful, since most participant graded the usefulness in the range normal to very high.

Next to the usefulness of virtual office walls we also asked participants of both groups to grade how easy or difficult it

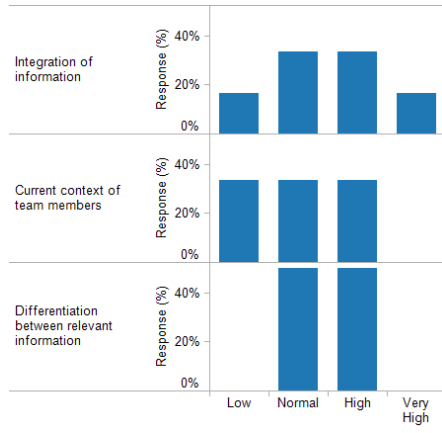


Fig. 13. Usefulness of virtual office walls

was to (i) see the active project of their team members and (ii) differentiate between information that was helpful to their current task and information that was not. The outcomes of these questions are depicted in figure 14. It is obvious that participants in the test group find it much easier to acquire the active context of their team members and to differentiate between relevant and non-relevant information. One of the

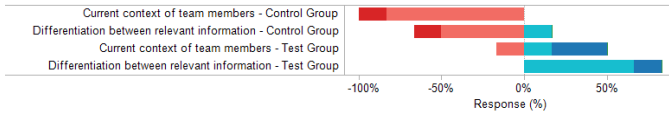


Fig. 14. Ease of use of virtual office walls

participants of the control group, Daniel, indicated he had the feeling there was a "Lock race" going on as there was no way to determine who would carry out which task on what resource.

These findings seem to correspond with the hypotheses regarding the usefulness and the ease of use of virtual office walls. Because, experienced software engineers indicated they considered the introduction of virtual office walls to be useful (H5). Participant also indicated that it was easy to differentiate between information that is relevant and information that is not, and that the introduction of these walls makes it easier to understand which other project members are working at the same time at the same project (H6).

VI. THREATS TO VALIDITY

In this section we discuss the threats to validity for this experiment on four aspects: construct validity, internal validity, external validity and conclusion validity.

Construct validity regards the extent in which the variables measured actually measure the constructs of interest. The main threat to construct validity of this experiment is that we use the modification and manipulation of PowerPoint maps instead of real programming work. One could argue that our study does not investigate distributed Software Engineering, but investigates distributed PowerPoint editing. However, it improves on internal validity by eliminating differences in individual programming skills, and improves throughput time.

We measured the speed by checking the time stamps of the events in the repository. Next, the accuracy was measured by comparing the outcomes to the predefined results. However, to measure the perception of the speed, accuracy and usefulness we have to ask the participant of their opinion. One could argue that in general it is difficult to measure opinions. As such we used simple metrics on a 5-point likert scale and asked directly for the perceived effects.

Internal validity is especially relevant in studies that attempt to identify a causal relationship, such as this study. The question regarding internal validity is whether the observed effect was actually caused by the researched factor. In this experiment we managed to keep the experimental conditions relatively stable, considering we undertook the experiment in a real-life company. However, we need to express that the engineers had the tool seen demonstrated. As such this previous experience (without the virtual office wall functionality) might have influenced their opinions. Furthermore, the control group and the test group consisted of different people which can have influenced the results as well. Combined with the limited number of participants, our findings are only of an indicative nature.

External validity is of interest in studies that want to draw generalized conclusions. Although, experiments are in general highly externally valid [18], for this specific study it is too early to make generic claims. First of all, the sample size is too small to draw statistically significant conclusions. Second of all, because the type of work the participants performed (placing and moving objects on a map) and the average duration of the tasks only an approximation of the actual work of software engineers is given.

Finally, conclusion validity expresses the extent in which the intervention (providing the virtual office walls) actually led to the observed outcome, and as such questions the reliability of the studies' conclusions. We acknowledge that the small number of samples and data points does not allow us to draw causal conclusions. This is however, a known problem, in experimental studies in Software Engineering teams [20].

VII. CONCLUSIONS AND FUTURE WORK

We conducted this research to discover how valuable virtual office walls are to distributed software engineers. To reach this goal we performed a controlled experiment. In this experiment we looked at the accuracy and speed of the work performed and perception of these. Furthermore we investigated the extent in which experienced distributed software engineers consider virtual office walls to be useful.

The main findings of the study are the following:

- 1) Virtual office walls make work coordination easier, because they assist in differentiating between information that is relevant to the current activity of an engineer and information that is not
- 2) Virtual office walls contribute to an increased perception on overall performance.

Even though, the data we gathered is not statistically significant due to the small sample size, it is interesting to see how these two main benefits are related to each other. This is more clearly shown in figure 15. In this figure each

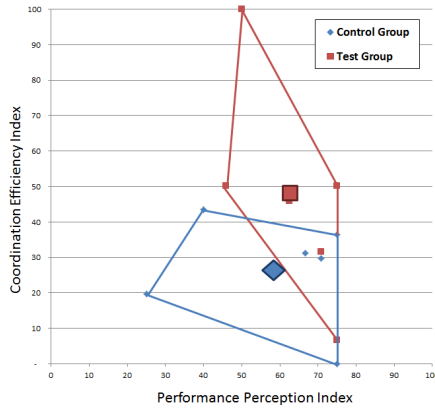


Fig. 15. Coordination - Performance system

data point represents the results in the experiment of one of the participants along two axes: the '*coordination efficiency index*' on the y-axis, the '*performance perception index*' on the x-axis. The '*coordination efficiency index*' is the normalized time the participant took to perform the coordination portion of the experiment. This value is normalized to give the most efficient participant the score '100' and the least efficient participant the score '0'. The '*performance perception index*' is the normalized performance perception the participant reported on the 5-point Likert scale. This value is normalized so the maximum possible score is '100' (6 times very high') and the minimum score is '0' (6 times very low). Furthermore, for each of the two groups the outliers are connected to create a tetragon to illustrate the space it encloses. Finally, the average values, roughly at the center of the the tetragons, is depicted as a fat data point. In the figure it can be seen that the test group overall scores higher than the control group on both metrics. It will be interesting to investigate this with a much larger sample to size to see if this conclusion will continue to hold.

Furthermore, the data collected in the questionnaire following the experiment also indicated the overall usefulness of the virtual office wall concept. In general, distributed software engineers are overloaded with an abundance of information, such as information available on their systems, mailbox and tools, so every improvement in helping them cope will be welcomed with open arms. And doesn't that hold true for all of us? Imagine a world in which our e-mail would configure itself to only include mails related to our current task. What if our computers would notify us if our colleague is editing the same code as us or recommend us who to ask for help when we are struggling with something? Wouldnt our lives be much less complicated if our working environment would adapt itself to us, instead of the other way around?

This is the promise of mechanisms such as virtual office walls and with the study presented in this paper we have shown indications there is value in pursuing this further.

VIII. ACKNOWLEDGMENT

We would like to acknowledge that we reused the materials from Espinosa, Nan and Carmel from [19] in this experiment. We would like to thank them for letting us reuse their materials.

REFERENCES

- [1] R. van Solingen, E. Berghout, and F. van Latum, "Interrupts: Just a minute never is," *Software, IEEE*, vol. 15, no. 5, pp. 97–103, 1998.
- [2] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, vol. 18, no. 2, pp. 22–29, 2001.
- [3] K. Schmidt, "The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'," *Computer Supported Cooperative Work*, vol. 11, no. 3-4, pp. 285 – 298, 2002.
- [4] A. Syri, "Tailoring cooperation support through mediators," in *Proceedings of the 1997 European Conference on Computer Supported Cooperative Work*. Kluwer Academic Publishers, 1997, pp. 157–172.
- [5] P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces," in *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work*. ACM Press, 1992, pp. 107–114.
- [6] J. Fogarty, S. Hudson, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 119–146, 2005.
- [7] K. Dullemond, B. van Gameren, and R. van Solingen, "Virtual open conversation spaces: Towards improved awareness in a GSE setting," in *Proceedings of the 2010 International Conference on Global Software Engineering*. IEEE Computer Society Press, 2010, pp. 247–256.
- [8] B. van Gameren, K. Dullemond, and R. van Solingen, "Auto-erecting virtual office walls," in *Proceedings of CollaborateCom*, 2012.
- [9] T. Allen and G. Henn, *The organization and architecture of innovation: Managing the flow of technology*. Butterworth-Heinemann, 2007.
- [10] E. Carmel, *Global software teams: collaborating across borders and time zones*. Upper Saddle River: Prentice Hall PTR, 1999.
- [11] J. Herbsleb and D. Moitra, "Guest Editors' Introduction: Global Software Development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.
- [12] J. Herbsleb, "Global Software Engineering: The Future of Socio-technical Coordination," in *Proceedings of the IEEE 2007 Workshop on the Future of Software Engineering*. IEEE Computer Society Press, 2007, pp. 188–198.
- [13] The Dieringer Research Group Inc., "Telework Trendlines 2009: A Survey Brief by WorldatWork," 2009.
- [14] J. Portillo-Rodríguez, A. Vizcano, M. Piattini, and S. Beecham, "Tools used in global software engineering: A systematic mapping review," *Information and Software Technology*, vol. 54, no. 7, pp. 663 – 685, 2012.
- [15] A. Sarma, A. Van der Hoek, and D. Redmiles, "The coordination pyramid: A perspective on the state of the art in coordination technology," *Computer*, vol. PP, no. 99, p. 1, 2010.
- [16] R. van Solingen and M. Valkema, "The impact of number of sites in a follow the sun setting on the actual and perceived working speed and accuracy: A controlled experiment," in *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*. IEEE, 2010, pp. 165–174.
- [17] R. Prikladnicki, "Exploring Propinquity in Global Software Engineering," *2009 Fourth IEEE International Conference on Global Software Engineering*, pp. 133–142, 2009.
- [18] C. Wohlin, P. Runeson, M. Host, C. Ohlsson, B. Regnell, and A. Wesslén, "Experimentation in software engineering: an introduction," 2000.
- [19] J. A. Espinosa, N. Nan, and E. Carmel, "Do gradations of time zone separation make a difference in performance? a first laboratory study," in *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*. IEEE, 2007, pp. 12–22.
- [20] F. Calefato, F. Lanubile, T. Conte, and R. Prikladnicki, "Assessing the impact of real-time machine translation on requirements meetings: a replicated experiment," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM, 2012, pp. 251–260.