# When to Interrupt Global Software Engineers to Provide them with What Information?

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

***Abstract*—Software Engineering is a highly collaborative activity in which knowledge about the work context is essential to collaborate effectively. Acquiring such knowledge is difficult in a distributed setting, since developers have to manually analyze, filter and combine available information in order to acquire a sufficient level of awareness. Therefore, it seems beneficial to construct a mechanism which automatically regulates information based on both the current activity of a software engineer and the importance of the new information. In this paper we present an Estimate-Talk-Estimate study, with experienced software engineers, in which we studied both (i) what information software engineers want to know immediately and (ii) when software engineers do not mind to be interrupted with such information. The main findings include a list of information items which software engineers want to be immediately informed about, and a list of activities during which software engineers prefer not to be interrupted.**

***Keywords*—*Global Software Engineering, Awareness, Virtual Office Walls, Information, Interruptions, Estimate-Talk-Estimate Study***

## I. Introduction

Software engineers need information about the context in which they are working to be able to collaborate effectively with their colleagues [1], [2]. In the traditional co-located setting all information is available in a single place, the office building, and is exchanged relatively passively and unobtrusively between all employees present at that location [1], [3], [4]. In a distributed setting, however, this information can only be exchanged by using technological support. The (Global) Software Engineering community has developed many technological solutions to support globally dispersed teams in performing their tasks [5], [6]. However, most of these solutions only support a specific type of information and this information cannot be processed by other solutions directly [5]. Therefore, global software engineers have to manually analyze, filter and combine available information to acquire a sufficient level of awareness.

To exchange awareness information relatively passively and unobtrusively in a distributed setting as well, this analytical process of accessing, combining and filtering available information needs to be automated [3]. In essence we need a mechanism which regulates information based on the context it encloses: *'Virtual Office Walls'* [3]. Such *'Virtual Office Walls'* have the potential to filter the information, noises and distractions software engineers face in a co-located setting. These auto-erecting walls are unfeasible in real life, but can be created in tooling for (global) software engineers, since such tooling has access to information on what everyone is working on. In this paper we focus on what information software engineers want to know immediately and during which activities they prefer not to be interrupted. As such the goal of this paper is:

*"To find out how to regulate information available to software engineers based on both the importance of that information and the current interruptibility of the engineer".*

This paper is structured as follows. First in section II we discuss background information and related work of this research. Following this, in section III we discuss the method of data collection we used in this study. Subsequently, we present the findings and results in section IV and V. In section VI, we reflect on these findings and discuss the most important results. Next, in section VII, we discuss the threats to the validity of this study. Finally, we present the conclusions of this research and discuss future research in section VIII.

## II. Background and 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 [1], [2]. Having access to such knowledge, in literature referred to as *'awareness'* [7], is essential since software engineers need to coordinate their effort to produce a functional system. In the traditional co-located setting this information is exchanged relatively passively and unobtrusively [1], [4], so much of this information is naturally propagated to all the members of the team. In a distributed setting, however, sharing such information becomes unfeasible without technological support [8]. Therefore, the (Global) Software Engineering community has developed many solutions to provide globally dispersed teams with all the information they might need. Both [5] and [9] provide an overview of some of the tools used for Global Software Engineering. These developments have led to one of the main challenges in the context of GSE, the lack of integration [5].

When all information is integrated into a single solution, software engineers need to abstract useful information without experiencing an overload of information. Tell et al. [10] propose to use the Activity Theory to both structure and describe activities in software engineering processes. This makes it possible to determine what information is relevant when performing a specific collaborative task. However, this

COLLABORATECOM 2013, October 20-23, Austin, United States

DOI 10.4108/icst.collaboratecom.2013.254099

does not imply software engineers should be immediately informed of this information. Because interrupting software developers during their work can significantly reduce a developer's efficiency [11].

## III. Method

The goal of this study is: *"To find out how to regulate information available to software engineers based on both the importance of that information and the current interruptibility of the engineer"*. To be able to reach this goal we identified the following research questions:

RQ 1. *"What information do software engineers want to know immediately?"*

RQ 2. *"During which activities do software engineers prefer to be interrupted to provide them with information?"*

The outcomes of these two research questions could be contradicting. Since, on the one hand, it is likely participants indicate that they are interested in direct updates of information. On the other hand, it is also likely that they prefer not be interrupted at that specific time. Therefore we are also interested in the following research question:

RQ 3. *"What information do software engineers want to know immediately, even though they are performing an activity during which they prefer not to be interrupted?"*

Fig. 1 provides a visual representation of these three research questions, in which their mutual relationship is shown.

These research questions span a wide range of domains, since we are interested in (i) information items useful to software engineers, (ii) common activities of software engineers, (iii) the distinction between information of which software engineers do want or do not want to be immediately informed, and finally (iv) the distinction between activities during which software engineers prefer to be or prefer not to be interrupted.

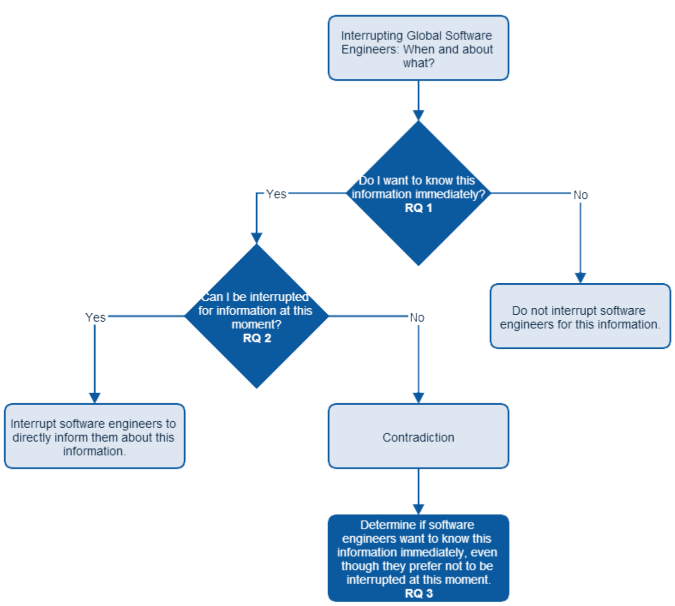



Figure 1. Mutual relationship of the research questions

Because the focus of this research is on differentiating between what information software engineers do want or do not want to know immediately and differentiating between activities during which software engineers prefer to be or prefer not to be interrupted, we decided to use the information items and activities defined in a generic life cycle model. We have looked at several of these models, including the CMMI for development, ISO 12207. IEEE 1074 and MIL 498, and could not identify significant differences considering the goal of this study. We chose to use the CMMI for development model[12], because its practice oriented design and because we expect the used terminology best matches with the industrial experts. In the CMMI for development five process areas are defined: Requirements Development, Technical Solution, Verification, Validation and Product Integration. These process areas consists of multiple specific goals which all describe an unique characteristic that must be present to satisfy this area. These specific goals in turn, consist of multiple practices which are important activities to achieve the associated goal.

Now we have a structured list of information items and activities, we have to find out what information software engineers want to know immediately and during which activities they do not mind to be interrupted. Therefore, we use a structured communication technique which allows study participants to systematically deal with these issues. We use the Estimate-Talk-Estimate method [13]. The main reason to use this method is that decisions made by a structured group of individuals are more accurate than individual judgments [13]. Another reason to use this method is that a combination of nominal and interacting group processes is desirable in judgmental problem solving [14], [15], [13]. In the remainder of this section we describe the criteria used to select study participants and the process we used to gather the empirical data.

### A. Members of the group of participants

Participants in a Estimate-Talk-Estimate study are asked to provide reasons for their decisions and to respond to the decisions made by the other participants. Based on this information participants can revise their opinions. Therefore, it is essential the members of this expert study have different backgrounds so they can provide each other with new information[16]. This makes choosing the appropriate subjects an important step in the entire process because it directly relates to the quality of the results generated [17].

We used the following criteria to select the members of the group of participants:

1) At least five years of experience in global software engineering
2) Currently working as a software engineer (e.g. architect, tester, designer or programmer)
3) Currently working at least half of the time on software engineering tasks
4) Masters the English language
5) Ability to conceptually argue about global software engineering

Selecting study participants based on one ore more characteristics is called purposive sampling [18]. We initially send out an invitation to 37 software engineers who met these criteria.

TABLE I
MEMBERS OF THE GROUP OF PARTICIPANTS

| Expert | Role | Professional background |
|---|---|---|
| 1 | Technical Lead | Working in a large team of about 50 people spanning locations in the Netherlands, India, Scotland and Ukraine |
| 2 | Engineer | Working in a small distributed team of 3 people |
| 3 | Engineer | Working in a small distributed team of 6 developers mainly working from home. |
| 4 | Technical Lead | Working at a large department with over 200 people, spanning multiple locations: The Netherlands, Belgium, USA, Ukraine and Malaysia |
| 5 | Project Manager | Working in a large team of about 60 people in Amsterdam, working together with departments in New York, San Jose, and Kiev |
| 6 | Engineer | Working in a small distributed spanning two locations: Delft and Kiev |
| 7 | Architect | Working in a large team of about 50 people spanning four locations: Delft, Moscow, Houston and Albuquerque |
| 8 | Engineer | Working in a small distributed teams ranging from 3 to 6 developers |
| 9 | Engineer | Working in a small distributed team of 7 people working from home 1 or 2 days a week |
| 10 | Architect | Working in a small distributed team of 6 developers mainly working from home |

We also asked them to forward the invite to other software engineers that meet this profile, however we only allowed up to two participants of the same company. Finally, ten participants were willing to participate in this study, see table I. This number of participants corresponds to the number of experts recommended for a Delphi panel [17].

### B. *Estimate-Talk-Estimate approach*

An Estimate-Talk-Estimate study [13] has much in common with a regular Delphi study [19]. Both studies rely on a panel of experts who answer multiple questions in two or more rounds. After each round the forecasts of the experts and the reasoning behind their judgments are discussed to encourage the experts to revise their earlier answers based on the argumentation of others. The goal of both studies is to reach consensus on a predefined list of items. The main difference between these two studies is that in a Delphi study a facilitator provides an anonymous summary of the expert's judgments after which the experts can revise their judgments anonymously, while in the Estimate-Talk-Estimate study all judgments are visible to everyone and the experts discuss their judgments with each other. We have chosen to use an Estimate-Talk-Estimate approach since it is expected that interactive group processes contribute to a higher quality of the estimates. In their research Gustafson et al. [13] emphasize written feedback appears to lead to a reduction in the quality of estimates.

The Estimate-Talk-Estimate study we conducted consists of multiple rounds to reach consensus, see Fig. 2. During *'Round 1'*, the members of the expert group remotely completed a


Figure 2. Rounds of the study

questionnaire[1]. The questionnaire consists of two parts. In the fist half, participants were asked to order the practices from most important to least important and distinguish between practices of which they want to be or not want to be immediately informed. In the second half the participants were asked to order the practices based on how disturbing an interruption would be and distinguish between practices during which they prefer to be or prefer not to be interrupted. After this round classifications of practices were accepted if at least 80% of the experts agreed with each other. So, even if a majority of participants, either six out of ten or seven out of ten, agreed with each other we did not accepted that classification as strong enough.

The resulting practices, on which no consensus was reached, were considered in *'Round 2'*. *'Round 2'*,*'Round 3'* and *'Round 4'* consist of two parts and were conducted during a two-hour meeting at the Delft University of Technology in the Netherlands. During this meeting both authors were present. One of the authors took the role of moderator while the other took notes. The second round started with a detailed description[2] of the practice at hand, to clarify its meaning. Next, the experts were asked to classify this practice. To reduce the risk of influencing the other participants, we used the planning poker procedure [20]. Each of the experts lays a card face down, a zero (*'no'*) or an one (*'yes'*), representing their estimate. Next, they all simultaneously call out their card by turning them over. Again classifications are accepted if at least 80% of the experts agreed with each other. *'Round 3'* and *'Round 4'* started with a time-boxed discussion in which both arguments in favor of and arguments against being informed immediately or being interrupted were discussed. During these discussions, the moderator had to ensure that everyone participated and had a chance to speak [21]. To stimulate an evenly contribution of the participants, we first provided the minority of the group the opportunity to explain their judgments, after which the majority of the group had the chance to explain their judgments *'Round 3'*. In *'Round 4'* first the majority of the group was given the opportunity to convince the others, after which the minority of the group had the chance to respond. Both rounds were concluded by a re-estimation of the current practice. Table II outlines, the total number of practices, the number of practices on which consensus was reached, and the number of practices on which no consensus was reached for each of the four rounds.

Finally, in the *'Post-Round'* the study participants remotely completed a questionnaire[3]. In this questionnaire we asked some general questions, process related questions and research related questions about the introduced contradictions. We asked the participants to indicate both (i) what information they want to know immediately, even though they are performing

TABLE II
RESULTS OF THE ESTIMATE-TALK-ESTIMATE STUDY

| | *Round 1* | *Round 2* | *Round 3* | *Round 4* |
|---|---|---|---|---|
| Number of practices | 80 | 53 | 22 | 13 |
| Accepted practices | 27 | 31 | 9 | 3 |
| Undecided practices | 53 | 22 | 13 | 10 |

[1] See http://aspic.nl/vow/questionnaire.pdf
[2] See http://aspic.nl/vow/practices.pdf
[3] See http://aspic.nl/vow/post-questionnaire.pdf

an activity during which they prefer not to be interrupted, and (ii) whether or not they prefer to be interrupted with information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted. The participants already reached consensus on six of the twelve practices after this initial round (*'Post-Round'*).

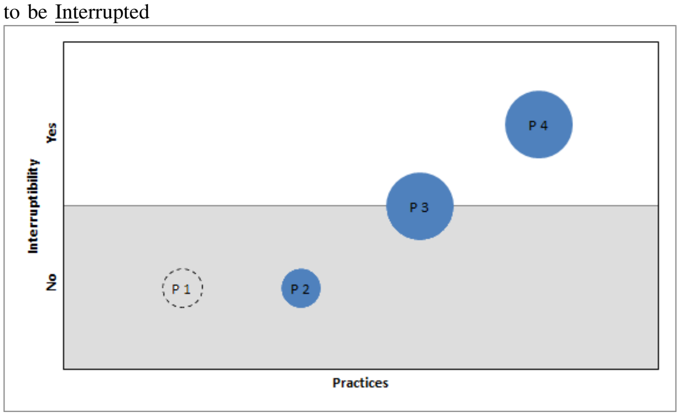
## IV. Findings Estimate-Talk-Estimate Study

To structure the findings of the Estimate-Talk-Estimate study, we use the engineering process areas defined in the CMMI, namely, Requirements Development, Technical Solution, Verification, Validation and Product Integration [12]. For each of these process areas we briefly describe its purpose, goals and practices. Subsequently, we discuss for each of the practices (i) whether or not software engineers want to be informed of such information immediately and (ii) whether or not software engineers prefer to be interrupted while performing activities corresponding to the practice at hand. The results of these classifications are presented in a uniform fashion to provide the reader with a clear overview of the results. A complete overview of the classifications of all rounds can be found online[4]. An illustrative example of such a representation is shown in Fig. 3.

This figure shows some illustrative classifications we have made for some of the activities you perform in the morning before carpooling to work with a colleague. This representation consists of two parts. The first part is a table summarizing both the results of what information about your colleague you want to know immediately, and the results of the classifications of activities during which you do not mind to be interrupted. In the second column the results of what information about your colleague you want to know immediately is shown. In this column a *'✓-sign'* indicates that the participants agreed they want to be informed immediately of information regarding the

| Practice | Inf | Int |
|---|---|---|
| P 1. Waking up | ? | ✗ |
| P 2. Taking a shower | ✗ | ✗ |
| P 3. Having breakfast | ✓ | ? |
| P 4. Leaving your home to go to work | ✓ | ✓ |

(a) Table containing both [Inf] Information you want to know immediately when carpooling to work, and [Int] Activities during which you do not mind to be Interrupted




(b) Visualization of the results

Figure 3. Illustrative classification of morning activities before carpooling to work with a colleague

[4]See http://aspic.nl/vow/classifications.pdf

associated practice. A *'✗-sign'*, however, indicates they do not want to be informed immediately. When the participants did not agree with each other and have not reached consensus a *'?-sign'* is used to indicate this. In the third column the results of the activities during which software engineers do not mind to be interrupted are summarized. A *'✓-sign'* indicates that they agreed they do not mind to be interrupted at that moment, a *'✗-sign'* indicates thy prefer not to be interrupted and a *'?-sign'* again indicates they did not reach consensus. The second part of this representation is a figure which visualizes the results of both classifications. In this figure each circle represents a single practice. The radius of the circle is used to depict whether or not the experts want to be immediately informed of information regarding the practice. A small radius indicates that the experts do not want to be immediately informed of that practice. A large radius, however, indicates that the experts do want to be immediately informed of that practice. When the circle has a small radius, is not filled and has a dashed border, the participants did not agree with each other. The location of the circle, in turn, depicts whether or not you prefer to be interrupted. A circle in the upper area concerns a practice during which you do not mind to be interrupted. A circle in the bottom area concerns a practice during which you prefer not to be interrupted. When a circle is placed on the border between these two areas the participants did not reach consensus about that practice.

### A. Requirements Development

The first engineering process area we discuss is *'Requirements Development'*. The purpose of this area is to elicit, analyze and establish customer, product and product component requirements [12]. The practices of this area belong to one of the following three goals:

SG 1 *Develop Customer Requirements*
Stakeholder needs are translated into a set of customer requirements

SG 2 *Develop Product Requirements*
Customer requirements are translated into a set of product requirements

SG 3 *Analyze and Validate Requirements*
Analyze and validate both the customer and product requirements with respect to the end user his intended environment

*1) Information updates:* Firstly, we asked the participants to order the practices based on importance, regardless of their current activity, and differentiate between the practices of which they want to be informed immediately and the practices of which they do not. The results of these classifications, after four rounds, are shown in Fig. 4a. In this table it can been seen that participants of the study indicated that they only want to be informed of new information regarding the establishment of product and product component requirements (SP 2.1). Furthermore, it can be seen that participants do not have to be informed immediately of new information regarding practice SP 1.1, SP 1.2, SP 2.2, SP 3.1, SP 3.2, SP 3.3 and SP 3.5. Finally, the participants did not reach consensus on SP 2.3 and SP 3.4.

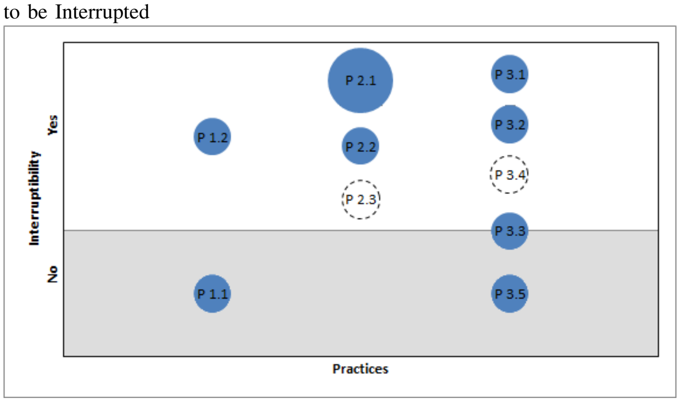
During the face-to-face meeting a extensive discussion took place about practice SP 2.3 on which no consensus was reached. The arguments used in favor of being immediately informed about identifying interface requirements focus on the urgency of this kind of information. One of the participants said: *"I want to be informed immediately when such information becomes available"*. Arguments used against this practice focus on the fact that this is not time crucial information, which can be illustrated by the following statement: *"This kind of information can wait"*. Overall the participants indicated that whether or not they want to be informed of such information strongly depends on the impact of the specific interface.

*2) Interruptibility:* Subsequently, we asked the participants to order the practices based on how disturbing an interruption would be while performing activities corresponding to that practice, regardless of the content of the interruption. Again we asked the participants to differentiate between practices during which they do not want to be interrupted and practices in which it is acceptable to be interrupted. Fig. 4a provides an overview of the classifications of the practices of the requirements development area. This table shows that the participants indicated that they do not want to be interrupted while performing activities corresponding to elicit stakeholder needs and validating requirements (SP 1.1 and SP 3.5). Next, the participants did not reach consensus on practice SP 3.3, analyzing requirements, while they indicated that they do not mind to be interrupted during all other practices.

During the discussion the participants unanimous indicated that they do not want to be interrupted while they are eliciting the needs of the customer (SP 1.1). They indicated that the activities corresponding to this practice are mainly of a highly interactive nature.

| **Practice** | Inf | Int |
|---|---|---|
| *SG 1. Develop Customer Requirements* | | |
| SP 1.1 Elicit Needs | ✗ | ✗ |
| SP 1.2 Transform Stakeholder Needs into Customer Requirements | ✗ | ✓ |
| *SG 2. Develop Product Requirements* | | |
| SP 2.1 Establish Product and Product Component Requirements | ✓ | ✓ |
| SP 2.2 Allocate Product Component Requirements | ✗ | ✓ |
| SP 2.3 Identify Interface Requirements | ? | ✓ |
| *SG 3. Analyze and Validate Requirements* | | |
| SP 3.1 Establish Operational Concepts and Scenarios | ✗ | ✓ |
| SP 3.2 Establish a Definition of Required Functionality and Quality Attributes | ✗ | ✓ |
| SP 3.3 Analyze Requirements | ✗ | ? |
| SP 3.4 Analyze Requirements to Achieve Balance | ? | ✓ |
| SP 3.5 Validate Requirements | ✗ | ✗ |

(a) Table containing both [Inf] information software engineers want to know immediately, and [Int] activities during which software engineers do not mind to be Interrupted




(b) Visualization of the results

Figure 4. Requirements Development

## B. Technical Solution

The second process area we discuss is the *'Technical Solution'* area. The purpose of this area is to select, design and implement solutions to the identified requirements [12]. This area consists of the following three goals:

SG 1 *Select product component solutions*
Product or product component solutions are selected from alternative solutions

SG 2 *Develop the design*
Product or product component designs are developed

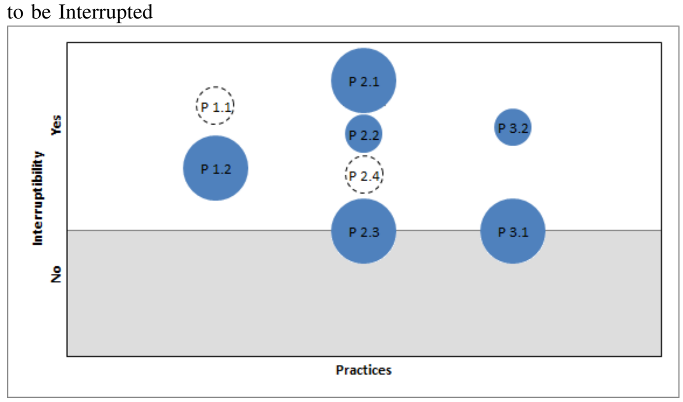
SG 3 *Implement the product design*
Product components are implemented from their designs

*1) Information updates:* The classifications of the practices necessary to achieve the goals of the technical solution area are shown in Fig. 5a. In this table it can been seen that the participants are interested in new information of four of the eight practices, namely information about selecting product component solutions (SP 1.2), designing the product (SP 2.1), designing the interfaces (SP 2.3) and implementing the design (SP 3.1).

During the meeting practices SP 1.1 and SP 2.4 were most discussed. Both practices consist of activities in which multiple analyses are conducted. On the one hand several possible solutions are examined (SP 1.1) while on the other hand many make, buy and reuse analyses are performed (SP 2.4). Some of the participants argued that they *"definitely want to know such information immediately, so that they are able to influence the outcomes"* while others argued that they *"do not necessarily want to be informed of this information immediately, but are*

| **Practice** | Inf | Int |
|---|---|---|
| *SG 1. Select Product Component Solutions* | | |
| SP 1.1 Develop Alternative Solutions and Selection Criteria | ? | ✓ |
| SP 1.2 Select Product Component Solutions | ✓ | ✓ |
| *SG 2. Develop the Design* | | |
| SP 2.1 Design the Product or Product Component | ✓ | ✓ |
| SP 2.2 Establish a Technical Data Package | ✗ | ✓ |
| SP 2.3 Design Interfaces Using Criteria | ✓ | ? |
| SP 2.4 Perform Make, Buy, or Reuse Analyses | ? | ✓ |
| *SG 3. Implement the Product Design* | | |
| SP 3.1 Implement the Design | ✓ | ? |
| SP 3.2 Develop Product Support Documentation | ✗ | ✓ |

(a) Table containing both [Inf] information software engineers want to know immediately, and [Int] activities during which software engineers do not mind to be Interrupted




(b) Visualization of the results

Figure 5. Technical Solution

*particularly interested in the choices being made"*.

*2) Interruptibility:* Also for this area, we asked the participants to indicate during which activities they do not mind to be interrupted. It is interesting to note that for none of the activities needed to satisfy this area, the participants indicated that they prefer not be interrupted.

Furthermore it is noteworthy to mention one of the arguments made during the discussion regarding whether or not you prefer to be interrupted when you are implementing the design (SP 3.1). During this discussion some participants indicated that they were *"in the zone"* when they are implementing specific functionality. They emphasize that interruptions in such a mental state have a huge impact, since it is extremely difficult to reach such a mental state again.

### C. Verification

The third engineering process area which we discuss is the *'Verification'* area. The purpose of this area is to ensure that work products meet their specified requirements [12]. This area consists of three goals:

SG 1 *Prepare for verification*
Preparation for verification to ensure that verification provisions are embedded in product requirements, designs, implementation and schedules

SG 2 *Perform peer reviews*
Peer reviews involve a methodical examination of work products to identify defects and recommend changes

SG 3 *Verify selected work products*
Verification methods, procedures and criteria are used to actually verify selected work products

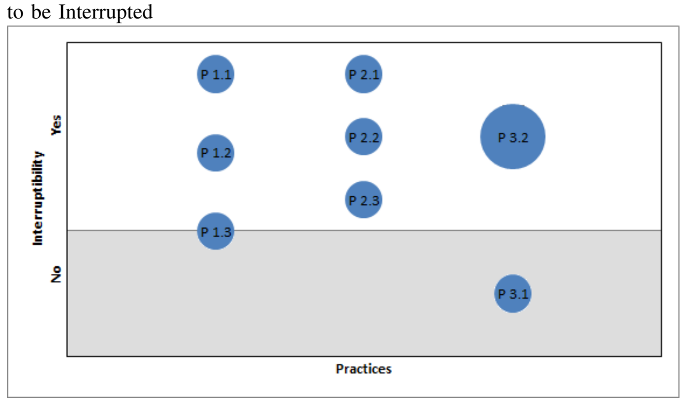
*1) Information updates:* The participants of this study only indicated that they want to be immediately informed of information regarding the verification results (SP 3.2), as can been seen in Fig. 6a.

There was, however, a discussion about the information regarding the analysis of peer review data (SP 2.3). One of the participants stated that such information is useful to him, especially when it concerns his own work, and that he immediately wants to be informed of such information. Another participant agreed, but said: *"I do not want to be informed of the results of the peer reviews of everyone else"*. So, overall they indicated that they are not interested in such information since they are only interested in a specific part of the information.

*2) Interruptibility:* The classifications of whether or not they prefer to be interrupted while performing activities related to the verification area are also depicted in Fig. 6a. The participants reached consensus on practice SP 3.1, performing the verification, for which they agreed they do not want to be interrupted. The discussion of practice SP 1.3, establish verification procedures and criteria, resulted in an interesting finding, namely that the participants use different processes to establish these procedures and criteria. Some of the participants establish these in close collaboration with the customer and therefore they would not like to be interrupted. Other participants establish these procedures and criteria on their own and therefore do not mind to be interrupted.

| **Practice** | Inf | Int |
|---|---|---|
| *SG 1. Prepare for Verification* | | |
| SP 1.1 Select Work Products for Verification | ✗ | ✓ |
| SP 1.2 Establish the Verification Environment | ✗ | ✓ |
| SP 1.3 Establish Verification Procedures and Criteria | ✗ | ? |
| *SG 2. Perform Peer Reviews* | | |
| SP 2.1 Prepare for Peer Reviews | ✗ | ✓ |
| SP 2.2 Conduct Peer Reviews | ✗ | ✓ |
| SP 2.3 Analyze Peer Review Data | ✗ | ✓ |
| *SG 3. Verify Selected Work Products* | | |
| SP 3.1 Perform Verification | ✗ | ✗ |
| SP 3.2 Analyze Verification Results | ✓ | ✓ |

(a) Table containing both [Inf] information software engineers want to know immediately, and [Int] activities during which software engineers do not mind to be Interrupted




(b) Visualization of the results

Figure 6. Verification

### D. Validation

The fourth area we discuss is the *'Validation'* area. The purpose of this engineering area is to demonstrate that a product fulfills its intended use [12]. This area consists of two goals:

SG 1 *Prepare for validation*
Preparation for validation include selecting products for validation and establishing and maintaining the validation environment

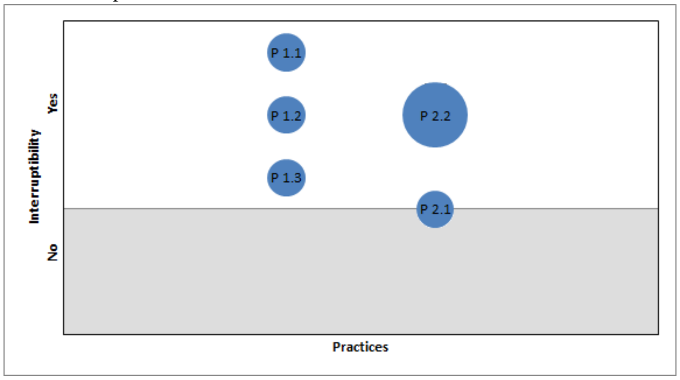
SG 2 *Validate product or product components*
Validation methods, procedures and criteria are used to actually validate selected work products

*1) Information updates:* The classifications of the validation practices are very similar to the classification of the verification practices, see Fig. 7a. Again, the participants indicated that they do not want to be informed immediately of information regarding the preparation of the validation process (SP 1.1, SP 1.2 and SP 1.3). While they do want be notified of information about the analysis of the results (SP 2.2).

*2) Interruptibility:* It is interesting to see that for none of the practices of the validation area the participants indicated that they prefer not be interrupted. The only practice on which they did not reach consensus is practice SP 2.1, performing the actual validation. Some of the participants argued that they prefer not to be interrupted by colleagues since this process is performed in close collaboration with the customer. Other participants, however, argued that they do not mind to be interrupted. Since, performing the validation can be a time-consuming activity, which can take days, it is unfeasible not to be disrupted at all.

| Practice | Inf | Int |
|---|---|---|
| *SG 1. Prepare for Validation* | | |
| SP 1.1 Select Products for Validation | ✗ | ✓ |
| SP 1.2 Establish the Validation Environment | ✗ | ✓ |
| SP 1.3 Establish Validation Procedures and Criteria | ✗ | ✓ |
| *SG 2. Validate Product or Product Components* | | |
| SP 2.1 Perform Validation | ✗ | ? |
| SP 2.2 Analyze Validation Results | ✓ | ✓ |

(a) Table containing both [Inf] information software engineers want to know immediately, and [Int] activities during which software engineers do not mind to be Interrupted




(b) Visualization of the results

Figure 7. Validation

### E. Product Integration

Finally, we discuss the *'Product Integration'* area. The purpose of this area is to assemble the product from the product components, ensure that the product fulfills all requirements, and deliver the product [12]. This area consists of the following three goals:

SG 1 *Prepare for product integration*
Preparation for product integration includes establishing an integration strategy, the integration environment and the integration procedures and criteria.

SG 2 *Ensure product interface compatibility*
Effective management of product component interfaces helps ensure that implemented interfaces will be complete and compatible

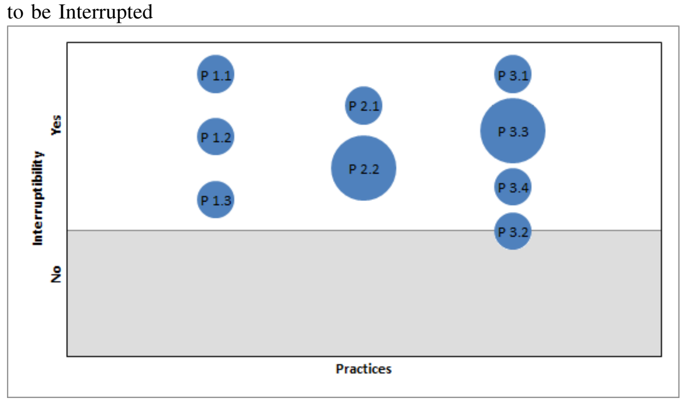
SG 3 *Assemble product components and deliver the product*
Integration of product components proceeds according to the product integration strategy and procedures.

*1) Information updates:* In Fig. 8a the results of the discussions are shown. In this table it can been seen that the experts are only interested in direct updates of information regarding managing interfaces (SP 2.2) and evaluation of assembled product components (SP 3.3). Again they are less interested in information about the preparation phase.

*2) Interruptibility:* Finally, we discuss the classifications regarding interruptability. The participants reached consensus for all but one practice. The only practice on which they did not reach consensus is practice SP 3.2. The two main arguments in the discussion on this practice regarding assembling product components are: (i) *"assembling product components is really important and requires a high level of concentration, so I do not want to be interrupted"* and (ii) *"I do not mind to be interrupted, since assembling product components does not require specialist knowledge"*. Overall the participants concluded that whether or not you prefer to be interrupted during the assembly of product components strongly depends on personal preference.

| Practice | Inf | Int |
|---|---|---|
| *SG 1. Prepare for Product Integration* | | |
| SP 1.1 Establish an Integration Strategy | ✗ | ✓ |
| SP 1.2 Establish the Product Integration Environment | ✗ | ✓ |
| SP 1.3 Establish Product Integration Procedures and Criteria | ✗ | ✓ |
| *SG 2. Ensure Interface Compatibility* | | |
| SP 2.1 Review Interface Descriptions for Completeness | ✗ | ✓ |
| SP 2.2 Manage Interfaces | ✓ | ✓ |
| *SG 3. Assemble Product Components and Deliver the Product* | | |
| SP 3.1 Confirm Readiness of Product Components for Integration | ✗ | ✓ |
| SP 3.2 Assemble Product Components | ✗ | ? |
| SP 3.3 Evaluate Assembled Product Components | ✓ | ✓ |
| SP 3.4 Package and Deliver the Product or Product Component | ✗ | ✓ |

(a) Table containing both [Inf] information software engineers want to know immediately, and [Int] activities during which software engineers do not mind to be Interrupted




(b) Visualization of the results

Figure 8. Product Integration

## V. FINDINGS POST-ROUND

In this section we present the findings of the post questionnaire [5]. The findings related to the first two research questions are contradicting since, on the one hand, for some practices the experts indicated that they prefer not to be interrupted when performing activities corresponding to that practice, while on the other hand the experts indicated that they want to be informed of information regarding several practices immediately. In order to elaborate on the needs of software engineers, we asked each of the participants to indicate both (i) what information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted, and (ii) whether or not they prefer to be interrupted with information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted.

The participants reached consensus on five of the nine practices regarding the information they want to know immediately. For each of these practices, RD 2.1, TS 3.1, VER 3.2, PI 2.2 and PI 3.3, they agreed they do not want to be informed immediately when they are performing an activity during which they prefer not to be interrupted. Furthermore, it is interesting to note the experts have not yet reached consensus

[5] See http://aspic.nl/vow/post-questionnaire-classifications.pdf

TABLE III
INFORMATION SOFTWARE ENGINEERS WANT TO KNOW IMMEDIATELY EVEN THOUGH THEY PREFER NOT TO BE INTERRUPTED

| | | RD | TS | | | | VER | VAL | PI | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 2.1 | 1.2 | 2.1 | 2.3 | 3.1 | 3.2 | 2.2 | 2.2 | 3.3 |
| RD | 1.1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | 3.5 | ✗ | ? | ? | ? | ✗ | ✗ | ? | ✗ | ✗ |
| VER | 3.1 | ✗ | ? | ? | ? | ✗ | ✗ | ? | ✗ | ✗ |

on a practice of which they do want to be immediately informed.

We also asked the experts to indicate whether or not they prefer to be interrupted with information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted. The only practice on which the experts reached consensus is practice SP 1.1 in the area of requirements development. They agreed they prefer not be interrupted while performing activities related to this practice even if it concerns information of which they want to be informed immediately.

These results are combined in table III. In this table a *'✗-sign'* indicates software engineers do not want to be immediately informed of information regarding the practice depicted in the column, while they are performing activities corresponding to the practice depicted in the row. A *'?- sign'* indicates the participants have not yet reached consensus. Currently, the participants have not reached consensus on a practice of which they want to be immediately informed.

## VI. DISCUSSION

In this section we reflect on the findings and discuss the most important results of this study.

Firstly, we discuss what information software engineers want to know immediately, see table IV. From the figures in the previous section it can be seen that participants of our study only want to be informed immediately of roughly one quarter of the practices. It is interesting to notice that the experts want to be informed immediately of at least one practice of each of the five process areas. For each of these areas, they are mainly interested in direct updates of information regarding completed artifacts (e.g. requirements, design and verification results), and are less interested in direct updates of information concerning the procedures used and environment needed to construct these artifacts. Another interesting result is the distribution of the practices, of which the participants want to be informed immediately, over the process areas. Nearly half of these practices belong to the technical solution area.

TABLE IV
INFORMATION SOFTWARE ENGINEERS WANT TO KNOW IMMEDIATELY

| Area | Practice |
|---|---|
| Requirement Development | SP 2.1 Establish Product and Product Component Requirements |
| Technical Solution | SP 1.2 Select Product Component Solutions |
| Technical Solution | SP 2.1 Design the Product or Product Component |
| Technical Solution | SP 2.3 Design Interfaces Using Criteria |
| Technical Solution | SP 3.1 Implement the Design |
| Verification | SP 3.2 Analyze Verification Results |
| Validation | SP 2.2 Analyze Validation Results |
| Product Integration | SP 2.2 Manage Interfaces |
| Product Integration | SP 3.3 Evaluate Assembled Product Components |

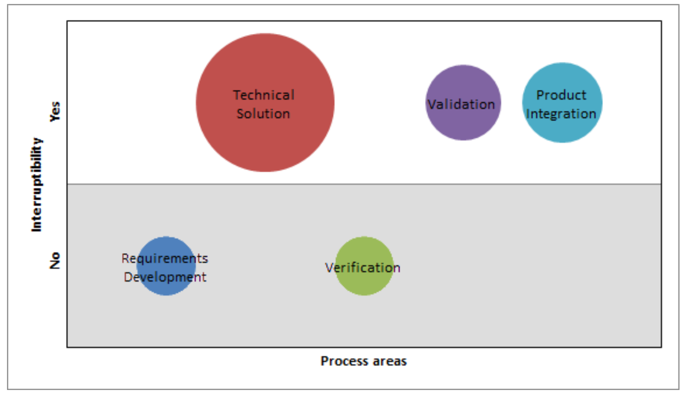



Figure 9. Summary of the process areas

Participants are not only interested in the implementation itself but are also interested in the selected solutions, the design of the component and the design of the interfaces. This is shown in Fig. 9, since we have plotted the ratio of all process areas, defined as the number of practices of which the participants want to be immediately informed, and the number of practices on which the participants reached consensus.

Secondly, we elaborate on research question 2: *"During which activities do software engineers prefer to be interrupted to provide them with information?"*. The study participants indicated in general they do not mind to be interrupted while performing software engineering related activities. They only reached consensus on three practices during which they prefer not to be interrupted, see table V. These practices belong to either the requirements development area or the verification area (see Fig. 9). The main arguments used to convince others that they prefer not to be interrupted while performing a certain activity are (i) the activity is of a high interactive nature involving customers and therefore you do not want to be interrupted and (ii) performing the activity requires a high level of concentration and therefore it is difficult to again reach this mental state after an interruption. Overall the participants agreed that whether or not you prefer to be interrupted during a specific activity strongly depends on the person.

Thirdly, research question 3 elaborates on the needs of software engineers when they are performing activities during which they prefer not to be interrupted in general. We asked the participants to indicate both (i) what information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted, and (ii) whether or not they prefer to be interrupted with information they want to know immediately, even though they are performing an activity during which they prefer not to be interrupted. The initial results of these questions indicate that software engineers do not want to be informed immediately when they are performing an activity during which they do not want to be interrupted.

TABLE V
ACTIVITIES DURING WHICH ENGINEERS PREFER NOT TO BE INTERRUPTED

| Area | Practice |
|---|---|
| Requirement Development | SP 1.1 Elicit Needs |
| Requirement Development | SP 3.5 Validate Requirements |
| Verification | SP 3.1 Perform Verification |

Finally, we reflect on the Estimate-Talk-Estimate study we conducted, because such a study is rarely conducted in the field of software engineering. In such a study it is important that all participants have a thorough understanding of the issues being discussed to make reliable decisions. In the first round we conducted a questionnaire in which we asked the participants to classify the practices based on a brief description. In the second round we asked the participants to classify the practices on which no consensus was reached during the first round based on a detailed description. In round three and four a interactive discussion took place among the experts, after which the practices on which no consensus was reached were classified. In the post round we asked for their understanding of the practices of the five process areas. Participants indicated to have a good or very good understanding of the practices out of the following options: *'Very-Good'*, *'Good'*, *'Acceptable'*, *'Poor'* and *'Very-Poor'*. As such, all participants were able to make well informed decisions. Another important element of this study are the interactive group discussions. In this discussions participants were asked to provide reasons for their decisions and to respond to the decisions made by others, after which participants could revise their opinion. To see if this process actually took place we asked the participants if they revised their opinion based on the arguments of others. They were given five options: *'Never'*, *'Rarely'*, *'Sometimes'*, *'Very Often'* and *'Always'*. Nine of the participants indicated this was sometimes the case. Only one of the participants indicated he often changed his opinion based on the arguments of others. Based on these experiences we consider an Estimate-Talk-Estimate study is a suitable way to allow participants to systematically classify multiple issues.

## VII. Threats to validity

In this section we discuss the threats to validity for this study on four aspects: reliability, construct validity, internal validity and external validity. We mitigated threats to reliability by providing a detailed description of the methods we used. We have described in detail how the study participants were selected and of which rounds this study consists. We also made the design of both questionnaires and the detailed descriptions of the practices available. Next to this we also make all classifications of the practices available in a anonymized form. We do this to make both our data gathering methods and the analysis of our data repeatable, and as such increase the reliability of this research. These thorough descriptions also mitigate some of the threads of construct validity, since a careful study design assures a higher construct validity. By pretesting both questionnaires with two software engineers, who did not participate in this study, we increased the reliability and construct validity of this research even further.

Next, there exist threats to internal validity. During this study several practices were classified during a collaborative session. These practices are relatively abstract because they could reference multiple information items and activities. To determine to which extent this influenced the results we asked the participants about their understanding of the practices. They all indicated they had a good understanding of the practices of the five process areas, one participant even stated to have a very good understanding. Since applied social research is a human activity, it is also possible social pressure influenced the outcomes of this research. Since the estimates made by the participants were not anonymous, as in a Delphi study, it is possible participants changed their behavior to fulfill the expectations of others as a result of real or imagined group pressure. Therefore, in the post questionnaire, we asked the participants if they changed their opinion because of the peer pressure of the group of experts. They were given five options: *'Never'*, *'Rarely'*, *'Sometimes'*, *'Very Often'* and *'Always'*. Seven of the participants indicated they never changed their opinion while the other three indicated this was rarely the case.

Finally, we discuss threats to external validity. External validity is of interest in studies that draw generalized conclusions. In this study we consulted ten experienced Dutch software engineers from nine different companies. The fact that we only consulted participants who had the Dutch nationality is a threat to external validity. To improve the external validity of the findings, the study should be repeated with an international expert panel.

## VIII. Conclusions and Future Work

In this paper we have reported on the empirical study we conducted on how to regulate information available to software engineers based on both the importance of the information and the current activity of the engineer. To structure this research we used the five engineering process areas defined in the CMMI [12]: Requirements Development, Technical Solution, Verification, Validation and Product Integration. These areas consist of multiple goals and practices, which we used to determine (i) of what information software engineers want to be immediately informed, and (ii) during which activities software engineers do not mind to be interrupted. The outcomes of the two research directions were contradicting in some cases, since participants of this study indicated that they would like to be informed of several practices immediately while for other practices they indicated that they prefer not to be interrupted. Therefore we also researched (iii) of what information software engineers want to be immediately informed, even though they are performing an activity during which they prefer not to be interrupted.

The main contributions of this paper are the answers to the research questions. First, we showed a list of information items of which software engineers want to be informed immediately. Subsequently, we presented a list of software engineering activities during which software engineers prefer not to be interrupted. Finally, we discussed a look-up table which can be used to determine whether or not software engineers want to be immediately informed, even though they are performing an activity during which they prefer not to be interrupted.

Next, when abstracting the findings, we can conclude that:

- Software engineers want to be immediately informed of a wide variety of information
- Software engineers are mainly interested in direct updates of information about completed artifacts
- Software engineers are especially interested in information regarding the technological solution itself
- In most cases software engineers do not mind to be interrupted to provide them with information

- Software engineers prefer not to be interrupted when they are performing activities of a highly interactive nature
- Software engineers prefer not to be interrupted when they are performing activities which require a high level of concentration
- Software engineers do not want to be immediately informed of any information when they are performing an activity during which they prefer not to be interrupted

Next steps of this research include (i) reaching consensus on information items on which no consensus was reached, (ii) reaching consensus on information items of which software engineers want to be immediately informed when performing activities during which they prefer not to be interrupted, and (iii) research the different practices in more detail, e.g. what artifacts and what specific information is needed. These results can then be used to construct virtual office walls which automatically regulate information available to a software engineer, based on both the the current activity of the engineer, and the information engineers want to know immediately. Until we all work in a *'Virtual Office'* in which information is regulated by *'auto-erecting virtual office walls'*, we should carefully consider whether to interrupt a colleague to provide him or her with information. So, when in doubt, do not disturb!


## ACKNOWLEDGMENT

We would like to thank Bas Baas, Bart Gloudemans, Peter Hoogerbrugge, Joep Joosten, Dennis Laumen, Robin van Loon, Arno van Nijnaten, Silvain Piree, Raymond Pols, Wim Pool, Corno Schraverus and Angelo Wentzler who have participated in this Estimate-Talk-Estimate study for their time and help.


## REFERENCES


[1] 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.

[2] 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.

[3] B. van Gameren, K. Dullemond, and R. van Solingen, "Auto-erecting virtual office walls," in *Proceedings of CollaborateCom*, 2012.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaíno, "Collaboration tools for global software engineering," *Software, IEEE*, vol. 27, no. 2, pp. 52–55, 2010.

[10] P. Tell and M. A. Babar, "Activity theory applied to global software engineering: Theoretical foundations and implications for tool builders," in *Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on*. IEEE, 2012, pp. 21–30.

[11] 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.

[12] C. P. Team, "Cmmi for development, version 1.3, improving processes for developing better products and services," *no. CMU/SEI-2010-TR-033. Software Engineering Institute*, 2010.

[13] D. H. Gustafson, R. K. Shukla, A. Delbecq, and G. W. Walster, "A comparative study of differences in subjective likelihood estimates made by individuals, interacting groups, delphi groups, and nominal groups," *Organizational Behavior and Human Performance*, vol. 9, no. 2, pp. 280–291, 1973.

[14] V. H. Vroom, L. D. Grant, and T. S. Cotton, "The consequences of social interaction in group problem solving," *Organizational Behavior and Human Performance*, vol. 4, no. 1, pp. 77–95, 1969.

[15] M. D. Dunnette, *Are meetings any good for solving problems?* Selbstverl., 1964.

[16] K. C. Green, J. S. Armstrong, and A. Graefe, "Methods to elicit forecasts from groups: Delphi and prediction markets compared," *Foresight: The International Journal of Applied Forecasting*, vol. 8, no. 8, pp. 17–21, 2007.

[17] C. Okoli and S. D. Pawlowski, "The delphi method as a research tool: an example, design considerations and applications," *Information & Management*, vol. 42, no. 1, pp. 15–29, 2004.

[18] E. R. Babbie, *The Practice of Social Research*, 13th ed. Cengage Learning, 2012.

[19] N. Dalkey and O. Helmer, "An experimental application of the delphi method to the use of experts," *Management science*, vol. 9, no. 3, pp. 458–467, 1963.

[20] J. Grenning, "Planning poker or how to avoid analysis paralysis while release planning," *Hawthorn Woods: Renaissance Software Consulting*, 2002.

[21] A. Gibbs, "Focus groups," *Social research update*, vol. 19, no. 8, 1997.