Project Management Metrics in the Age of Global Software Engineering

Christian Lescher[a], Helmut Naughton[a], Bernd Bruegge[a], Christopher Schulz[b], Christian Neubert[b], and Florian Matthes[b]

[a] Technische Universität München
Chair for Applied Software Engineering
D-85748 Garching, Germany

[b] Technische Universität München
Chair for Software Engineering for Business Information System
D-85748 Garching, Germany

{lescher,naughton,bruegge,schulzc,neubert,matthes}@in.tum.de

Abstract

Globalization has long since found its way into software engineering. Companies transfer part of their development activities to low wage countries in order to achieve a better cost position, get access to local markets and react to the prevailing lack of specialized workforce. However, Global Software Engineering (GSE) has also brought new challenges: Geographic separation, different time zones, and culture and language barriers make the collaboration of team members more difficult and often lead to quality problems, project delays, and cost overruns. Project management metrics to monitor cost, time, and quality characteristics of a development project, are state of the practice today. However, these metrics often reveal problems in GSE projects too late, since they measure the symptoms – exceeded cost, missed milestones or poor quality – rather than the root causes of the problems.

In this paper, we report on a study of new project management metrics based on measurement of communication and collaboration, which provide project managers with an early warning system. We introduce a collaboration-based measurement model – a set of "GSE metrics" – and describe case studies of its application in multiple projects. In the observed projects, the GSE metrics were helpful to identify and resolve collaboration problems. Despite the subjective nature of the metrics, the results showed the GSE metrics to be useful and to accurately reflect the reality. The effort for the application of the GSE metrics turned out to be a "low investment" with a high "return on invest".

Keywords

global software engineering, project management, metrics, collaboration, communication

Kurzfassung

Die Globalisierung hat längst auch Einzug in die Softwareentwicklung gehalten. Firmen verlagern Teile ihrer Entwicklungsaktivitäten in Niedriglohnländer, um eine bessere Kostenposition zu erreichen, den Zugang zu lokalen Märkten zu schaffen und auf den vorherrschenden Fachkräftemangel zu reagieren. Doch globale Softwareentwicklung (GSE) bringt auch neue Herausforderungen mit sich: Geografische Verteilung, das Arbeiten in verschiedenen Zeitzonen sowie verschiedene Kulturen und Sprachen erschweren die Zusammenarbeit der Teammitglieder und führen häufig zu Qualitätsproblemen, Terminverzug und Kostenüberschreitungen. Projektmanagement-Metriken zur Überwachung von Kosten-, Zeit- und Qualitätsaspekten eines Entwicklungsprojekts sind heute zwar in der Praxis etabliert. Doch anhand dieser Metriken sind Probleme in GSE-Projekten oft erst spät zu erkennen, denn sie messen eher die Symptome – überschrittene Kosten, verpasste Meilensteintermine oder schlechte Qualität – statt der Problemursachen.

Dieser Beitrag berichtet von einer Studie über neue Projektmanagement-Metriken, die auf der Messung von Kommunikation und Zusammenarbeit beruhen und Projektmanagern ein Frühwarnsystem an die Hand geben. Wir führen ein kollaborationsbasiertes Messmodel ein – ein Satz von "GSE-Metriken" – und beschreiben Fallstudien seiner Anwendung in mehreren Projekten. In den untersuchten Projekten waren die GSE-Metriken hilfreich, um Probleme in der Zusammenarbeit zu identifizieren und zu lösen. Trotz der subjektiven Natur der Metriken zeigten die Ergebnisse, dass die GSE-Metriken nützlich sind und die Realität richtig wiedergeben. Der Aufwand für die Anwendung der GSE-Metriken erwies sich als niedriges Investment mit einem hohen "Return on Invest".

Schlüsselwörter

Globale Softwareentwicklung, Projektmanagement, Metriken, Zusammenarbeit, Kommunikation

1 Introduction

The globalization of software engineering activities has become a mainstream trend in industry [30]. Many companies conduct software development in teams which are globally distributed, because they expect to benefit from the proximity to markets or customers, access to talent regardless of location, and a better cost position. However, Global Software Engineering (GSE) has also brought new challenges to development teams: Geographic separation, working in different time zones, and different cultures and languages make the collaboration[1]

[1] We use the term *collaboration* in the sense of "to work jointly with others or together especially in an intellectual endeavor" [27].

of team members more difficult than in co-located projects and often lead to quality problems, project delays, and cost overruns [15][5][17]. Typical collaboration problems include ineffective communication, unclear responsibilities, lack of informal communication, and impeded team cohesion.

Metrics for project management support have been in place for decades now. They are used by project managers to observe the status of their project, including cost (e.g., actual and planned cost), time (e.g., milestone trend analysis), and quality aspects (e.g., defect rates). Metrics can support project managers in their decisions and help them to perform corrective actions to keep projects on track [11][24]. However, project management metrics addressing the specific collaboration and communication challenges of GSE are still in their infancies. In a systematic literature review [20, p.24] of all publications of the IEEE International Conference on Global Software Engineering (ICGSE) it was found that while almost all papers refer to collaboration resp. communication challenges, only few of them (7%) actually deal with metrics, but for examination of completed projects (post mortem analysis). No evidence was found for project management metrics tailored to the needs of GSE.

In this paper, we report on our research [20] of new project management metrics for GSE based on the measurement of communication and collaboration. We introduce a collaboration-based measurement model – a set of "GSE metrics" – and describe case studies of its application in multiple projects.

2 Research Process

In order to introduce a collaboration-based measurement model for GSE, we take a three step approach, which is expressed through the following research questions:

- **RQ1 (Understanding the problem domain):** What are typical collaboration challenges in GSE projects?
- **RQ2 (Construction of the measurement model):** Which project management metrics can be defined to monitor collaboration in GSE?
- **RQ3 (Validation of the measurement model):** What is the accuracy, usefulness, effort for data collection, and innovation grade of these metrics?

We follow the principles of *formative* evaluation, which is concerned with forming a program or technology (in our case a measurement model), "by examining the delivery of the program or technology, the quality of its implementation, and the assessment of the organizational context, personnel, procedures, inputs, and so on" [31]. The primary goal of the application is to develop and improve the program or technology. The application of the measurement model to a large set of projects in the sense of a *summative* evaluation is not in the scope of this study.

With our research, we open up the new field of collaboration-based project management metrics for GSE. These new metrics make collaboration aspects in software engineering teams measurable and in this way provide a foundation for empirical and statistical analysis of software engineering projects.

3 Related Work

The distinctive characteristic of GSE is *distance*: Team members are located in different countries (*geographic distance*), they work in different time zones (*temporal distance*), and they are from different cultural backgrounds (*cultural distance*). Consequently, collaboration and communication become more complicated than in co-located projects where all team members are located in immediate vicinity [15]. A key effect in GSE is delay. Herbsleb and Mockus found in their studies that distributed work items take about 2.5 times longer to complete than similar items where all the work is co-located [16]. We conducted a controlled experiment on a prioritization task in a distributed team, which revealed a similar result (average delay of 2.4 on the same task in a distributed compared to a co-located setting) [20, pp.31–36].

3.1 The Challenges of Global Software Engineering

In order to obtain a profound understanding of characteristic challenges in GSE, we conducted a comprehensive literature research, case studies as well as controlled experiments. We identified multiple models dealing with different aspects of GSE which we systematized into a meta model (see Figure 1). The meta model is organized according to a taxonomy introduced by Rumbaugh and colleagues [29, p.17]. In recent publications, the taxonomy is also called Structure–Behavior–Function framework [13, 14]. The meta model relates GSE models to three categories (the columns in Figure 1):

- **Structure (Object Model).** This category deals with the structure of objects, i.e., components of the objects and connections among them [29,

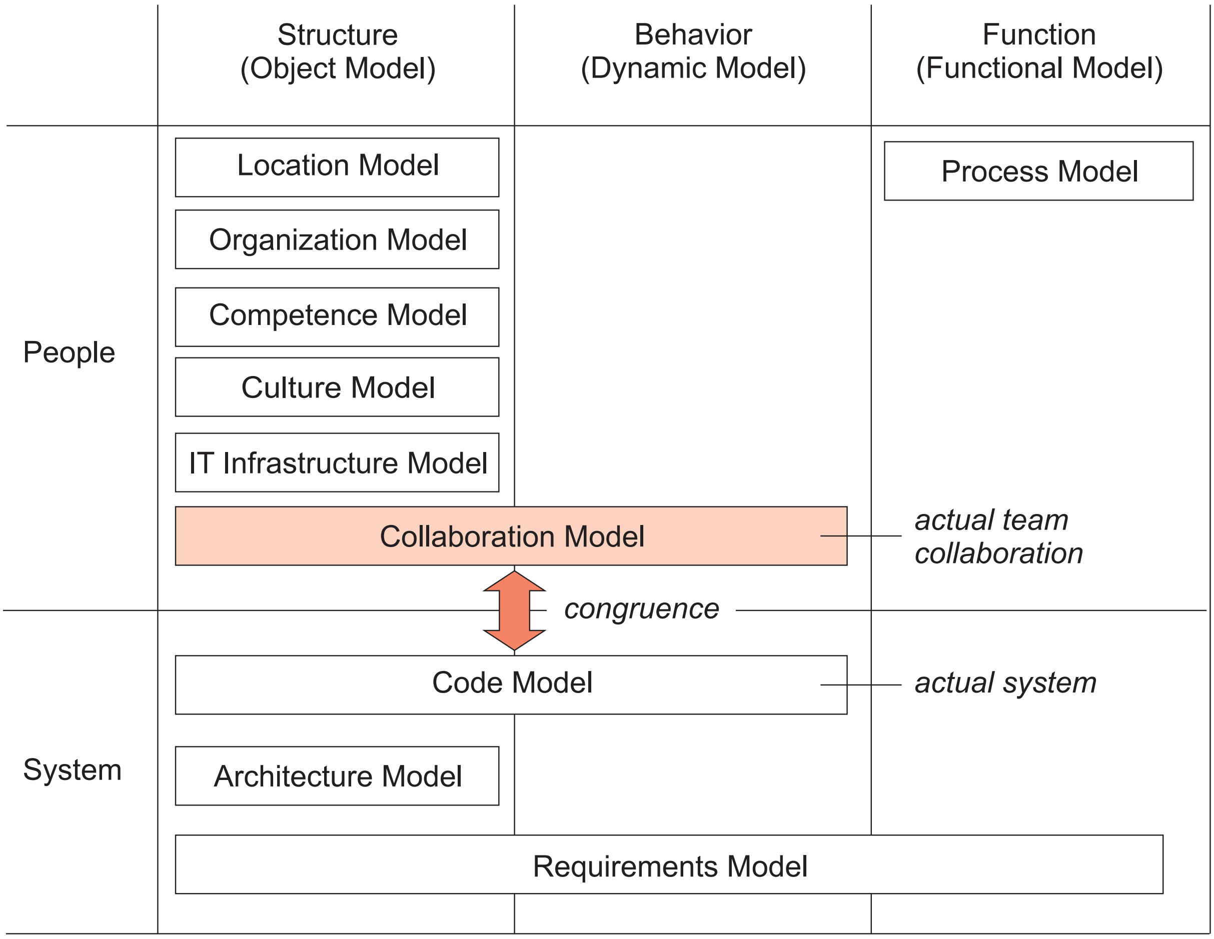


Figure 1: Models of Global Software Engineering

p.17][13, 14]. The object in the GSE meta model are either the people involved in software engineering or the system under development.

- **Behavior (Dynamic Model).** This type of models "describes those aspects of a system concerned with time and the sequencing of operations" [29, p.18], it refers to dynamic behavior, e.g., a sequence of states and transitions between them [14].
- **Function (Functional Model).** Functional models describe the teleology of the objects (i.e., what it is for) [13] resp. functions characterized by preconditions and postconditions [14].

The GSE meta model distinguishes between people-related models and system-related models (the rows in Figure 1). People-related models deal with the team members of a development team. The people-related models contained in the GSE meta model are: Location Model, Organization Model, Competence Model, Culture Model, IT Infrastructure Model, Collaboration Model and Process Model. System-related models are concerned with the technical system under development. The system-related models in the GSE meta model are: Code Model, Architecture Model and Requirements Model.

Table 1 provides an overview of the identified models. We use the term *challenge* to refer to a characteristic which makes GSE difficult. A challenge either decreases the ability to collaborate effectively (e.g., degree of distribution) or increases the need to collaborate (e.g., uncertain component interfaces). Please note that we give examples of typical challenges for GSE, but we do not strive for completeness here.

The *Collaboration Model* represents communication structures and behavior of the team members in a GSE project. This model is the focus of our GSE metrics approach, because it represents the actual team behavior, while Location, Organization, Competence, Culture, IT Infrastructure, and Process Model are concerned with the frame conditions for the development teams. The Collaboration Model involves the following characteristic challenges:

Impeded communication and awareness. Distribution impedes communication and awareness. Typical challenges are difficulties of knowing who to contact about what, of initiating contact, and of communicating effectively across sites [15].

Loss of communication richness. According to Watzlawick [32], a communication message comprises both a factual information (content level) and information about the relationship of the communicants (relationship level). The latter is often expressed through non-verbal communication such as facial expressions or gestures. GSE teams use different kinds of communication media, e.g., video conferencing, telephone, or e-mail. They differ with respect to the level of communication richness, where *rich communication* is defined as "two-way interaction involving more than one sensory channel" [8, p.48]. GSE causes a loss of communication richness and therefore impacts the ability of the team to collaborate effectively.

Lack of team cohesion. Successful teams are cohesive: Team cohesion – "the act or state of sticking together tightly" [26] – leads to enhanced motivation, increased morale, greater productivity, harder work, more open communication, and higher job satisfaction compared to non-cohesive groups [8, p.52]. Team cohesion is more difficult to achieve for a globally distributed team; distance is

Model	Scope	Challenges
Location	Geographic distribution of the project team	Geographic distance, degree of distribution (number of sites), time zone differences
Organization	Organizational units, e.g., companies or company departments	Coordination of suppliers, unclear escalation path, divergent incentive structures
Competence	Knowledge, skills and process abilities of the workforce	Domain expertise, technology expertise, language skills, experience on GSE projects
Culture	National culture, organizational culture	Cultural differences leading to misunderstandings and ineffective collaboration
IT Infrastructure	Communication and development tools	Divergent development environments, unreliable or inadequate IT/telecom infrastructure
Process	Development process	Divergent or unsuitable development processes
Collaboration	Communication structures and behavior of team members in a GSE project	Impeded communication and awareness, loss of communication richness, lack of team cohesion
Code	The actual system under development	Integration difficulties, code quality problems
Architecture	Top level design of the system	Architectural change, uncertain component interfaces, uncertain allocation of functionality
Requirements	Desired functionality, non-functional properties and constraints of the system to be developed	Uncertain requirements, stringent non-functional requirements, central components or cross-cutting features

Table 1: Models of GSE and characteristic challenges

an impediment to building relationships of trust [8, p.53].
According to Conway's Law, "A design effort should be organized according to the need for communication", because "organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations" [9], which implies a congruence between the people and the system level of the GSE meta model, in particular between the Collaboration Model and the Code Model.

While geographic, temporal and cultural distance in general impede collaboration, they are no adequate indicator of collaboration difficulties. Another measurement is needed referring to the *collaboration distance* between team members.

3.2 Software Measurement and Communication Metrics

Introducing objective *metrics* is a well established management technique to monitor the current status of a project. We described basic measurement concepts in an earlier publication [21]. Software measurement helps to understand more about software development as a basis for making operational decisions as well as estimation and prediction. It is the basis for specifying and achieving objectives [11, p.2], in accordance with the saying 'You can't control what you can't measure' [10]. Measurement must be goal-oriented [11, p.21]. A well-established approach for definition of goal-oriented metrics is the Goal Question Metric (GQM) paradigm introduced by Basili and his colleagues [4], which consists of three steps:

1. **Goal.** Which goal shall be achieved by the measurement?
2. **Question.** Which question(s) shall be answered by the measurement?
3. **Metric.** Which metric(s) are suitable to answer the question?

In project management, metrics can provide essential decision support: Project managers use metrics to observe the status of their project, including cost, time, and quality aspects.

Social psychologists have done comprehensive research by studying "the manner in which the personality, attitudes, motivations, and behavior of the individual influence and are influenced by social groups" [25]. This includes measurement of collaboration. Already in the 1930s Moreno [28] developed the so-called *sociometry* or *social network analysis*: Based on communication data the communication relationships between team members are analyzed. The resulting network can be displayed as a graph, and analyzed applying methods of graph theory. The social network analysis approach has been adapted by GSE researchers for a 'post mortem' analysis of completed projects. Ehrlich and Chang [12] introduced six categories for social network analysis in GSE projects: *communication* (how often have you communicated with this person?), *availability* (how easy is it for you to reach this person?), *general awareness* (how aware are you of this person's professional background?), *current awareness* (how aware are you of the current set of tasks that this person is working on?), *familiarity* (how closely did you work with this person on your last project together?), and *importance* (how important is it for you to interact with this person?). Damian and her colleagues conducted social network analysis in various contexts, e.g., analyzing communication relationships in so-called requirement-dependency social networks (i.e., social networks of teams, which work on requirements that are dependent on each other) [22].

Project retrospectives [19] are dedicated team retreats which are used to reflect on issues in team collaboration inside a software engineering team. The focus of a project retrospective is on learning, not fault-finding. The idea of project retrospectives is also embedded in the principles of agile development (see agile manifesto): "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly." [6]. Kerth provided a comprehensive handbook for how to conduct project retrospectives in software engineering, including proven team exercises [19]. In this context, Kerth [19, p.42] described 13 characteristics for observation of elements of dysfunctional and functional cultures in software development organizations.

4 GSE Metrics

In the following we introduce a measurement model for Global Software Engineering projects and define a set of collaboration-based metrics to be used by project managers. We based the design of the GSE metrics on the GSE challenges described in the previous section, and in particular the following sources: 1) the social network analysis methodology for GSE as described by Ehrlich and Chang [12], 2) a GSE survey conducted by Herbsleb and Mockus [16], and 3) characteristics of dysfunctional cultures as described by Kerth for project retrospectives [19].

The model of GSE metrics consists of three levels: Project level, Interac-

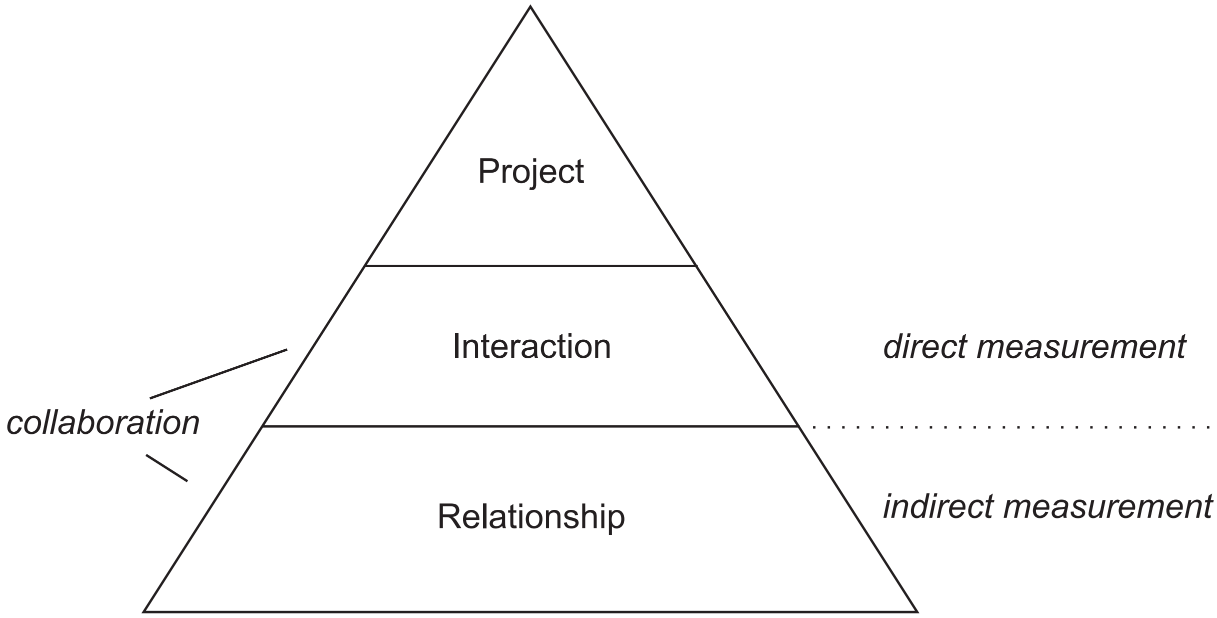


Figure 2: Model of GSE Metrics: The GSE metrics pyramid

tion level, and Relationship level. The first level, the *Project level*, represents traditional measurement of cost, time, and quality status [11][24]. Since these metrics are state of the art today, they are out of scope of our research.

In addition to the Project level we introduce two more levels in our model to measure collaboration, the *Interaction level* and the *Relationship level*, which are not state of the art today. They refer to both interaction (e.g., communication) and relationship (e.g., team cohesion) between team members of a globally distributed development project. The three measurement levels of our model are illustrated in Figure 2.

Project level. This level refers to traditional project management metrics for progress monitoring, which are used by project managers in industry today. Measurement on the Project level usually includes cost, time, and quality metrics for monitoring the project staying within the budget (budget compliance), keeping milestone dates (schedule compliance), and achieving quality goals (defect rates). Also more sophisticated metrics like *earned value* [11, pp.212–217] are contained in this level.

Interaction level. The Interaction level aims at measuring visible interactions between project team members such as (formal) communication, traveling, and delay of work items due to missing information. It includes the following metrics:

- Communication Frequency: How often do team members communicate?
- Reachability: Are team members easy to reach?
- Communication Media: How much communication is done via medium X?
- Delay Due to Missing Information: How often and how long was work delayed because team members had to wait for information from others?
- Traveling: How many team members have visited other development sites and how much time have they spent there?

For metrics of the Interaction level a *direct measurement* is possible, e.g., by analyzing e-mail data, traveling records, or dates in a change management system. Because of the *direct measurement*, data required for Interaction metrics can be collected either manually or automatically.

Relationship level. This level refers to measuring interpersonal relationship between team members. The following metrics are defined on Relationship level:

- Team Cohesion: How is the perceived team cohesion?
- Informal Communication: Do team members have informal communication?
- Culture and Language: What is the impact of culture and language skills?
- Awareness: Do team members know who to contact, are they aware of what others are working on?
- Team Satisfaction: How comfortable do team members feel in the project team and how satisfied are they with the project's achievements?

It is not possible to measure this data directly or derive it from existing data. Instead only an *indirect measurement* is possible by asking or observing the team members, for example by surveys or interviews, which implies a manual data collection. Please note that although the measurement questions are formulated as closed questions, they are supposed to be answered by a degree of confirmation (e.g., *fully agree, tend to agree, tend to disagree, fully disagree*).

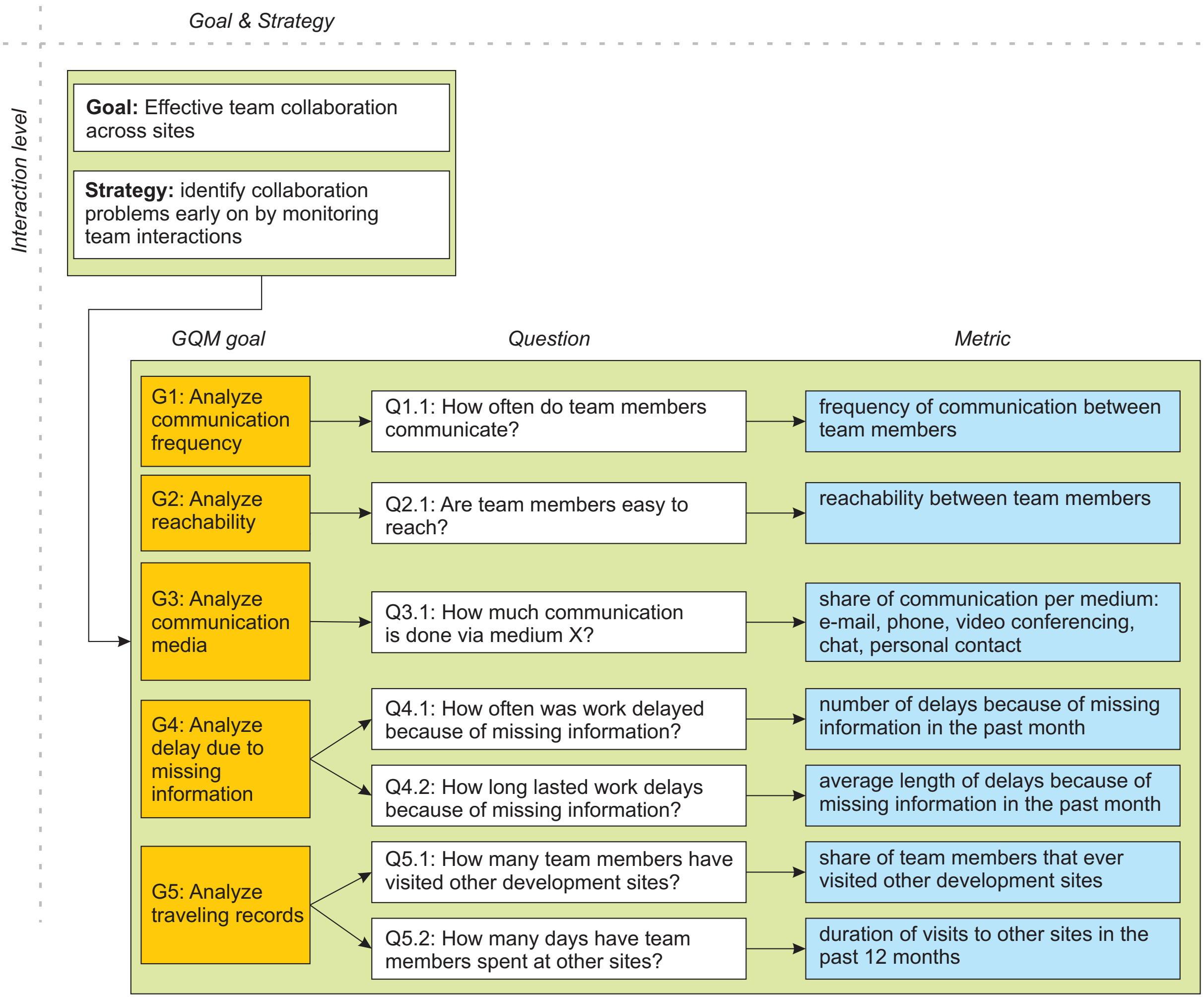


Figure 3: Metrics on Interaction Level

For the definition of metrics, we adapted the GQM$^+$Strategies approach, an extension of the Goal Question Metric (GQM) approach [3]. For instance, in Figure 3 the metrics we define on the Interaction level are shown, including the detailed measurement goal, question, and metric. The overall goal of Interaction level metrics is to achieve effective team collaboration across sites. It is to be achieved through the strategy to identify collaboration problems early on by monitoring team interactions. This is further detailed in the measurement goals G1 to G5. For each measurement goal there are one or more measurement questions defined, which are translated into metrics. For example, the measurement goal 'Analyze traveling records' is associated with the two questions 'How many team members have visited other development sites?' and 'How many days have team members spent at other sites?'. The corresponding metrics are the 'share of team members that ever visited other development sites' and the 'duration of visits to other sites in the past 12 months'.

The metrics definitions were further detailed into formal definition sheets with the description of the indicator, business goal, strategy, measurement goal, measurement questions, definitions and formulas, the base measures to be collected, guidance for interpretation, a sample diagram, as well as a list of references.

5 Case Studies

To validate our approach, we applied our GSE metrics in multiple projects and evaluated their effectiveness for GSE project management. In a first case study, we introduced the metrics to four GSE projects. In a second case study, we applied the GSE metrics to a large co-located project in order to be able to compare the globally distributed and co-located setting. In the following, we give an overview of the two case studies and describe the data analysis and the results obtained.

5.1 Case Study 1: NEREID

In summer 2008 Prof. Laurini from INSA (Institut National des Sciences Appliquées) de Lyon, France initiated a university course coined by the name Network of Engineering univeRsities Educating in Intercultural Design (NEREID). Ranging from September to January, the course aimed at teaching students project management on GSE projects, targeting at late bachelor and master students. Originally, NEREID was started as a bilateral cooperation between INSA de Lyon and Tecnológico de Monterrey campus Puebla, Mexico. However, the number of NEREID participants quickly increased admitting Technische Universtität München, University of Applied Science Esslingen (both Germany), as well as Universidad Tecnica Federico Santa Maria in Chile. Focusing on the development or improvement of an existing business application, involving external companies as real customers, the course's curriculum included the design and implementation of software, the compilation of a final report, and the presentation of the final results. This enabled the distributed student teams to gain hands-on experience required when working in an international setting. Further details with regards to the course, roles, and deliverables are described in an earlier publication [23]. For the course's 3rd edition during the winter term 2010/11, above introduced metrics were applied to 4 of the 8 project teams. Consisting of 4–9 student members who were located at 2–4 different sites in Chile, Mexico, France, and Germany, the teams were exposed to realistic GSE challenges often encountered in industrial projects. The four teams had four different tasks to solve: The team *Tricia Google Translate* had the task to extend Tricia – an open source knowledge management and collaboration tool, which provides wikis, blogging, social networking, file sharing, etc. but only with an English user interface – with multiple language support by means of the Google Translate API. The task of the *Touch Screen Tablet* team was to develop a touch screen menu system for restaurants. The team *Electronic Billing* worked on an electronic billing system for a company in Mexico, where a new law required companies to provide electronic invoices effective January 2011. Finally, the team *Carpooling and Ride Sharing* had to develop an Internet platform as market place for ride sharing (i.e., offer and search for rides).

The GSE metrics were applied twice: at the beginning of November 2010, shortly after the first interim presentation by the students and at the beginning of December 2010, about two weeks before the end of the projects. The students had about one week time to fill in an online survey for collection of the GSE metrics data. The results of each application were directly provided to the teams, helping them to adjust their project management style immediately. The data collection was done via an online survey platform (LimeSurvey) [1]. The answering of the survey was anonymous.

At the end of the projects, we conducted semi-structured interviews with the student teams in Munich and used an online questionnaire to collect detailed feedback from the students at all four sites involved in the projects to assess the accuracy, usefulness, and the effort involved in our approach. Additionally, we asked the supervisors to fill in a feedback questionnaire. We organized the interviews and feedback surveys after the students had submitted their work and final reports and received their grades, in order to prevent that they would have reservations to speak openly. Additional sources of feedback were the final reports compiled by the student teams.

5.2 Case Study 2: DOLLI4

The DOLLI (Distributed Online Logistic and Location Infrastructure) student courses are a series of projects conducted by the Technische Universität München in collaboration with the Flughafen München GmbH (FMG), the operating company of the Munich airport. In the DOLLI student courses, the FMG provides students with the opportunity to work on problems of real interest to the FMG in the form of large scale (30–50 students) project courses in industrial application landscapes [7]. The students are expected to complete the whole software development lifecycle from requirements elicitation to design, construction, and finally delivery of their results in the course of one semester, all in a self-organized way. The purpose of these courses is to let 3rd semester undergraduate students, most of which are without any industrial expertise, gain experience in real-world projects. This has proven to be quite successful, as the results of several of these projects have made it into productive use at the airport. The DOLLI4 project was the 4th installment of the series, held in the winter term of 2010 to 2011 with 30 participants distributed over 4 teams working in the fields of facility automation, data visualization, and mobile development. Since all participants were students at Technische Universität München and the client was also located in Munich this was not a real GSE project; nevertheless, since the students were only working part time on the project and mostly from home and not in a lab, the GSE metrics were employed to monitor communication between teams as well as with the customer. The results were used to improve inter-team communication and to facilitate closer coordination with the client.

DOLLI4 consisted of the four teams: Building Management, Security, MUC App, and Landside Server. The Building Management team had to develop a system to enable the employees in the airport offices to control the lighting and air conditioning inside their office rooms from a web browser. The task of the Security team was to develop a system to notify the security staff which is patrolling at the airport about alarm events and to instruct them for inspection of the situation. The objective of the MUC App team was to develop a mobile application for the Apple iPhone to provide airport-related information to passengers or other airport visitors, e.g., flight status information or a shopping guide. Finally, the Landside Server team was responsible for the interfacing with the real data sources in the airport IT systems, i.e., to provide the server part for the other three teams. For each of the four teams, a *team coach* was established who was responsible for guiding the team and was acting as project manager.

The DOLLI4 project had two major distinct phases: A prototyping phase which started with the project kick-off (October 25, 2010) and ended with the system integration test (February 3, 2011). In this phase the Rational Unified Process was followed with the core process steps business modeling, requirements, analysis & design, implementation, test, and deployment [18]. The second phase was conducted as agile development using Scrum [2] and required full time on-site attendance at the airport (March 7–18, 2011).

The GSE metrics were applied on a monthly basis during the prototyping phase (survey 1–3). In addition, there was a control survey with the same questions at the end of the Scrum phase at the airport (survey 4). Furthermore, we experimented with daily measurements of a small set of metrics during the Scrum phase in conjunction with the daily Scrum meetings. The project ended with the customer acceptance test (March 23, 2011), where the students officially presented their results to the customer and offered exhibition stands to demonstrate and discuss their final products. To gather data regarding our validation criteria – accuracy, usefulness, effort and innovation grade –, we collected the feedback of the team coaches in an online feedback survey, after the whole project had been completed and grades had been established.

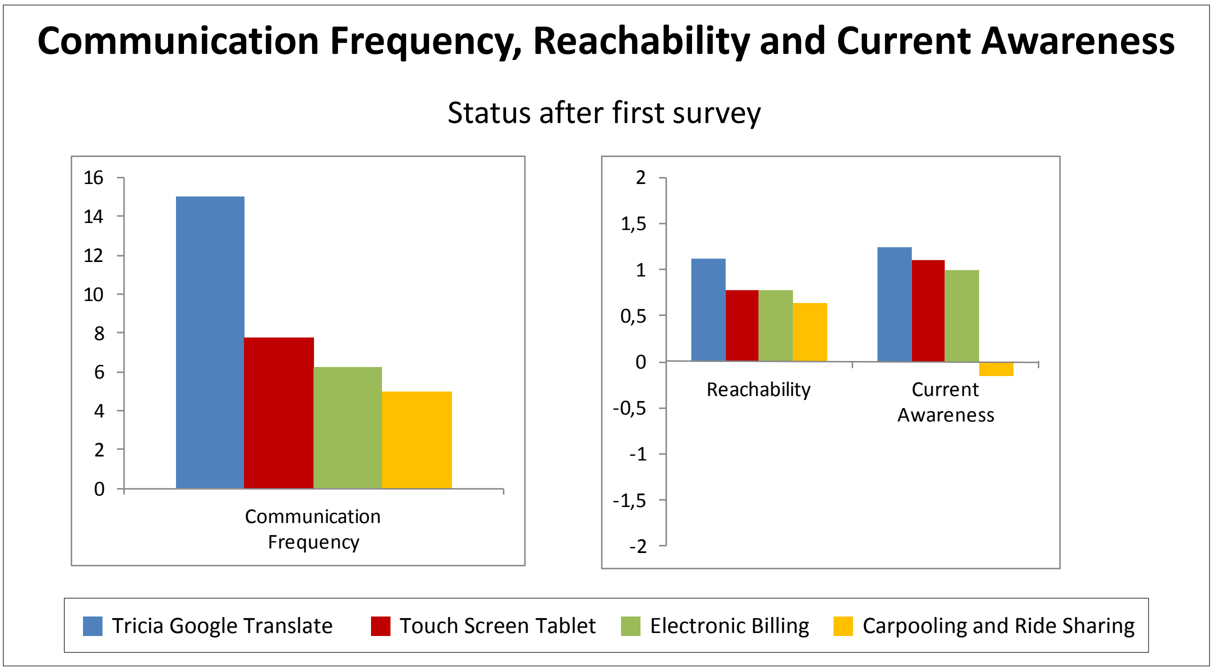
5.3 Data Analysis of Case Study 1 (NEREID)

To assess the reliability of the data collected, we validated the following hypotheses, which reflect the expected behavior in the data:

Hypothesis 1 *Communication Frequency, Reachability, and Current Awareness decrease with the degree of distribution (number of sites, team size).*

Hypothesis 2 *There are more frequent and longer Delays Due to Missing Information across sites than within a site.*

Hypothesis 3 *Team Cohesion, Informal Communication and Awareness are better within the local site than across sites.*



Communication Frequency: daily = 20, every 2-3 days = 8, weekly = 4, every 2 weeks = 2, less than every 2 weeks = 1, not at all = 0
Reachability: very easy = 2, rather easy = 1, rather hard = -1, very hard = -2
Current Awareness: very good = 2, good = 1, bad = -1, very bad = -2

Figure 4: NEREID: Communication Frequency, Reachability, and Current Awareness per project, sorted by degree of distribution

Figure 4 depicts Communication Frequency, Reachability, and Current Awareness from the first survey of the four GSE projects, sorted by degree of distribution (from 2 sites to 4 sites). A clear trend is visible: The higher the degree of distribution is, the less frequent the average communication. Also Reachability and Current Awareness decrease with the degree of distribution. Hypothesis 1 is confirmed.

In all four NEREID projects the average Delay Due to Missing Information revealed more frequent and longer delays across sites than within a site. Thus Hypothesis 2 is confirmed. The average ratio of the duration of distant delays and local delays is 2.4, which even corresponds well to the factor 2.5 identified by Herbsleb [16].

The results for Team Cohesion, Informal Communication, Culture & Language and Awareness of the four GSE projects are illustrated in Figure 5. The ratings of Awareness and Informal Communication are better for the local site compared to the distant sites, which is in conformance with Hypothesis 3. With respect to Team Cohesion, 6 of the 8 subitems show better results for the local than for the distant site. The two exceptions are: no team competition and clear responsibilities. The former is explainable by the fact that there were multiple

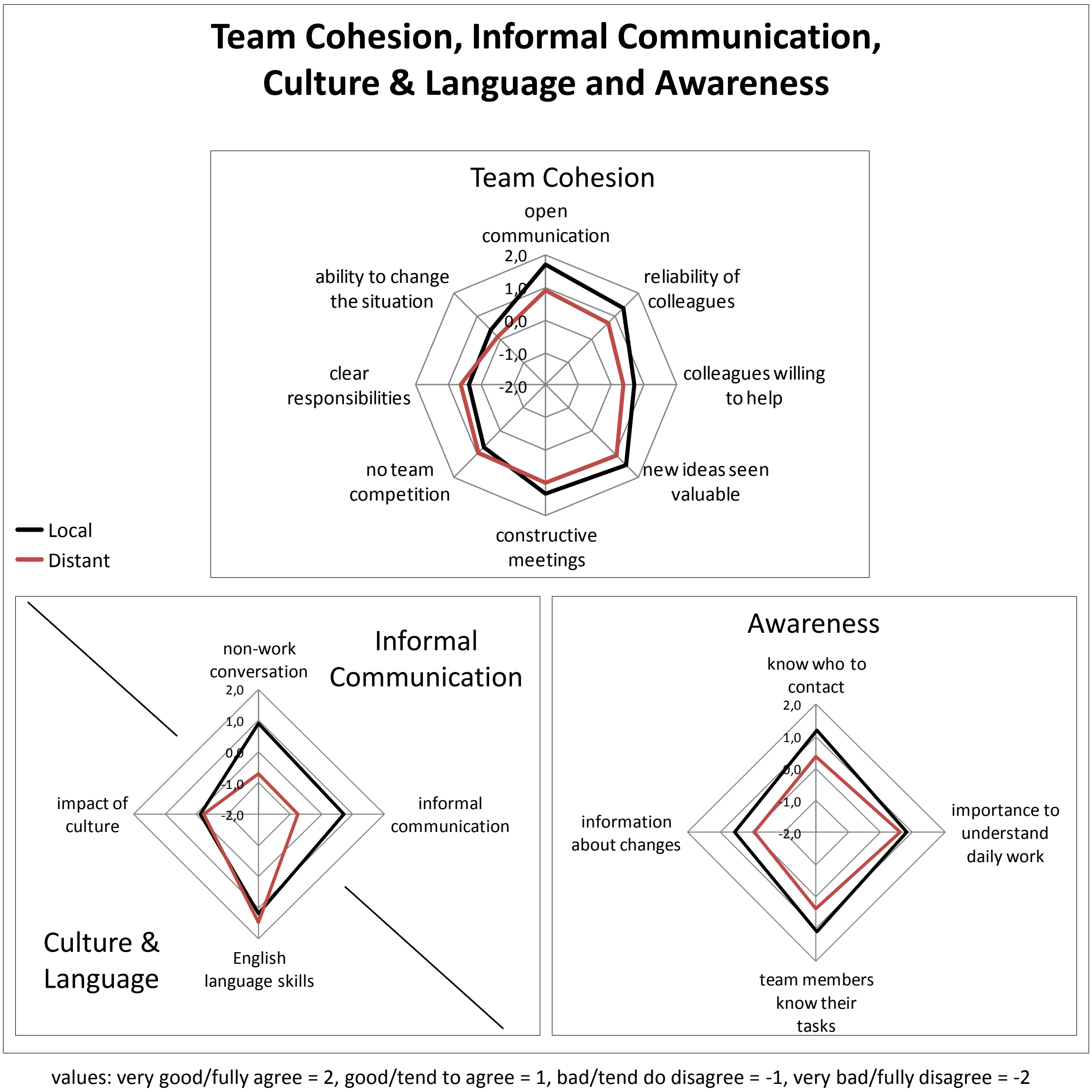


Figure 5: NEREID: Team Cohesion, Informal Communication, Culture & Language, and Awareness, average of all projects

NEREID student projects running in parallel under the same supervisor which was perceived as kind of competition between the teams. The latter can be justified by the fact that teams had established a clear worksplit between sites, but not for individual team members at one site, which lead to increased discussion over who has a particular responsibility within one site. The total value of Team Cohesion is 1.0[2] with respect to the local site and 0.7 for the distant sites. We therefore consider Hypothesis 3 confirmed as well.

Even though the data collected is based on subjective measurements, the data shows the expected behavior with respect to Hypothesis 1, 2 and 3.

Right after each round of data collection via the GSE metrics survey, we prepared a *collaboration analysis report*, which was then distributed to the students of each project team of the four NEREID projects. To each collaboration analysis report, we added our interpretation of the GSE metrics survey data in the form of an executive summary. Figure 6 lists the main findings which we derived from the survey data. Please note that the interpretations were prepared from an 'outsider' perspective, without being involved in what was going on in the project teams, and before the semi-structured interviews took place.

The student feedback survey after the completion of the NEREID projects was answered by 17 of the 26 students which equates to a response rate of 65%. In the first part of the student feedback survey we collected the general feedback of the students: We asked them to evaluate the following statements related to accuracy, effort, and usefulness on a four-point scale:

- The collaboration analysis provided insights into the team dynamics of our project team
- The collaboration analysis accurately reflected the reality observed in our project team
- The effort for data collection was acceptable
- The collaboration analysis was useful for our project team

Additionally, we asked them how much time they needed to fill in the GSE metrics survey. Figure 7 contains the feedback results. The overall feedback was very positive: all four statements were confirmed at an agreement level (*fully agree* and *tend to agree*) of 81% to 88%. The answers regarding the time needed to fill in the GSE metrics survey were between 8 and 25 minutes, where

[2] The range of values is from +2.0 to –2.0, where +2.0 is the best and –2.0 is the worst value.

Tricia Google Translate

- S1.1 From Survey 1 to Survey 2 the collaboration has worsened
- S1.2 At the time of Survey 2, the majority of the team was unsatisfied with the project's achievements
- S1.3 Survey 2: There was honest and open communication within the local team, but not with the distant colleagues
- S1.4 Survey 2: Distant colleagues didn't provide timely information about changes in current plans
- S1.5 There was almost no informal communication across sites
- S1.6 English language skills were no major problem

Touch Screen Tablet

- S2.1 From Survey 1 to Survey 2 the overall status of collaboration was more or less constant
- S2.2 Communication inside the team was honest and open
- S2.3 There was almost no informal communication across sites
- S2.4 The majority of the team felt powerless to change the project's situation
- S2.5 At the time of Survey 2, the collaboration between Valparaiso and the other sites was particularly challenging
- S2.6 There were some issues related to English language skills

Electronic Billing

- S3.1 There were issues with communication and reachability across sites
- S3.2 There was almost no informal communication across sites
- S3.3 At the time of Survey 2, there were issues with timely information about changes by the distant colleagues
- S3.4 The team felt rather powerless to change the project's situation
- S3.5 English language skills were no major problem

Carpooling and Ride Sharing

- S4.1 From Survey 1 to Survey 2 the collaboration has improved
- S4.2 At the time of Survey 2, most of the team knew what other colleagues are currently working on
- S4.3 Survey 2: The majority of the team was satisfied with the project's achievements
- S4.4 There were local problems inside the Munich team
- S4.5 English language skills were no major problem

Figure 6: NEREID: Main findings based on GSE metrics data

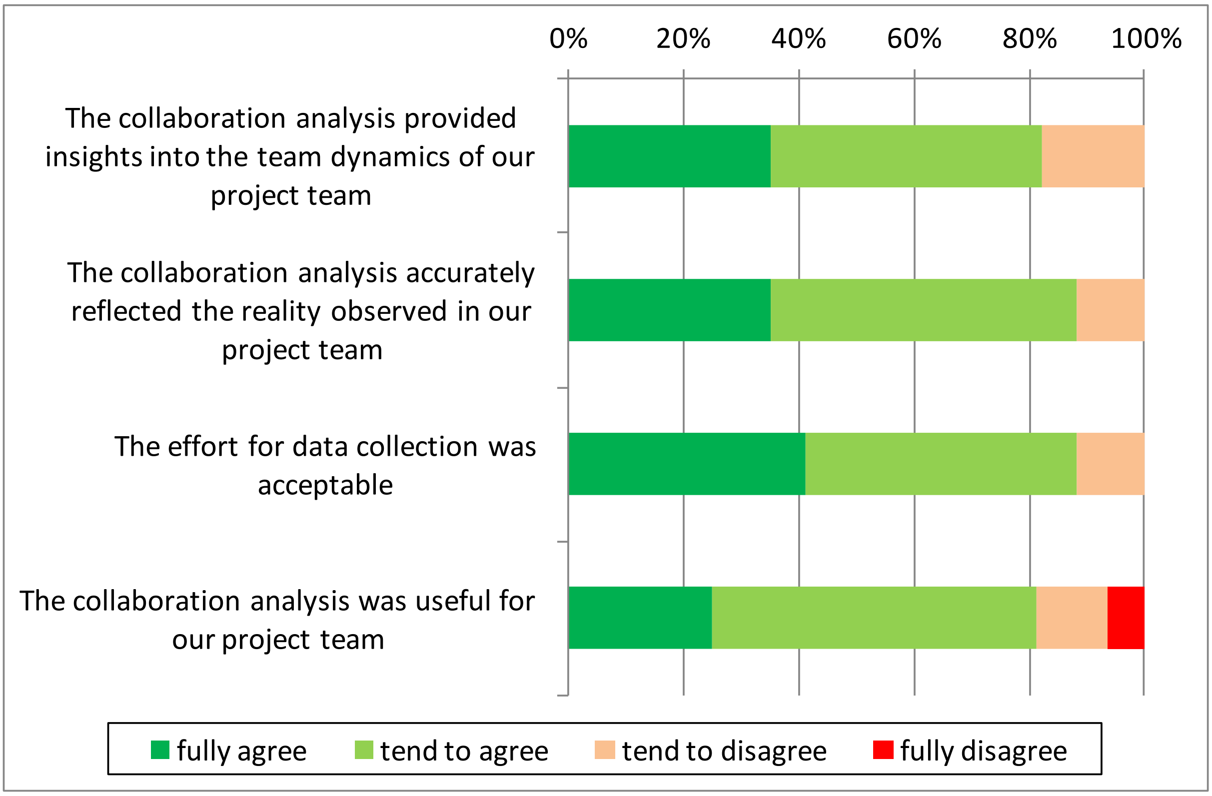


Figure 7: NEREID: General feedback from student feedback survey

14.1 minutes was the average. This was seen as acceptable according to the feedback of the students, see Figure 7.

Furthermore, we asked the team members to evaluate the main findings from our interpretation (see above). The results are shown in Figure 8. The majority of our interpretations were confirmed to be correct: the degree of confirmation (share of answers that *fully agree* or *tend to agree*) ranged between 68% for the Electronic Billing team to 100% for the team Carpooling and Ride Sharing. It is to be noted that some findings were controversial to the team members: Statement S2.4 (*The majority of the team felt powerless to change the project's situation*) of the team Touch Screen Tablet was confirmed by half of the team and rejected by the other half. The explanation we found from the interviews was that the customer which was an external company in Mexico was seen by some team members as a higher authority which was setting targets and frame conditions that could not be changed by the project team, while others did not see a problem. In addition, the statement S3.2 (*There was almost no informal communication across sites*) is striking since it is the only finding which team members disagreed more than agreed with. Here an explanation is that we formulated our finding in an too extreme way, which the team members

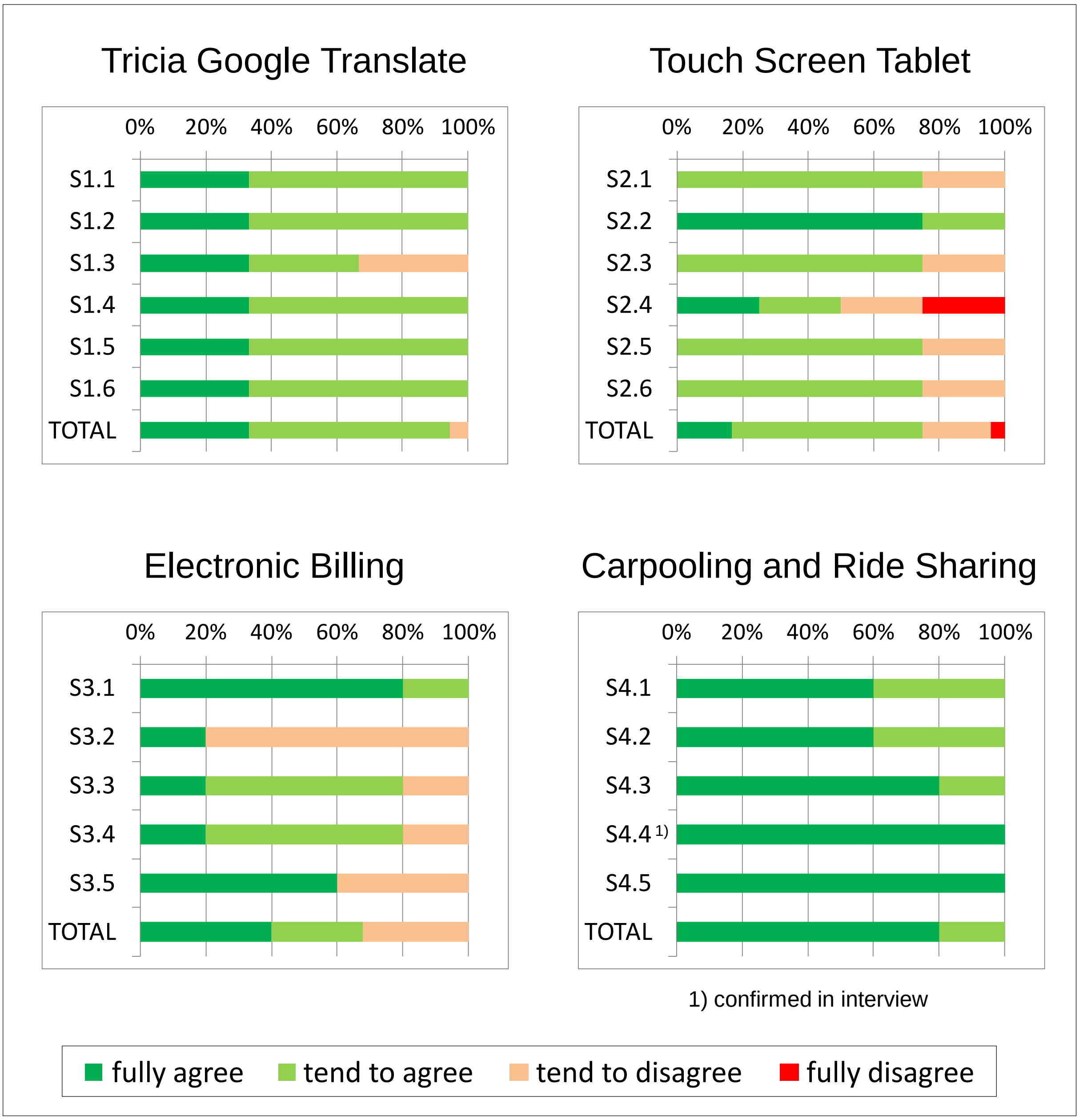


Figure 8: NEREID: Evaluation of main findings from student feedback survey

could not agree with. In fact, in the two GSE metrics surveys, the majority answered the questions regarding *I discuss non-work related matters with other project team members* and *I often get useful work-related information through informal communication (e.g., hallway conversations)* negatively for the distant site. However, even though informal communication was impeded, there was no complete lack of informal communication across sites.

From the GSE metrics data – without knowing what was going on in detail in the projects –, we had observed that at the time of survey 2 in the team Touch Screen Tablet the collaboration between Valparaiso (Chile) and the other sites was impeded, we interpreted it as 'particularly challenging' (See S2.5). As it turned out from the interviews, two students from Chile had left the project team because of a late technology switch from Android/Java to PHP according to a customer request. However, these two students had contributed to the project before and there were no major collaboration issues. The reason why the metrics indicated an abnormal behavior was the absence of the Chilean team after this event.

Besides the evaluation of the main findings, we also asked the team members to rate the accuracy (*How accurately did the indicators reflect the reality observed in your team?*) and usefulness (*How useful are the indicators?*) of the GSE metrics. Altogether it was stated in the feedback that the indicators reflected the team reality well: Positive accuracy ratings of the indicators (*accurate* and *rather accurate*) range from 94% to 69%, the average of all indicators is 80%. Positive ratings of usefulness (*useful* and *rather useful*) are between 100% and 60%, the average of all indicators is 86%. While seven of the ten indicators received very high ratings (4 times 100%, twice 93% and once 86%), three of the indicators were seen less useful: Communication Media Profile (69%), Culture & Language (64%), and Informal Communication (60%).

In an open question we asked the students to describe how they used the collaboration analysis reports: *Please describe shortly what your team did with the two collaboration analysis reports (e.g., discuss them in the team, take action).* 83% of the respondents who answered this question mentioned that they had discussed the collaboration analysis reports within their teams. In 67% of the answers it was mentioned that the analysis was useful to the team resp. that the team had used it to take corresponding action based on the results. This was also underlined by the feedback statements, for example:

> "Through the analysis report, we have found common problems which trouble our teamwork. The efficiency has improved a lot after that.

> Moreover, we communicate more often with other team members, which is a great help for improving our working quality."

We asked the supervising professors to rate the innovation grade of our approach. All four supervising professors saw the collaboration analysis approach as new in GSE and therefore as innovative.

5.4 Data Analysis of Case Study 2 (DOLLI4)

Altogether, the collaboration went smoothly in DOLLI4. However, one major issue with the cross-team collaboration was detected in the analysis: There were collaboration problems between the Landside Server team and the other teams, in particular low awareness of the work of the others and low communication frequency, which was critical because the server part had to integrate well with the client part and vice versa. As a consequence, the team coaches and the project teams finally decided to introduce a major reorganization: Between survey 2 and 3, the Landside Server team was dissolved. The former team members of the Landside Server team were integrated into the other three teams. This is in line with Conway's law according to which "a design effort should be organized according to the need for communication" [9]. However, this organizational requirement was not recognized prior to the collaboration analysis.

Figure 9 and Figure 10 provide an overview of the results of the GSE metrics surveys in DOLLI4. From these figures, the development of the team situation over time is visible. Communication Frequency and the number of Delays Due to Missing Information have increased steadily over time. Also the Team Cohesion, Informal Communication, Awareness, and Current Awareness with respect to the other teams grew steadily. These effects are explainable by the increased intensity of collaboration and also by the reorganization of the Landside Server team between survey 2 and 3, which improved cross-team collaboration. It is also obvious that the situation at the time of survey 4 was much different from the situation during the previous three surveys. The reason is the difference between the prototyping phase and the on-site Scrum phase. During the Scrum phase, Communication Frequency increased heavily, Reachability was optimal within and between the teams, and also the Reachability of the customer was much better (even though not as good as within and between the student teams), and as it can be seen from the Communication Media Profile almost only personal communication was used. Due to the intensive work, the number of Delays Due to Missing Information increased, however, the duration of Delays Due to Missing Information dropped notably. The number of meetings increased as well. On the Relationship level, the values related to other teams improved significantly according to survey 4, i.e., Team Cohesion, Informal Communication, Awareness, and Current Awareness. Noticeable is also that the Team Cohesion inside the own team went down slightly. The intensive work in the Scrum phase and the prevailing time pressure increased tensions between the team members, e.g., if somebody made changes to the code which caused problems to another team member's work or broke the build.

6 Reflection

In summary, our approach of the collaboration-based project management metrics was successful in both the GSE projects (NEREID) as well as the co-located student project (DOLLI4) which we studied during the validation phase. In total the metrics reflected the reality in the project teams well. Furthermore, they were seen as useful by the teams and the effort involved was acceptable. The finding that the GSE metrics were useful was also expressed through the improvement actions taken by the teams, e.g., the reorganization of the DOLLI4 teams after the collaboration analysis revealed collaboration deficiencies. Our approach was rated to be very innovative and new to GSE project management. In the following we present lessons learned from our studies. Because of the usage of our metrics during project lifetime instead of a post mortem analysis, the following aspects were particularly important from our point of view:

Just enough measurement. As data collection and analysis can cause high effort during project lifetime, it is important to define a suitable measurement strategy. In our experience it is more appropriate to have selective measurements and random samples instead of an all-embracing approach. The set of metrics in our GSE model and the effort associated with collecting this data was acceptable according to the feedback which we received from the validation projects. However, if the frequency of measurements should be further increased (e.g., from monthly to bi-weekly or weekly) towards real-time feedback, it would be crucial to further reduce the effort for data collection and analysis per survey.

Privacy and open feedback. It is essential to preserve a relationship of mutual trust with the team members and use the collaboration analysis results only for the good of the team. If the team members have to fear negative consequences,

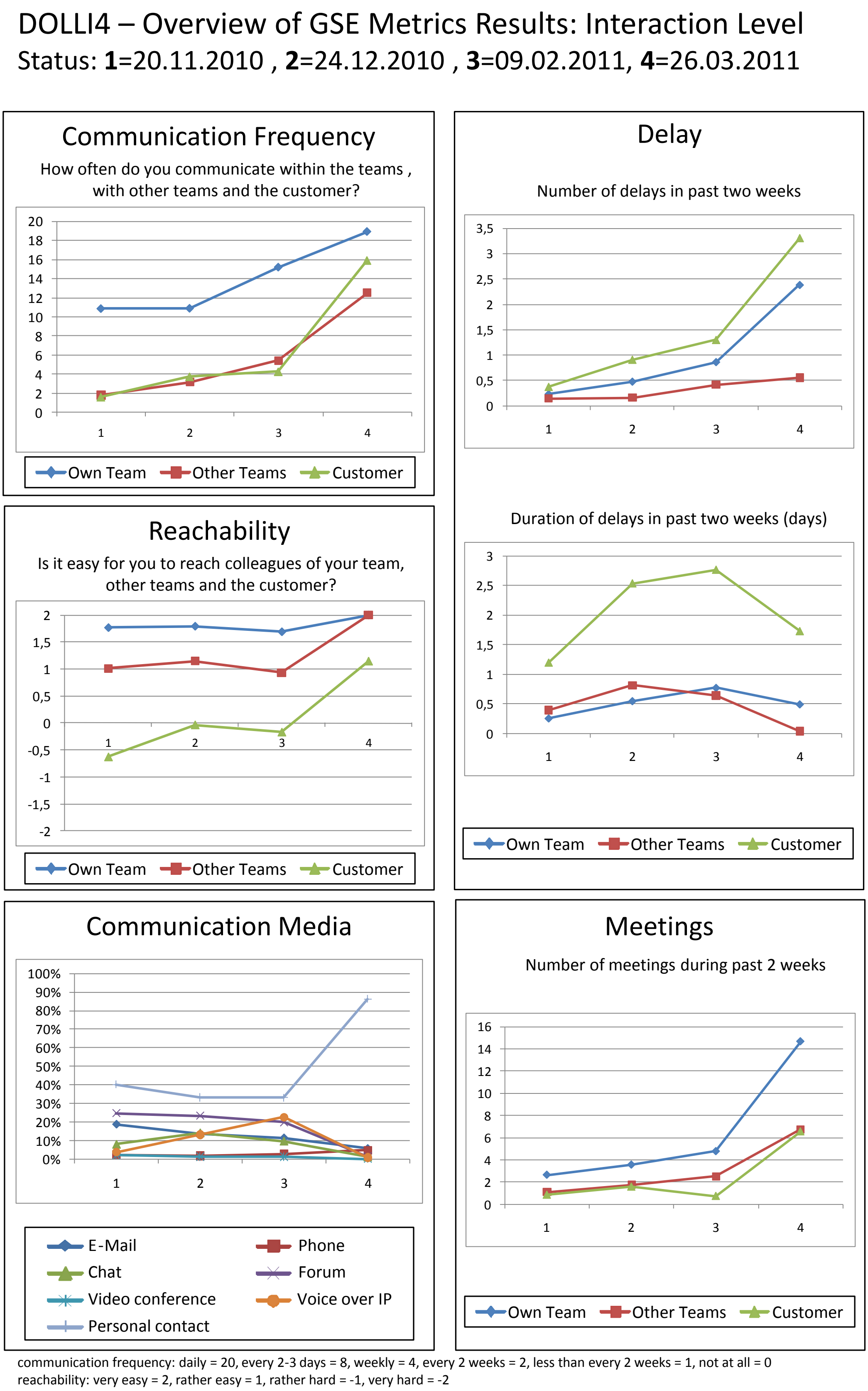


Figure 9: DOLLI4: Overview of GSE metrics results of survey 1–4, Interaction level

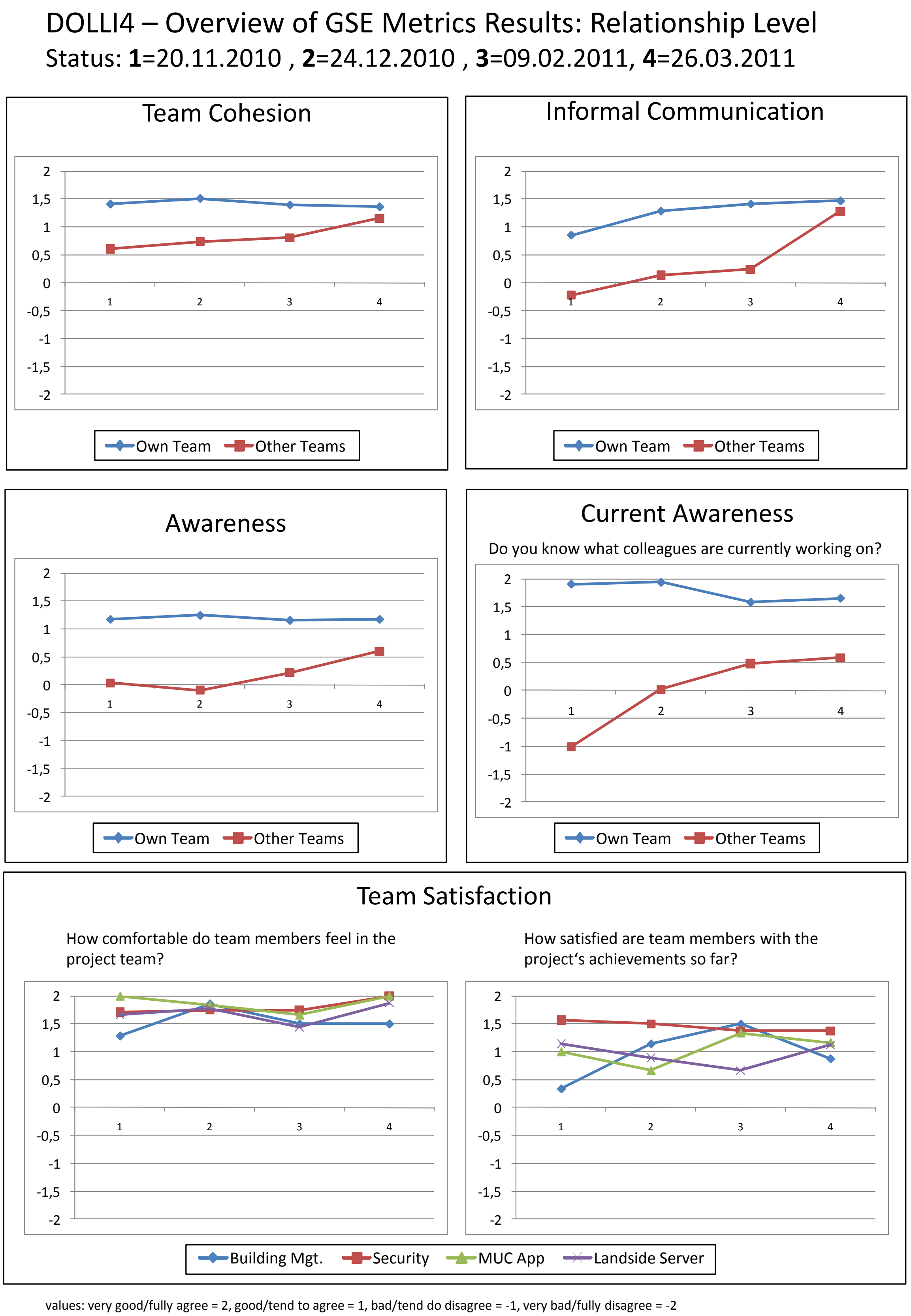


Figure 10: DOLLI4: Overview of GSE metrics results of survey 1–4, Relationship level

they will not answer openly. For example the daily feedback in the on-site Scrum phase of DOLLI4 were done with flipcharts and thus it was visible to all team members who was voting in which way. Team coaches reported in the feedback that their team members did not answer the question *How comfortable do I feel in my team?* openly because they did not want to be conspicuous and found it embarrassing to be asked in front of the other team members to explain their rating, if they voted differently from the others. However, open feedback is essential to get the real picture, therefore the data should be collected anonymously. Furthermore, one should be careful with setting target values, because the survey respondents may easily 'fake' the data just to achieve the target.

Depending on the legal situation of a certain country, data collection of personal data is restricted or prohibited. In this case, the data needs to be anonymized and kept confidential, or the approval of the team members is a necessary prerequisite for data collection. In the two case studies, social network analysis data was collected on team resp. site level instead of individuals. Our original motivation was to protect the privacy of the team members. However, it turned out that this reduction in granularity actually was beneficial because it also increased the lucidity (simple graphs) and reduced the effort for analysis, while the results were still meaningful.

We experienced that the introduction of our GSE metrics in case of the globally distributed projects was more challenging than in the co-located project because of the discussed effects of distribution. In contrast to the DOLLI4 project, we did not have the chance to meet the NEREID project team members in person and needed to communicate mostly via e-mail, and there was a risk that project team members would not participate in the surveys. Therefore one should not underestimate the impact of distribution also when introducing such a measurement system. In case of the NEREID projects, we argued via the benefits for the team members (they could use it to improve their projects and finally to deliver a better result), but did not exert pressure on them.

7 Threats to Validity

In this section we discuss threats to the validity of our results.

7.1 Construct Validity

A central threat to validity of our study is the *subjectiveness of data*. The data collection for our GSE metrics was done based on survey questionnaires that were filled in by the team members of the projects under observation. Consequently, the data reflects the subjective view of the respondents. There is even the risk that project team members just checked arbitrary answers in the survey questionnaire. However, although there might be a certain fuzziness in data collected by surveys, the project team members and team coaches clearly confirmed the accuracy and usefulness of the metrics in their feedback as well as in the interviews, as we described above. To probe the reliability of the data, we established three hypotheses of expected behavior, which were confirmed by the data. This would not have been the case, if the data was arbitrary or meaningless. Also the findings from the interviews were consistent with the data.

7.2 Internal Validity

A core problem in the interpretation of the survey results was the lack of context information: Because we were not part of the team, there is the risk that interpretations may be weak. One example we have seen above was the anticipated collaboration problem between Valparaiso and the other sites in the Touch Screen Tablet project of NEREID, where the real cause was that the team members from Valparaiso had left the team due to technical reasons. It is to be noted that the GSE metrics are intended for project managers, insofar the situation in the research setting was artificial. If the approach is used by project managers, they also have the required context information at hand.

7.3 External Validity

In our work a small set of projects was analyzed, which provides anecdotal evidence only. However, although the set was small, we tried to cover multiple project types. Although there is the risk that the student projects might be not representative for the reality in industry, we found that these projects were facing quite realistic challenges, comparable to collaboration challenges in industrial GSE projects.

8 Conclusion and Future Work

In this paper, we introduced new project management metrics for GSE based on collaboration. We described the application of the GSE metrics in two case studies (i.e., five disjunct projects). Despite the threat of subjective data and a certain fuzziness, the results were useful and accurately reflected the reality. In particular, the GSE metrics helped the teams to identify, discuss and mitigate collaboration problems. At the same time, the effort for data collection was reasonable. In summary, collaboration-based metrics can be recommended as extension of traditional project management metrics in GSE.

In future work, we plan to *quantify the benefit of improved collaboration*: The approach presented in this paper is based on the assumption that improved collaboration also implies improved project performance. Future work should explore the relationship of collaboration-based metrics and project performance in terms of cost, time, and quality to quantify the benefits. To get a broad experience on the application of the GSE metrics, a *summative evaluation* with a large set of case studies is needed. This will allow to build an experience database with reference values for future measurements. Furthermore, *alternative ways of data collection* can be investigated. The data collection in the two case studies relied on survey questionnaires. Future research should explore deriving the metrics data from existing data repositories such as traveling records.

Acknowledgment

The authors would like to thank the students and the supervisors of the DOLLI4 and NEREID projects for the participation in the case studies, in particular Prof. Robert Laurini, Béatrice Rumpler, Prof. David Sol, and Prof. José Contreras.

References

[1] LimeSurvey homepage, retrieved September 9, 2012 from `http://www.limesurvey.org`.

[2] Scrum homepage, retrieved September 9, 2012 from `http://www.scrum.org/`.

[3] Victor R. Basili, Caldiera Gianluigi, and Dieter Rombach. Goal Question Metric Paradigm. *Encyclopedia of Software Engineering*, 1:528–532, 1994.

[4] Victor R. Basili, Mikael Lindvall, Myrna Regardie, Carolyn Seaman, Jens Heidrich, Jurgen Munch, Dieter Rombach, and Adam Trendowicz. Linking Software Development and Business Strategy Through Measurement. *Computer*, 43:57–65, 2010.

[5] Matthew Bass, James D. Herbsleb, and Christian Lescher. A Coordination Risk Analysis Method for Multi-Site Projects: Experience Report. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'09)*, 2009.

[6] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Principles behind the Agile Manifesto, retrieved September 9, 2012 from `http://agilemanifesto.org/principles.html`, 2001.

[7] Bernd Bruegge, Helmut Naughton, and Michaela Gluchow. SLPC++: Teaching software engineering project courses in industrial application landscapes – A tutorial. In *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, pages 549–551, 2011.

[8] Erran Carmel. *Global Software Teams – Collaborating Across Borders and Time Zones*. Prentice Hall, 1999.

[9] Melvin E. Conway. How Do Committees Invent? *Datamation*, 14(5):28–31, April 1968.

[10] Tom De Marco. *Controlling Software Projects: Management, Measurement and Estimation*. Yourdon Press, New York, NY, USA, 1982.

[11] Christof Ebert and Reiner Dumke. *Software Measurement: Establish – Extract – Evaluate – Execute*. Springer, 2007.

[12] Kate Ehrlich and Klarissa Chang. Leveraging Expertise in Global Software Teams: Going Outside Boundaries. In *ICGSE '06: Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'06)*, pages 149–158, Washington, DC, USA, 2006. IEEE Computer Society.

[13] John S. Gero and Udo Kannengiesser. The Situated Function–Behaviour–Structure Framework. *Design Studies*, 25(4):373–391, 2004.

[14] Ashok Goel, Spencer Rugaber, and Swaroop Vattam. Structure, Behavior & Function of Complex Systems: The SBF Modeling Language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23(1):23–25, 2009.

[15] James D. Herbsleb. Global Software Engineering: The Future of Socio-technical Coordination. In *FOSE '07: 2007 Future of Software Engineering*, pages 188–198, Washington, DC, USA, 2007. IEEE Computer Society.

[16] James D. Herbsleb and Audris Mockus. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *Software Engineering, IEEE Transactions on*, 29(6):481–494, 2003.

[17] Anne Hoffmann and Christian Lescher. Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS). In *Proceedings of the RE'09 International Workshop on Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS'09)*, 2009.

[18] IBM Corporation. IBM Rational Unified Process, retrieved September 9, 2012 from `ftp://public.dhe.ibm.com/software/rational/web/datasheets/RUP_DS.pdf`, 2007.

[19] Norman Kerth. *Project Retrospectives: A Handbook for Team Reviews*. Dorset House Publishing, 2001.

[20] Christian Lescher. *Collaboration-based Metrics for Project Management Support in Global Software Engineering*. PhD thesis, Technische Universität München, Germany, August 2011.

[21] Christian Lescher and Frances Paulisch. Establishing a Common Software Measurement System at Siemens: An Experience Report. In *Business Process Engineering, Proceedings of the CONQUEST 2007, 10th International Conference on Quality Engineering in Software Technology*, pages 1–15. dpunkt Verlag, 2007.

[22] Sabrina Marczak, Daniela Damian, Ulrike Stege, and Adrian Schröter. Information Brokers in Requirement-Dependency Social Networks. In *RE*

'08: Proceedings of the 2008 16th IEEE International Requirements Engineering Conference, pages 53–62, Washington, DC, USA, 2008. IEEE Computer Society.

[23] Florian Matthes, Christian Neubert, Christopher Schulz, Christian Lescher, Jose Contreras, Robert Laurini, Beatrice Rumpler, David Sol, and Kai Warendorf. Teaching Global Software Engineering and International Project Management: Experiences and Lessons Learned from Four Academic Projects. In *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU)*, 2011.

[24] John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, and Fred Hall. *Practical Software Measurement: Objective Information for Decision Makers*. Addison-Wesley, 2001.

[25] Merriam-Webster. Medical Dictionary, retrieved September 9, 2012 from `http://www.merriam-webster.com/medlineplus/social%20psychology`, 2011.

[26] Merriam-Webster. Merriam-Webster Dictionary, retrieved September 9, 2012 from `http://www.merriam-webster.com/dictionary/cohesion`, 2011.

[27] Merriam-Webster. Merriam-Webster Dictionary, retrieved September 9, 2012 from `http://www.merriam-webster.com/dictionary/collaboration`, 2011.

[28] Jacob L. Moreno. *Who Shall Survive?: Foundations of Sociometry, Group Psychotherapy, and Sociodrama*. Nervous and Mental Disease Publishing Co., Washington, DC., 1934.

[29] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, 1991.

[30] Raghvinder Sangwan, Matthew Bass, Neel Mullick, Daniel J. Paulish, and Juergen Kazmeier. *Global Software Development Handbook*. Auerbach Publications, 2006.

[31] William M.K. Tochim. Research Methods Knowledge Base: Introduction to Evaluation, retrieved September 9, 2012 from `http://www.socialresearchmethods.net/kb/intreval.htm`, 2006.

[32] Paul Watzlawick, Don D. Jackson, and Janet Beavin Bavelas. *Pragmatics of Human Communication: A Study of Interactional Patterns, Pathologies, and Paradoxes*. W. W. Norton & Company, 1967.