Requirements Engineering in Global Software Development -

Challenges and Opportunities

**by Ian Bui**

**Systems Engineering & Management**

**University of Texas in Dallas**

**Spring 2012**

**ABSTRACT**

For the past several decades software development has become increasingly globalized with customers, management and engineering teams spread across multiple time zones and geographic locations. As a result Requirements Engineering - a critical component in the development lifecycle, is facing unprecedented challenges and difficulties. This paper discusses those issues and offers some suggestions on how they can be mitigated.

The problem that Requirements Engineering in a Global Software Development (GSD) environment faces today typically has four components: Communication; Knowledge Management; Culture; Time. Depending on the organizational structure and the nature of the project, each of these components can have varying degrees of negative impact on the outcome of the project.

Current Requirements Engineering methodologies are sufficiently well-defined and do a fairly good job of providing RE practitioners with a framework with which to carry out most RE tasks in a consistent manner. However, no new solution or model has yet emerged to address the challenges of GSD. Our view is that the software industry, standard bodies and academia need to come up with practical modifications or augmentations to the existing models (and the tools to support them) that take into account GSD-related issues in order to provide a more reliable and consistent process.

To that end, this paper examines two Case Studies that describe the main issues and highlight areas where we believe future research might be focused. Specifically, we believe opportunities for process advancements exist in the following areas: Tools, Standards and Education.

**INTRODUCTION**

The Software Industry has grown significantly in the past few decades and has gone through some very important phases of maturation. The introduction of object-oriented programming and tools such as UML, for example, have transformed the traditional programming paradigms and methods. New processes such as Iterative Development and Agile have also contributed to a more robust and rapid development cycle. Not surprisingly, Requirements Engineering has also grown into a scientific discipline in its own right, with accompanying theories, methodologies and models that software companies have relied on when developing their products. Many universities now offer Requirements Engineering as part of their Computer Science curriculum, both in the undergraduate and graduate programs.

With the advent of the Internet and reduced costs of global communication networks, the Software Industry has found itself in yet a new transformation – moving from the traditional co-located development paradigm to a globally distributed environment across multiple continents and time zones. The software market is now global; customers are spread all over the world and companies have to compete globally for engineering resources who can do the job for the least costs.

For practitioners of Requirements Engineering, the growth of Global Software Development (GSD) presents some unprecedented challenges. The most important problems RE in GSD faces can be summed up as follows:

1. *Inadequate Communication*: Distance creates barrier to communication. So does language and lack of consistent communication infrastructure or tools.
2. *Knowledge Management*: Gathering information from many stakeholders spread around the world is not easy; managing changing requirements in a global environment is even harder.
3. *Cultural Diversity*: Even when teams speak the same language, cultural differences across organizational units and sites contribute to misunderstanding and mistrust.
4. *Time Difference*: Reliance on asynchronous communication due to time differences is a major cause for delays, increased costs and inefficiency.

Even though Requirements Engineering is a fairly well understood science and many of its methodologies and processes have been “formalized”, it is still a relatively new and evolving discipline. It is encouraging to note that there exist many software tools to support RE activities, best known and oldest among them is RequisitePro from Rational. It is expected that with the increased adoption of Agile software development methods, RE activities will be more tightly integrated within existing Agile tools.

However, tools alone will not solve the complex problems that GSD present to Business Analysts tasked with managing always changing system requirements. What’s needed is a new framework based around the people behind the processes, relying on real-world issues experienced by stakeholders in GSD projects. It is our view that solution will require a cooperative effort from academia, standard bodies, and the software industry itself.

In the first Case Study we will examine the issues from the perspective of a real software company doing GSD. The second Case Study examines the problem of Stakeholders Management and proposes some mitigation techniques. Combined, we hope these case studies can provide the reader with a satisfactory overview of the issues and point to specific areas where more indepth research is needed.

**CASE STUDY I**

The first case study was done on a Global Software Development company whose name was withheld for confidentiality. It is headquartered in the US, its development team is located in Australia, and its customers are spread across several continents. Research was carried out to identify major issues that confronted the development teams in eliciting, negotiating and validating requirements for a real project. Research methods included face-to-face interviews of stakeholders as well as in the form of open-ended questionnaires sent out to different team members involved. Some examples of the questions are:

* *‘What challenges do you face in managing requirements in a distributed setting?’*
* *‘Which one is the most significant and requires urgent improvement?’*
* *‘Which technologies are used to overcome distance?’*

The research results identified four major categories of issues facing GSD projects:

1. **Inadequate Communication**

It should come as no surprise that distance creates barriers to face-to-face and informal discussions, which are crucial to understanding. What more, the quality of conferencing tools were found to be not consistent from site to site. For example, Internet connectivity in some countries may be slow or unreliable during certain parts of the day, thereby making video conferencing awkward and ineffective. Most surprisingly, some stakeholders tried to exploit these communication problems to advance their own agenda, even at the cost of the success of the overall project.

1. **Knowledge Management**

Because stakeholders are spread around the globe, requirements must be gathered from multiple sources that span multiple geographical areas, time zones, cultures and languages. Managing such a diverse source of information was a big challenge for the Requirements Engineers, especially as requirements began to change once the project got under way. It was also found that not all sources of information are adequately shared with developers. Sometimes information was even deliberately withheld from developers for business, political or personal reasons.

1. **Cultural Diversity**

There is no question that language and customs can seriously impede the ability to communicate and collaborate in GSD projects. The research found that even when teams do speak the same language, differences in organizational cultures also contributed to conflicts, mistrust and misunderstandings. Furthermore, some remote sites had developed their own methodologies that were not consistent with the rest of the Project Team, causing confusion, inefficiencies and errors.

1. **Time Differences**

Because of time differences, much of the communication had to take place asynchronously, e.g. emails, documents, etc. And even when synchronous communication such as teleconference was used, oftentimes it was difficult to arrange a time where all required team members could be available. As a result, it took longer to resolve requirements issues during the Negotiation, Prioritization, Validation phases.

Perhaps not surprisingly, these four categories of issues appear to be consistent in many GSD organizations, as evidenced by the current research literature available on the subject [6]. The study also identified the following major challenges that all GSD organizations face to one degree or another:

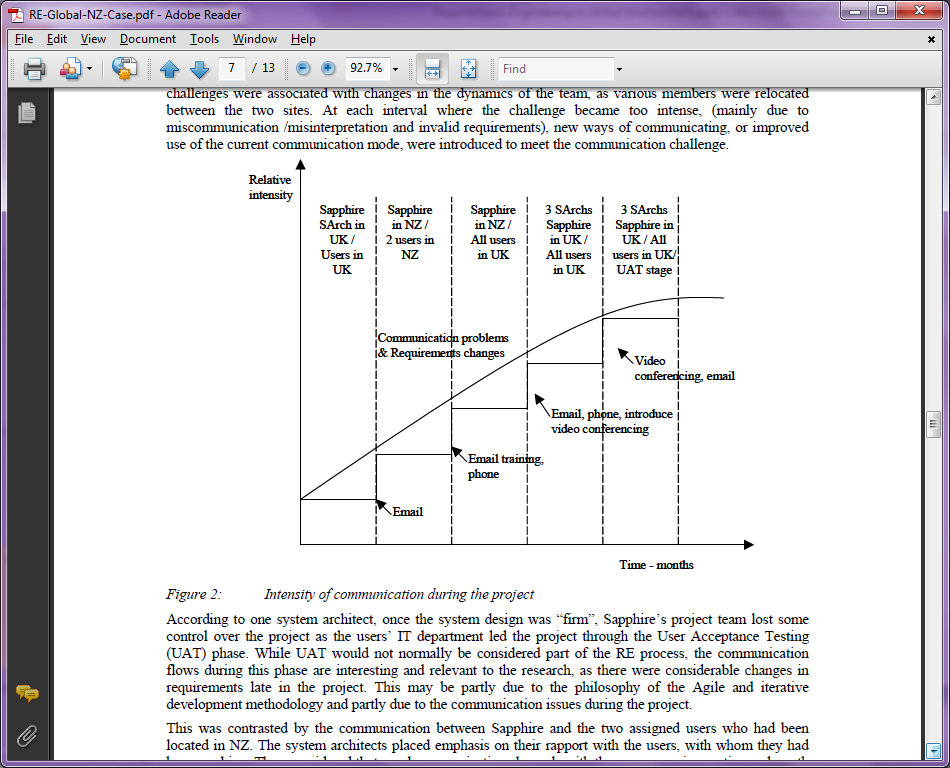
1. ***Geography*:** Because sources of information are spread out, field investigations become more expensive and difficult.
2. ***Communication*:** Requirements Engineering in practice relies on the use of natural language, in this case the language barrier plays a particularly important role in elicitation, negotiation and validation of system requirements.
3. ***Politics*:** There will always be power struggles among various stakeholders, especially when they are separated by distance, culture and language. The farther apart the teams are, the higher the risk of distrust.
4. ***Trust*:** Information sharing is crucial in Requirements Engineering. Yet it is often the case that certain stakeholders may choose to withhold information from the development team for reasons that may or may not be justifiable. For example, a client may choose to not disclose certain business data that might help developers understand the business problem better. Distance also makes it harder for remote teams to know and trust each other. Of all the challenges a GSD entity must overcome, trust is perhaps the most critical and least appreciated.

The study concludes by suggesting that a new RE model is needed that can:

1. Assess the impact of distance on Requirements Engineering
2. Quantify the extent to which distance affects collaborative activities
3. Identify the relationship between distance and the RE issues listed above

In a different study [2] the effects of distance can be represented graphically as shown in this example below:

**Fig.1 Effects of Distance on Communication Intensity**

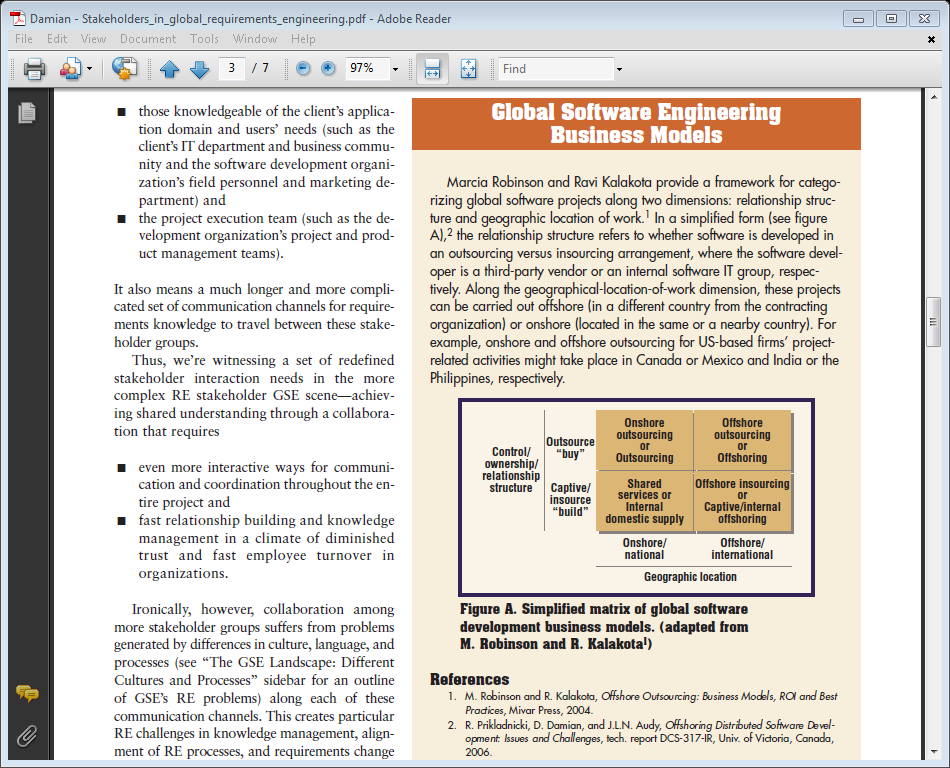


Sapphire Software is a company based in New Zealand that has developers and clients spread across several continents. As the graph indicates, the intensity of communication rises as the project progresses AND as work is spread across more sites. Whereas simple asynchronous communication was only required early on, more direct and synchronous communication would be needed during later phases of the project. Note also that as time went on changes to requirements increased as well; this also explains why more direct communication was needed to manage the changing scope and to keep the project on schedule.

**CASE STUDY II**

The second Case Study in this paper examines the challenges of managing Stakeholders requirements and expectations in a Global Software Development setting. In it the author also suggested some strategies to effectively deal with those challenges. [3]

Part of the motivation for the paper comes from the recognition that the globalized economy has introduced a new software business model, a simplification of which is shown below.



**Fig. 2 - Simplified matrix of global software development business models**

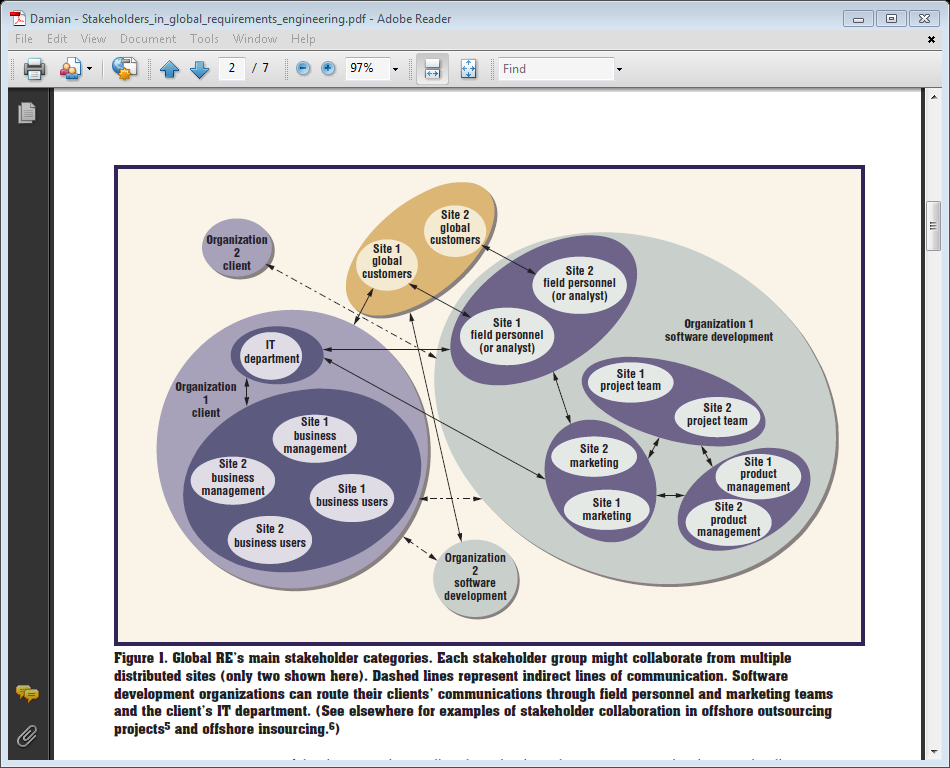
**(adapted from M. Robinson and R. Kalakota)**

*“In this framework, software projects can be categorized along two dimensions: Relationship Structure and Geographic Location. The relationship structure refers to whether software is developed in an outsourcing vs insourcing arrangement, where the software developer is a third-party vendor or an internal IT group, respectively.*

*Along the geographical-location-of-work dimension, these projects can be carried out offshore (in a different country from the contracting organization) or onshore (located in the same or a nearby country). Forexample, onshore and offshore outsourcing for US-based firms’ project related activities might take place in Canada or Mexico and India or the Philippines, respectively.”[3]*

Given this new business model, stakeholders in today’s GSD environments are typically distributed over wide geographical areas, which makes it difficult for them to interact with each other. Furthermore, their different backgrounds (ethnic, cultural, language, etc.) also increase the complexities involved in gathering information about their needs and achieve a shared understanding of system requirements.

**Fig. 3 - A new model of Stakeholders and their interactions in a GSD environment**



*“In this model, each Stakeholder group might collaborate from multiple sites (only two shown here). Dashed lines represent indirect line of communication. Development teams can route their client’s communication through Field Personnels, Marketing teams, or client’s IT Department.”[3]*

Note the addition of new Stakeholder groups such as Field Personnel and Analyst Teams located near the client’s site. These are the people who directly interact with the client’s IT department and business users during the Elicitation and Validation phases of RE. Note also, that this model allows for Project Teams, Product Management and Marketing Teams from different sites to collaborate and coordinate their activities in managing Requirements and changes.

It goes without saying that as the distance between clients and developers increase, the gap between those with knowledge of the user domains and the execution team also increases. At the same time, the communication channels between the stakeholders become more complex, and delays become more common.

To combat these effects, the organization must come up with more innovative ways to communicate interactively with their clients and to coordinate the disparate teams’ activities. It must also learn to quickly build trusting relationships among its stakeholders and to somehow cope with a modern business environment where employee turnover is high, not only in its own organizations but also in its client’s as well.

The author identifies three processes that can help GSD stakeholders better achieve shared understandings. They are:

1. **Knowledge Acquisition and Sharing**

This process is of critical importance in fostering understanding. Because of the many layers of stakeholders involved in the GSD model, system designers have fewer opportunities to learn about user needs firsthand.

Furthermore, as already pointed out in the first Case Study, some stakeholders may choose to withhold information from others (e.g. priorities, standards, policies, etc.) Conversely, system architects and designers may (inadvertently or otherwise) fail to provide accurate estimates of level of efforts and/or work priorities.

These issues not only lead to diminishing trust, they are often compounded by the turnover problem all modern organizations must face today, consequently impacting the lines of communication created by the new GSD model.

1. **Aligning RE Processes and Tools**

Not all organizations use the same processes and tools when it comes to Requirements Engineering. Nor do all of them possess the same level of maturity in terms of implementing or applying those processes. Thus, when these organizations become interconnected in a GSD project problems inevitable emerge that cause delays, overhead, rework and cost overruns.

It is therefore vital to ensure that GSD organizations align their RE Processes and tools in such a way that the flow of information is enhanced, not impeded. GSD entities must also take special care to avoid misunderstanding due to different business procedures and terminologies.

1. **Effective communication and coordination**

Relationship building is often hampered by a lack of *informal* communication, and in a global team setting this is often the case. Successful requirements negotiation, on the other hand, depends on a certain level of trust which often is based on relationships. Informal communication, as opposed to formal communication, can oftentimes provide stakeholders with the proper context to understand certain requests, especially if they are motivated by political or personal agendas.

Moreover, time differences between sites also reduce the opportunity for informal conversations. This can cause delays in identifying requirements issues and their resolutions, which in turn can lead to more mistrust among stakeholders. Finally, requirement changes need to be coordinated and resolved in a timely manner through effective communication channels and processes.

**Coping Strategies**

The study’s author suggests two complementary sets of strategies to deal with the complexities of Requirements Engineering in GSD projects: Supporting the Interorganizational Structures, and Supporting the Communication Practices.

1. **Supporting Interorganizational Structures**
2. *Define interorganizational structure with clear communication responsibilities*

Within the distributed project, stakeholder roles must be clearly defined with specific responsibilities. New roles can be created as the need arises.

More importantly, lines of communication must be clearly established indicating which role must communicate with each other between various stakeholder groups.

1. *Establish peer-to-peer links at all management, project, and team levels*

There must be a established link of communication across all of the distributed sites that connect all the stakeholder groups at the peer-to-peer level. This will help develop better client-developer relationship and will facilitate the RE elicitation and validation activities. It will also help manage requirement changes.

1. *Synchronize interorganizational processes and perform frequent iterations and deliveries*

One of the first things stakeholder groups must do is to agree on a common “vocabulary” and a set of “work products” or “deliverables”. Templates for common artifacts such as Requirements Specifications, Use Cases, Test Scenarios, and Functional Designs also need to be developed.

It is also important to have frequent reviews and validations of these artifacts. This iterative process allows the distributed teams to have visibility into work progress done at remote sites and to raise any concerns in a timely manner. It also helps tremendously in managing stakeholders’ expectations.

1. *Establish cultural liaisons*

To help bridge any culturap gaps and build trusting relationship, it is crucial to have some individuals who can act as a cultural liason at remote sites. These individuals play a pivotal role during the Elicitation, Negotiation and Validation phases of Requirements Engineering. They can be relied upon to translate tacit knowledge into functional requirements and to have a better grasp of working and cultural characteristics at remote sites.

1. **Supporting Communication Practices**
2. *Maintain open communication lines between well-defined stakeholder roles*.

Throughout the project, representatives from all stakeholders group need to have regular meetings to review previously defined project targets, artifacts, work products, etc., and approve any necessary changes. Stakeholders must also be able to map these common work products to business requirements and any risk-management plan they may have.

1. *Frequently inform and monitor progress on commonly defined artifacts*.

It is best (and less costly) to create small teams which can meet frequently (e.g. weekly) and synchronously (e.g. video conferencing) to speed up the process of Requirements Management activities. The use of mdern communication and collaboration tools should be maximized to improve the efficiency and effectiveness of these meetings.

**Collaborative Tools for RE in GSD Settings**

Finally, the author provides a useful list of tools that organizations might want to consider using to facilitate their RE-related activities. These tools have been tested in a GSD environment and found to be “useful in empirical studies in industry and academic settings.” [3]

**EGRET**: Eclipse-based global requirement tool for distributed requirements management. Supports Change Management, Knowledge Management, Awareness, and Informal Collaboration.

**IBIS**: Internet-Based Inspection System. Support asynchronous communication of stakeholder teams in inspecting requirements specifications. Support development of shared understanding between remote clients and developers. Provide synchronous distributed requirements negotiations.

**Björn Regnell**: Method for visualizing agreement in distributed prioritization of Market Requirements; allows stakeholders from distributed market segments to participate in an iterative process of requirements prioritization.

**DCPT**: Distributed Collaboration and Prioritization Tool, using WinWin Requirements and Negotiation system; has been used successfully in large stakeholder teams.

**Rational RequisitePro**: A commercial Requirements Management tool from Rational (now owned by IBM)

**DOORS**: Another popular commercial Requirement Management tool from Telelogic.

**CONCLUSIONS**

Software Development lifecycle and processes are well understood. But the rise of Global Software Devlopment has presented new challenges, and the industry has generally been adapting its practices to cope with them. However, Requirements Engineering, which plays a crucial role in this lifecycle has lagged behind in this evolution. While researches on the impact of GSD on RE methodologies have started to appear, it is not yet clear if any new model will emerge as the winner to be widely adopted.

Meanwhile, software companies doing business globally must continue to learn through trial and errors what the best methodologies would be for their particular business. In this regard, we believe several opportunities exist that can propel the art and science of Requirements Engineering to the next level.

The first of these opportunities lies with the Software Tools industry. Besides traditional Requirements Management tools and Change Management tools, collaboration and communication tools will have an indispensable role in any modern GSD organizations, to help them better coordinate their RE activities across multiple sites and stakeholder groups. A comprehensive tool solution where stakeholders can easily model their new business processes and requirements would be extremely helpful.

Secondly, standard bodies such as IEEE should play an active role in developing new RE models and artifacts that will take into account many of the issues specific to GSD environments as described here and elsewhere.

Finally, academia also has an important role to play, both as an incubator for research as well as a training ground for future Requirements Engineers and Business Analysts working on GSD projects. For example, Computer Science departments across the world could institute programs geared specifically toward RE for GSD, where students from distant universities spanning multiple locations and time zones can learn to work together on a project. For those interested in this subject, we highly recommend one such experiment that was done at the University of Victoria (Canada), University of Technology (Sydney, Australia), and University of Bari (Italy) [4].

**REFERENCES**

[1] THE STUDY OF REQUIREMENTS ENGINEERING IN GLOBAL SOFTWARE DEVELOPMENT: AS CHALLENGING AS IMPORTANT - Daniela E. Damian - University of Technology, Sydney

[2] REQUIREMENTS ENGINEERING DURING GLOBAL SOFTWARE DEVELOPMENT: SOME IMPEDIMENTS TO THE REQUIREMENTS ENGINEERING PROCESS – A CASE STUDY. Jo Hanisch - School of Accounting and Information Systems, University of South Australia, Brian Corbitt - Deakin University

[3] STAKEHOLDERS IN GLOBAL REQUIREMENTS ENGINEERING: LESSONS LEARNED FROM PRACTICE. Daniela E. Damian, University of Victoria.

[4] TEACHING REQUIREMENTS ENGINEERING IN GLOBAL SOFTWARE DEVELOPMENT: A REPORT ON A THREE-UNIVERSITY COLLABORATION - Daniela Damian, Ban Al-Ani, Davor Cubranic, Lizveth Robles. University of Victoria (Canada), University of Technology (Sydney)

**OTHER READINGS**

[5] REQUIREMENTS CHANGE MANAGEMENT IN GLOBAL SOFTWARE DEVELOPMENT: A CASE STUDY IN PAKISTAN - Waqar Hussain, Linnaeus University

[6] THE 3RD INTERNATIONAL WORKSHOP ON GLOBAL SOFTWARE DEVELOPMENT - International Conference on Software Engineering Edinburgh, Scotland, May 24, 2004

[7] WIDE AUDIENCE REQUIREMENTS ENGINEERING (WARE): A PRACTICAL METHOD AND CASE STUDY - Tuure Tuunanen (University of Auckland, New Zealand), Ken Peffers (University of Nevada, USA), Charles Gengler (City University of New York, USA)