1st International Global Requirements Engineering Workshop – GREW'07

Tony Gorschek[1], Samuel Fricker[2,3], Robert Felt[1], Richard Torkar[1], Claes Wohlin[1], Michael Mattsson[1]
[1] Blekinge Institute of Technology, School of Engineering, Sweden,
[2] ABB Switzerland Ltd., Protection & Substation Automation, Switzerland
[3] University of Zurich, Department of Informatics, Switzerland
tony.gorschek@bth.se, fricker@ifi.unizh.ch, {robert.felt,richard.torkar,claes.wohlin,michael.mattsson}@bth.se

Abstract

GREW´07 was held in conjunction with the International Conference on Global Software Engineering in Munich Germany. The aim was to bring researchers and industry practitioners together to discuss the area of global product development from a requirements engineering and product management perspective. The workshop aimed to analyze selected challenges put forward by accepted papers from both industry and academia. The session discussions then focused on identifying future needs for research, the relevance of which was assured by good industry presence at the workshop. The workshop resulted in a number of findings that can play an important role to further develop the field of global product management and requirements engineering.

Keywords: Global Product Management, Global Requirements Engineering, Distributed Software Engineering, Distributed Product Development, Globalization, and Outsourcing.

Introduction and background

Distributed multi-site software product development is increasingly becoming commonplace as companies become global not only in terms of customer base, but also with regards to large parts of the software product development that is spread over countries, continents and cultures. Distribution is driven by that it enables companies to leverage their resources, and to draw on the advantage of proximity to customers and markets during large-scale software development [1,6,4].

These potential opportunities come with new challenges for the product planning, management and development organizations of a company that affect the requirements engineering of software products. The threat of defect increase and in multi-site development has been documented in literature [7]. According to industry experience reports, some of the main problems are attributed to heterogeneous understanding of requirements and substantial differences in domain understanding and interpretation [2,7, 5]. This is further complicated by the fact that multi-site development usually is detrimental to informal communication between stakeholders like product managers, experts, and developers, as these roles are often separated geographically [4].

The goals of the Global Requirements Engineering Workshop (GREW) was to identify, report, discuss, and address the challenges associated with requirements engineering (RE) and product management (PM) from two main perspectives:

(i) PM/RE for Global Software Development – assuring that the handling of requirements and products are effective and efficient in relation to a global/distributed development environment where development is conducted over multiple sites, and

(ii) Distributed PM/RE – conducting PM and/or RE activities over globally distributed sites, with software development conducted in one single site or distributed over a number of sites.

Sessions

The workshop consisted of three major sessions. Every session started with presentations that were followed by ample time to discuss the session topic. These discussions aimed at sharing positions of the workshop participants, experiences, challenges, and research results and ideas. Both groups, industry representatives and researchers, contributed greatly to the discussions.

The following presentations were given:

Product Lines for Global Markets:
H. Cho: Requirements Management in Software Product Lines.
A. Thurimella, T. Wolf: Issue-based Variability Modeling.

Globally Distributed Communication:
T. Mallardo, F. Calefato, F. Lanubile, D. Damian: The Effects of Communication Mode on Distributed Requirements Negotiations.
I. Kwan, D. Damian, S. Marczak: The Effects of Distance, Experience, and Communication Structure on Requirements Awareness in Two Distributed Industrial Software Projects.
D. Gumm: A Model of Requirements Engineering at Organizational Interfaces: An Empirical Study on Distributed Requirements Engineering.
K. Hermann: Release Planning in Distributed Projects.

Challenges of Global Requirements Engineering: Consequences for Research:
T. Illes-Seifert, A. Hermann, M. Geisser, T. Hildenbrand: The Challenges of Distributed Software Engineering and Requirements Engineering: Results of an Online Survey.

The following subsections summarize the discussions and the conclusions that were drawn from the three workshop sessions.

Product Lines for Global Markets

This session focused on addressing the definition, use, and evolution of product lines in a multi-site distributed environment. The discussion topics included:

1. scalability of methods and tools to the needs of (globally distributed) industry,
2. the role of decision rationale in distributed development,
3. design and selection of tools for globally distributed collaboration, and
4. tailoring the presentation of global data for different roles and decisions.

(1) Scalability

Most of the models, techniques, methods, processes (called *technologies* for the remainder of this report) and tools presented by researchers as solutions to industry problems do not scale to the

needs of industry and/or globally distributed companies, hence will be difficult to be applied. For example, it is not unusual that a company has to handle thousands or even tens of thousands of requirements per year. Most of the presented requirements engineering tools and techniques disintegrate when faced with large amounts of requirements. Researchers need not only to take scalability into account when formulating potential solutions, but should also validate the new technologies in industry to assure e.g. scalability.

Another aspect of scalability raised was cost. Many methods for specifying requirements are too resource intensive and are not adapted for large scale requirements engineering.

(2) Store and Reuse Decision Rationale
The discussion highlighted the importance of capturing and making available decision rationale associated with the requirements. Storing such decision rationale in rich, but relatively informal format seems to be a key factor for its successful use. This ensures that decisions are reusable. Hence effort is not wasted on discussing and taking one decision several times. This also contributes to enhanced understanding and improved interpretation of information that is spread to multiple roles over distributed development sites.

Such support for decision rationale, however, puts new demands on tools and technologies. Current technology does not adequately support storage, maintenance and search for decision rationale. More research is needed to enable tracking and control of important decisions over time and space.

(3) Roles and Responsibilities
Product management and requirements engineering in general, and in relation to product lines in particular, span multiple roles and areas of responsibility, from management, marketing, and technical management to architecture and development. Technologies need to take this into consideration by tailoring the presentation of the same data, such as requirements and other information, to these roles and responsibilities. The gap between these views needs to be closed using intelligent technologies and adequate tool support.

Today, most technologies and tools support one view, and several tools need to be used to cover a larger spectrum of roles. This not only hinders the effective use of these technologies and tools, but leads to the need for establishing high-cost and complex traceability policies, which are hard to enforce.

(4) Tools
The acquisition of tools in industry is seldom performed based on proper investigation of actual needs and available options. Instead, a number of factors separated from tool use influence the tool selection and acquisition process. This results in installed tools that, while improving some practices, require the adoption of practices that are at odds with a company's true needs.

The workshop participants concluded that for good tool selection it is best to start with studying people, the problems facing the company, and the needed process, before choosing a tool that appropriately supports these needs. To enable such a selection process, researchers have a responsibility not only to develop technologies and tools that fit into one specific situation, but to design them to be adaptable to different practices and processes. Such a goal, however, needs to be balanced with the fact that there may not be a one-size-fits-all.

Some of the issues and questions raised during the tool discussion revolved around the possibility and potential of providing tailored views of one body of data for different roles and decisions. An interesting question raised was whether it is possible to "toolify" domain knowledge as a way to promote a shared understanding over different development sites.

Globally Distributed Communication
This session addressed requirements engineering and release planning in a distributed organization and the effects of factors like distance, communication mode and organizational structure on activities like negotiations and maintaining awareness of requirements.

The discussion topics included:
1. the role of formal processes and leadership,
2. shift in resource needs,
3. balancing the formality of information,
4. the role of motivation for enabling distributed communication, and
5. how trust can be achieved and maintained.

(1) Processes and Leadership
It was the experience and conclusion of the discussion participants that a development organization with a process culture that relies on informal ways of working and communicating can be successful in centralized development situations. Faced with distribution, this process culture is likely to lead to failure. However, when transiting from centralized to distributed development, the problems will be realized and may encourage process improvement.

Distribution also puts high demands on the leadership of managers, in particular project management. Similar to the process observations, collocated development is an environment where leadership based on informal procedures and communication can work. However, this is not possible to the same extent in distributed environments.

The cultural differences between development sites were put forward as a critical factor to be managed. Experiences from industry showed that leadership competence was crucial in alleviating these differences. For example, the ability to catch problems early and give instant feedback to practitioners distributed over different sites was seen as a success factor. If this was not managed, small problems, if not caught early, could evolve to showstoppers.

Another aspect, which can make or break success of distributed development, was predictability and repeatability of communication and problem resolution. This aspect should be addressed by combining process and leadership measures. For example, process, roles, and responsibilities should be made clear: every practitioner should know who should resolve an issue, how it should be resolved and the expected timeline for the solution.

Finally, staff turnover was mentioned as a factor that complicates management and leadership. Staff turnover generally increases in distributed environments. This can threaten development initiatives, as valuable competences are lost mid-project. Organizations with ad hoc processes that rely on the competence and heroics of individuals are particularly vulnerable. This problem can only be

addressed by managing knowledge and providing leadership through well defined and working processes, moving away from dependencies on a handful of key individuals.

(2) Resource Allocation
Distributed development requires not only more managerial overhead than collocated development, but also increased resources for fundamental tasks such as product management and requirements engineering. For example, low-quality requirements specification is not necessarily a serious problem in a collocated environment, as tacit knowledge can readily be exchanged: proximity, informal communication, and shared domain knowledge can compensate for insufficiencies. This is not the case in distributed environments, and this fact leads to a noticeable increase of resource needs.

There was agreement that distribution comes not only with potential savings, but also with added costs and modified distribution of resource needs. For example, moving programming to a region with high competence but lower salary levels can decrease resource needs for the development part of a project. This demands, however, that parts of these "savings" are allocated to management and requirements engineering activities. Business as usual can derail perfectly viable projects and increase overall costs.

(3) Modeling and Formalization
The use of modeling and formalization of information like requirements needs to be kept on a realistic level. Translating all knowledge into formal or semi-formal models is unrealistic as resources will not be available. Thus, researchers need to address the issue of distributed development in two ways. First, modeling needs to be brought closer to reality by allowing the capture of imperfect information. Second, acknowledging that not everything will be formalized, there is a need for mechanisms to identify critical parts that should be formalized. The boundary between informal and formal parts should be seamless.

Another important research topic is to demonstrate the benefits of formalization and modeling using empirical data from large-scale distributed development. The use of "toy" examples is common practice in research, but not relevant, rather there is a need to understand how technologies can be applied in real live circumstances. Also there is a genuine need for research into when and under what circumstances formalization and modeling are beneficial, and, even more important, when not. It is hard to find such knowledge, also in literature beyond distributed development.

(4) Motivating Distributed Communication
As the need for documentation and process formalization increases from collocated to distributed development, so does the need for communication. Communication is important for making all practitioners aware of the processes to be followed and to exchange knowledge and work results. To reflect the achievements of milestones, documents, previously seen as unnecessary, need to be created and maintained. To improve work efficacy, a constant flux of information needs to be managed that enables practitioners to stay up to date with the evolving work.

In addition to process transparency, this issue concerns the problem of motivation. Reward systems, which reward initiatives and good work that are aimed at making the distributed environment work, may play an important role. Management should assure that coordination delays are minimized and that engineers can stay productive. The distribution of multiple parallel tasks to developers assures that a delay in one task can be alleviated by continuing work on a second task.

Management should also assure that the necessity of creating documentation is understood. Such work that enables distributed development must be considered as important as other activities including actual development. Documentation should not be seen as pure overhead, but rather a necessity that originates from the decision of distributing the development in the first place.

(5) Trust
Trust and working communication channels between development and customers was seen a crucial success factor of distributed development. This applies for both kinds of customers, company-internal customers such as a product manager and customers external to the development organization.

The use of scenario-based communication utilizing enriched media was mentioned as a promising communication method that had been studied. Still open issues include situational awareness, the difficulty of grasping the current state of development in terms of information and expectations, which in turn can damage trust between customer and development.

Challenges of Global Requirements Engineering: Consequences for Research
This session discussed challenges of distributed requirements engineering elicited from IT professionals. The aim was to build a basis for shaping future research in the global requirements engineering area.

The discussion topics included:
1. the definition of the research area and
2. future research initiatives.

(1) Shared Understanding of the Area
The discussions during the workshop had on occasions shown misunderstandings and even disagreements about distributed development. The terms *distributed, multi-site*, and *global* were used and interpreted differently by different researchers and industry representatives. For example, is outsourcing the same as distributed development? Many argue that development is outsourced to another company at the same location cannot be considered distributed development. In the opposite example, can a situation with a company-internal customer that is geographically separated from development, where the customer is seen as an integral part of the requirements engineering process, be considered distributed development? Many of the challenges can be relevant for both situations, for example cultural differences and domain background, while other challenges, like different time zones, are particular to one constellation. One important step in formulating a research agenda for the area is to agree on a vocabulary, in order to make it possible for research to properly describe and communicate the context in which studies and research are performed.

In addition, a need to analyze and investigate the uniqueness of the challenges posed by distributed development was identified. Many of the challenges discussed in the context of distributed development were applicable in most non-distributed development settings. For not "reinventing the wheel", the study of currently

available solutions to some of the identified problems was considered a prerequisite for future research. The identification of challenges that are unique to distributed development was seen as a natural evolution and the next step.

(2) Future Research Initiatives
Future research should include empirical studies. Two main perspectives were identified. One, the study of distributed development efforts needs to be continued to further identify, understand, and describe their challenges. Two, new "solutions" need to be tested not only in a clinical setting using illustrative and potentially unrealistic examples, but piloted in industry to assure their scalability.

Summary and Conclusions

The first international global requirements engineering workshop (GREW'07) [8] provided a lively forum for sharing positions, experiences, challenges, research results, and ideas in the areas of requirements engineering and product management in a global context. The workshop was attended by experienced practitioners and young and senior researchers. Hence, it contributed to establishing a good understanding of current practices and problems across a number of companies.

The discussed challenges need to be addressed by industry and research together. In addition to exploiting emerging knowledge, a need for informing management was identified: tailoring management and facilitation to the globalized situation was seen as critical as methods, technologies, and tools.

GREW'07 provided the opportunity to present research results and to discuss the current state and the future of research in global requirements engineering and product management. Besides the specific research opportunities that are described in this report, general needs for the evolution of global requirements engineering were identified. First, the scope of the research field needs to be defined. This includes agreeing on terms for communicating research and separating issues that are aggravated due to distribution from issues that appear as a fundamental consequence of distribution. Second, empirical research that aims at enabling scalability of research results to real-world global situations needs to be increased.

It is the hope and desire of the workshop participants that the workshop results presented in this report will contribute to shaping the forthcoming research area for global requirements engineering and product management. The contacts that were established should further support collaboration between research and industry.

References

[1] Battin, R.D., Crocker, R., Kreidler, J., Subramanian, K.: Leveraging Resources in Global Software Development. In IEEE Software 18(2) (2001), pp. 70-77.
[2] Damian, D., Zowghi, D.: RE Challenges in Multi-Site Software Development Organisations. In Requirements Engineering 8(2003), pp. 149-160.
[3] Ebert, C., De Neve, P.: Surviving Global Software Development. IEEE Software 18(2001), pp. 62-69.
[4] Fricker, S., Gorschek, T., Myllperkiö, P. (2007): Handshaking between Software Projects and Stakeholders Using Implementation Proposals. In Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'07), Trondheim, Norway, 2007, pp. 144-159.
[5] Herbsleb, J.D., Mockus, A. (2003): An Empirical Study of Speed and Communication in Globally Distributed Software Development. In IEEE Transactions on Software Engineering 29 (2003), pp. 481-494.
[6] Herbsleb, J.D., and Moitra, D.: Global Software Development. IEEE Software 18(2001), pp. 16-20.
[7] Herbsleb, J.D., Paulish, D., Bass, M.: Global Software Development at Siemens: Experience from Nine Projects. In Proceedings of the 27th International Conference on Software Engineering. ACM, St. Louis MO, 2005, pp. 524-533.
[8] International Global Requirements Engineering Workshop (GREW) homepage: http://www.bth.se/grew07

Workshop Participants

Alberto	Avritzer	Siemens, SCR	USA
Fabio	Calefato	University of Bari	Italy
Oliver	Creighton	Siemens, CT SE	Germany
Daniela	Damian	University of VIctoria	Canada
Robert	Feldt	Blekinge Inst of Tech	Sweden
Samuel	Fricker	ABB Switzerland and University of Zurich	Switzerland
Alex	Gansallo	Ebendick Ent	South Africa
Tony	Gorschek	Blekinge Institute of Technology	Sweden
Dorina	Gumm	University of Hamburg	Germany
Gerald	Heller	Hewlett Packard GmbH	Germany
Andrea	Herrmann	University of Heidelberg, Institut für Informatik	Germany
Korbinian	Herrmann	Technische Universität München	Germany
Christiane	Kuntz-Mayr	SAP AG	Germany
Teresa	Mallardo	University of Bari	Italy
Sabrina	Marczak	University of Victoria	Canada
Audris	Mockus	Avaya Labs Research	USA
Dieter	Nungesser	Siemens, SBT	Germany
Horst	Pichler	Siemens SIS PSE	USA
Richard	Torkar	Blekinge Institute of Technology	Sweden
Claes	Wohlin	Blekinge Institute of Technology	Sweden