The Impact of Neglecting Domain-Specific Security and Privacy Requirements [*]

John Wilander
Omegapoint AB, and
Dept. of Computer and Information Science
Linköpings universitet

Jens Gustavsson
Dept. of Computer and Information Science
Linköpings universitet

E-mail: {johwi, jengu}@ida.liu.se

Abstract

In a previous field study of eleven software projects including e-business, health care and military applications we documented current practice in security requirements. The overall conclusion of the study was that security requirements are poorly and inconsistently specified. However, two important questions remained open; what were the reasons for the inconsistencies, and what was the impact of such poor security requirements? In this paper we seek the answers by performing in-depth interviews with three of the customers from the previous study. The interviews show that mature producers of software (in this case IBM, Cap Gemini, and WM-Data) compensate for poor requirements in areas within their expertise, namely software engineering. But in the case of security and privacy requirements specific to the customer domain, such compensation is not found. In all three cases this has led to security and/or privacy flaws in the systems. Our conclusion is that special focus needs to be put on domain-specific security and privacy needs when eliciting customer requirements.

Keywords: security and privacy requirements, requirements engineering

1 Introduction

The security (confidentiality, integrity, availability) and privacy properties of custom-made software relies heavily on the requirements specified by the customers. If the requirements are poorly specified there is no guarantee that the producers of the software will strive for security.

Security and privacy (henceforth grouped as *security* where possible) are often conceived as *non-functional requirements* [5, 6, 7, 9], which are generally hard to manage. But our previous field study on requirements on eleven systems showed that more than 75 % of security requirements found in specifications are in fact functional [23]. We concluded that customers are generally better at specifying functional requirements including functional parts of security. Therefore the hard part of specifying security lies in the truly non-functional aspects such as security processes, security testing, and security evaluation. Section 3 in this paper briefly presents the results of the previous field study.

The outcome of the field study led us to a few hypotheses as to why security requirements are poorly specified, and what the impact of such poor requirements would be. We present these hypotheses in Section 4. To verify the hypotheses we conducted in-depth interviews with customer project leaders from three of the systems in the previous study. Section 5 summarizes the outcome of these interviews. The answers confirmed most of our hypotheses and the discussion can be found in Section 6. In Section 8 we draw the conclusion that customers and producers of software should put special focus on domain-specific privacy

[*] This work has been supported by the national computer graduate school in computer science (CUGS) commissioned by the Swedish government and the board of education.

needs when eliciting requirements for security critical systems.

2 Terminology

Software systems and stakeholders described in this paper include *customers* that have needs, typically specified as software requirements. The customers buy systems from software *producers* who fulfill requirements and deliver systems. In our frame of reference, customers are not necessarily IT-experts, but rather experts within their own domains such as health care processes or traffic and infrastructure. A few terms we use need to be defined:

Requirement. A requirement is a specification of what the customer needs to be implemented during system development—a description of how the system should behave, or of a system property or attribute [22].

Unspecified need. An unspecified need is a left-out requirement—the customer needs a certain function or system behavior but has so far failed to express this need as a requirement.

Over-delivery. An over-delivery is performed when a producer fulfills more than the customer has explicitly required, typically when the producer fulfills the customer's *unspecified needs*.

Local hero. A local hero is a person with expert knowledge in a subset of a field that is wrongly consulted as an expert in the field in general. An example could be a person with much experience in how to set up and and effectively manage security logging. By others that person could very well be consulted as an expert in system security in general, being the "local security hero".

3 Previous Study

In 2005 we published a field study of current practice covering eleven requirements specifications on IT systems being built 2003–2005 [23]. All specifications were made for public procurement in Sweden, in compliance with EU procurement directives. This choice was made primarily because of the public availability of the specifications. The study covered:

- Five systems for billing, accounting, salary, and e-business
- Three health care systems
- One system for defense materiel
- One system for reporting hazardous materials
- One system for managing highway tolls

Requirements found in the specifications were categorized into security areas and divided into functional and non-functional requirements (for details on how this was done see the original paper [23]). Table 1 contains an overview of all security requirements we found. Note that requirements not present in any of the studied specifications are not in the table, thus no rows with zero requirements. The overall conclusion was that security requirements were poorly specified due to three things: inconsistency in the selection of requirements, inconsistency in level of detail, and almost no requirements on standard security solutions.

4 Hypotheses

The outcome of the field study lead us to four general hypotheses about the delivered systems. These were our hypotheses:

4.1 Security Requirements Incomplete

In our previous study we did not have access to any risk analysis documents, nor did we speak with the people involved—we just studied the requirements specifications as such. Therefore we could not know if certain security requirements had been left out because of deliberate decisions or because of lack of information or knowledge. As a consequence we did not judge the requirements specifications as complete or incomplete, but rather analyzed consistency and the use of standards. However, our hypothesis was that the security requirements were indeed incomplete or underspecified.

4.2 Lack of Risk Analysis

In several of the specifications studied we noted that some security requirements were fairly well specified

Requirements **Systems**	Billing	Accounting	Salary/Staff 1	Salary/Staff 2	E-Business	Defense Materiel	Medical Advice	Health Care 1	Health Care 2	Highway Tolls	Hazmat
Access Control/Roles	1	11	6	5	8	5	4	5	3		3
Attack Detection							2	4		3	
Backup		5	9	2			2	2			
Digital Signatures			1		1	1	1	2			1
Encryption							4	1			1
Integration							2	1			
Logging		9	3	1	11	1	5	8	1		
Login		5	3	3	8	2		2	1		2
Privacy			2							1	
Authentication							2	4	2		1
Availability	1		3			1	6	4		3	1
Design/Implementation					1			6			1
Physical Security								6			
Risk Analysis										1	
Security Management								2		2	
Security Testing								1			

Table 1. Overview of previous study—security requirements on eleven IT systems built 2003-2005. The double horizontal line divides the requirement categories into mostly functional (above) and mostly non-functional (below). Numbers tell how many requirements were found in each category.

whereas related ones were completely left out. Examples of such inconsistencies could be seen in access control/roles where all systems had requirements indicating that restricted access was important. At the same time only three out of eleven specifications required some kind of encryption of data communication and only two specifications had requirements on restricted physical access.

Our hypothesis was that the specifications had not been preceded by risk analyzes. Such analyzes should have identified a greater variety of threats against important assets, and thus resulted in more consistent requirements.

4.3 Heavy Trust in Local Heroes

Some security requirements had a high level of detail whereas others in the same specification were only specified on a general level. This might indicate that the organizations specifying the security requirements relied heavily on local competence and not standards. Such local competence tends to be strong in certain areas and weak in others. We call this the *local heroes phenomenon*.

Such inconsistent levels of detail could for instance be seen in the "E-Business" system where the requirements on logging were very detailed (eight requirements on what info to be logged) and at the same time digital signatures were specified as "The system should be able to handle the use of electronic signatures" with no further details.

Our hypothesis was that the studied organizations had relied heavily on local heroes.

4.4 Systems Insecure

Considering the three previous hypotheses we arrived at an overall hypothesis that the delivered systems were insecure, and that this had manifested itself as security flaws and insecure operation.

5 Interviews

To verify our hypotheses we conducted interviews with the customers behind three of the requirements specifications, namely Health Care 1, Highway Tolls, and Medical Advice. Before we present the actual interviews we describe our methodology, scope, and potential shortcomings.

5.1 Methodology and Scope

We conducted oral, open-question interviews with the customer project leaders. The interviews lasted 1-2 hours and were recorded and transcribed (approximately 30 pages of text per interview) to allow for an accurate, qualitative analysis. We chose to interview the project leaders since they had a good overview of the requirements process, of the systems and of relations with the producers.

The three systems were chosen specifically because they were all security and privacy critical, they were fairly large, and they represented three interesting categories—a standard system with configuration (Health Care 1), a combination of standard components and development (Highway Tolls), and one system completely built from scratch (Medical Advice). Further, these systems had some of the best security requirements in the previous study (in the case of the Highway Tolls system there were proper references to security standards) which hopefully would work as an upper bound on our analysis, i.e. the other systems were unlikely to show substantially better results when verified against our hypotheses.

The systems were built by WM-data (Swedish company with 9.000 employees, now part of LogicaCMG), IBM, and Cap Gemini. We have chosen not say which company delivered which system, and the customers asked us not to publish man hours or code size since such figures were considered business secrets.

5.2 Potential Shortcomings

We have based our studies on requirements specifications made for public procurement in Sweden—a choice made primarily because of the availability of them. Commercial entities tend to have little interest in making their requirements specifications available for research. This limited scope affects the validity of the study.

A potential problem was the project leaders' technical competence but apart from a few unanswered questions this was never an obstacle. The interview analysis is qualitative and thus subject to the authors' interpretation. All customer quotes presented are translated by the authors.

5.3 Systems

Brief presentations of the systems studied:

Health Care 1 (Customer: Stockholm County Council). Integration platform to support personal medical information following nearly two million patients between various health care organizations. This system is a standard system with only minor new development, and is maintained and run by the producer.

Highway Tolls (Customer: Swedish Road Administration). Equipment, software and services for handling environmental fees for all vehicles entering the city of Stockholm. This system is a combination of standard components and new development, and is maintained and run by the producer.

Medical Advice (Customer: The Federation of County Councils). System for managing medical advice by phone on a national level. It handles redirection of calls, queue management, work-flow management, medical documentation, and statistics. This system was built from scratch for the customer, and is maintained and run by the producer.

5.4 Interview Health Care 1 System

Outcome of the interview with the Health Care 1 project leaders:

Security a critical requirement. The customer considers security to be a critical factor in the system since it contains patient information. A general risk analysis was used to drive parts of the requirements elicitation process, but no specific focus was put on security risks according to what the customer remembers.

Security logs checked by producer. The logs are managed by the producer and any incidents are discussed on a monthly meeting. Since the producer owns the auditing process we asked the customer if they had confidence in the producer telling them of incidents—”We think so. That’s a question of conscience!” But what if the producer detects a vulnerability, patches it, but never investigates if the vulnerability was ever exploited? “That’s not unlikely. But we’re going to hire a security manager that will perform audits of security maintenance.”

Security management standard not implemented. The requirements specification referred to the ISO/IEC 17799 standard for security management [13] but the customer admits it has not been implemented yet—“Unfortunately I don’t think so.”

Vague requirement on separation. Regarding confidentiality the specification stated that “It should be possible to separate different types of information both logically and in terms of security.” When asked if they have a clear picture of what was meant the customer replied “No ... Surely, that requirement must have sparked a lot of questions. But I would imagine it refers to privacy categorization of patient information.”

Vague requirement on encryption. The health care system was required to “... have functions for protecting the information in the database, for instance through encryption.” The customer did not specify what kind of encryption or even that it has to be a standard cryptographic algorithm. So we asked if standard encryption was delivered—“Yes, I think so. They’ve been talking ... a lot of three letter abbreviations ... It’s ongoing. We had a lot of discussions regarding this and it became an issue of negotiation in the end.” Here the producer covered up for a poorly specified requirement.

Local heroes phenomenon confirmed. Contrary to the vague specification on encryption the customer had requirements on input and output validation which must be considered being on a low technical level. When asked if this was a manifestation of the local heroes phenomenon they replied “Yes, I think so. We know their names too.”

Automatic recovery requirement not fulfilled. The specification contained a requirement on automatic recovery—“The system should automatically handle errors and restart functions and processes.” The customer admitted that the requirement was vague, and utterly impossible to fulfill if *all* errors were to be handled automatically. But an incident with a faulty load balancer had revealed that the requirement was not even fulfilled to a basic level. The system had gone down and not restarted.

Need for specific handling of protected identities unfulfilled. After delivery the customer had realized that the system lacked support for handling personal information for patients with protected identities. They considered this a severe flaw. The privacy of such patients introduces a whole new dimension to access control and to date it is not clear if and how such functionality can be introduced.

Summary. The customer feels that the producer of the health care system is mature and has fulfilled most of the fairly well specified security requirements. Vague requirements have been costly both in terms of money and time. The lack of support for handling personal information for patients with protected identities is a clear case of a domain-specific need not specified as a requirement by the customer and not fulfilled by the producer.

5.5 Interview Highway Tolls System

Outcome of the interview with the Highway Tolls project leader:

Security requirements very high. The customer considers the “security requirements very high” for the highway toll system, and thus a risk analysis has been performed both before initial release and during the current development iteration.

Logging features missing. The system logs are continuously checked but “there are deficiencies”. The deficiency turned out to be a disability to log what information is *accessed* in the system—only *modifications* are logged. This means that employees could violate both confidentiality (e.g. checking when and where cash transports leave and enter the city) and privacy (e.g. systematically checking people’s movements in the city area) without being noticed.

The customer considers this a serious flaw due to an incomplete requirements specification and risk analysis. It was clearly a domain-specific need and if the customer had the chance to re-write the requirements they say they would have hired third-party experts to identify such needs and specify them as requirements.

Security incident despite penetration testing. No penetration testing was explicitly required but was performed by the producer anyway, which is a clear case of over-delivery. The pentest reported an insecurely configured server within the system. But the server was never reconfigured and was later successfully abused in a spam attack. Apart from that the customer does not know of any security incidents, but they are "... not sure the maintainer tells us everything".

Security management standard implemented. The requirements specification referred to the ISO/IEC 17799 standard for security management [13] and the customer feels it has been implemented.

Fraud analysis requirement forgotten. The customer required a holistic risk analysis of potential frauds but does not recall that such an analysis has been done. We asked if they did not know they had required a fraud analysis and customer responded "No, that seems to be the truth".

Summary. The customer feels that the producer of the highway toll system is mature and has fulfilled some of their unspecified needs, i.e.needs not part of the requirements specification. Despite this, security problems have surfaced (e.g. logging), mostly due to unspecified security and privacy needs understood or noted by the producer.

5.6 Interview Medical Advice System

Outcome of the interview with the Medical Advice project leader:

Security requirements high. The customer considers the security requirements high "since the system handles medical records". Several laws restrict the handling of such information. Despite this, no risk analysis was performed. Instead the customer relied on in-house experience and competence.

Log analysis process undefined. The specification contains five different requirements on logging. But when asked about processes and routines for checking the logs the customer replied "We don't know. That aspect is not taken care of." It might be that the producer checks the logs—"Perhaps. We hope so." When asked if the producer knows what security and privacy issues to check for in the logs the customer replied "To be honest, I don't think we've discussed such security processes. But I expect the producers to tell us if something happens."

Protection of logs forgotten. The specifications contained a requirement stating that "... the logs shall be protected against manipulation." When asked if this requirement has been met the customer replied they "... don't have a clue".

Vague logging requirement. The specification said that "... sensitive information shall be logged and protected from manipulation." What *'sensitive information'* means is never specified. The customer agrees "... that the requirement is vague. It has been made more concrete during project iterations." But such iterative refinements can "... become a time and money discussion" according to the customer.

No known security incidents. We asked the customer if they have had any security incidents so far—"No, I don't think so."

Deliberately no security standard. In the customer's view it was valid to specify the security requirements without using security standards. One person, admittedly a local hero, was responsible for security—"He has not brought up standards as a possibility. He knows standards and legislations and thus I believe he processed the issue himself and chose not to point toward standards." But when the lack of requirements on physical security (fire, theft etc.) was brought up, the customer admitted that some security issues have been overlooked. "If we would have had a standard those areas would have been covered." To integrate standards into the requirements the customer says they would have needed help from a third party.

No security testing. No security tests apart from stress testing of availability were required or performed. "But it would have been nice" the customer commented.

Summary. The customer admits that proper measures to ensure security and privacy in the system have not been taken. Questions regarding processes for continuous log analysis revealed that such aspects had not been thought of before, and that the privacy needs had not been properly specified as requirements. The firm belief in their local hero had obvious disadvantages.

5.7 Results on General Security Requirements

In all cases the customers have had higher security and privacy needs than their requirements specifications reflected. They have relied heavily on their producers to handle technical security. In cases where security requirements were specified they were often vague or incomplete, which had led to negotiations and minor disputes. Also worth noting is that all three customers had security requirements that were either not implemented by the producers or forgotten by themselves. Despite this, the customers were mostly satisfied with the general security of the delivered systems.

5.8 Results on Domain-Specific Security Requirements

In all three cases the customers have had problems with domain-specific privacy concerns:

- Health Care 1: No support for handling patients with protected identities.
- Highway Tolls: No logging of accesses to privacy sensitive vehicle data.
- Medical Advice: No process for checking or protecting security logs although the system handles privacy sensitive patient data.

Part of their privacy needs were not expressed as requirements and were not fulfilled by the producers. Our impression was that the customers had not realized in what ways the systems could violate privacy until after delivery.

6 Discussion

According to our own experience, effective methods and so called “best practices” for software security are more and more becoming part of general software engineering expertise. Therefore, relying on the producers to deliver security despite vague or left-out requirements is not all bad. Besides, some security requirements can be fulfilled with commercial off-the-shelf products engineered by security specialized vendors, in which case the customer most likely gets much more security features than specified.

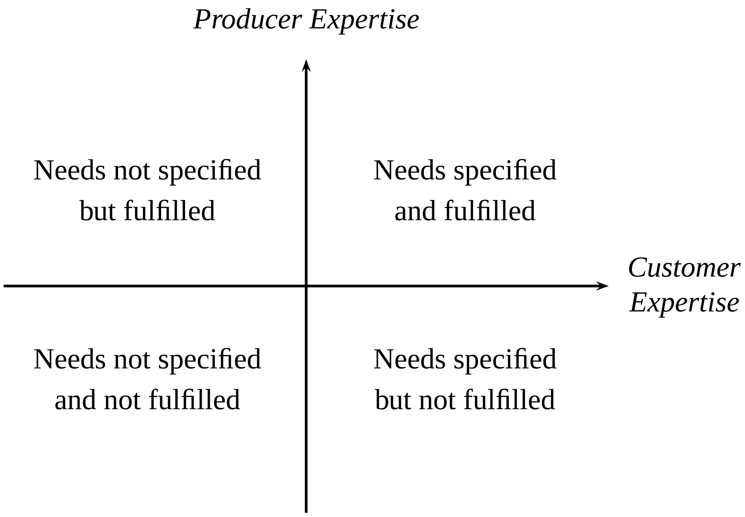
Producer Expertise
Needs not specified but fulfilled
Needs specified and fulfilled
Customer Expertise
Needs not specified and not fulfilled
Needs specified but not fulfilled

Figure 1. Quadrant diagram visualizing the separation between customer and producer expertise, and its consequences for customer needs. When producer and customer understand each other they end up in quadrant one (north east). When a certain need is not specified as a requirement by the customer and not noted by the producer they end up in quadrant three (south west).

In two of the three cases (Health Care 1 and Highway Tolls) the producers delivered more security than required by the specifications. We call this phenomenon *over-delivery*. When asked about over-delivery, the customers said that the producers were “mature” and did not want to deliver an insecure system. But the customers raised doubts as to whether some over-delivered parts, such as documentation and defined processes, were actually considered *intellectual property* of the producers. In the third case (Medical Advice) the customer often had had to re-negotiate to compensate for poor initial requirements.

All three customers had had problems with domain-specific privacy concerns, and the nature of the problems suggests that privacy needs are especially prone to being domain-specific. The customers themselves did not know they had such needs or did not realize in what ways the systems could violate privacy. Contrary to unspecified but more technical security needs, the producers did not over-deliver. A reasonable explanation for this would be that the producers did not know there were such needs within the domains the systems were being built for.

We tried to visualize the separation of domain knowledge and its consequences in terms of fulfilled and unfulfilled needs in a quadrant diagram (see Figure 1). The customers agreed that this was a relevant model of reality and that they had experience from all four quadrants in the projects.

6.1 Verification of Hypotheses

Our hypotheses (see Section 4) were verified against the interview outcome:

- **Security requirements incomplete**. All three customers admitted that their requirements specifications contained vague, inconsistent and incomplete security requirements.
- **Risk analyzes performed**. Two of the systems had performed at least some kind of risk analysis (unspecified which kind). Thus the poorly specified requirements were not clearly due to lack of risk assessment.
- **Local heroes phenomenon confirmed**. All three customers had relied heavily on so called local heroes and could actually name them. Nevertheless, there was an understanding that third-party consultants would probably have mitigated the problems with vague and unspecified needs.
- **Systems partly insecure**. Only one of the systems had had a known security incident, but none of the customers had a defined process for investigating if any security incidents occurred. Two of the systems had serious privacy flaws that had to be fixed in future versions.

6.2 Validation Against Maintainability Requirements

We conducted a parallel study on another non-functional requirement category, namely *maintainability*. The full results of that study is still to be published, but the material allowed us to do a comparative validation of our results on security and privacy requirements.

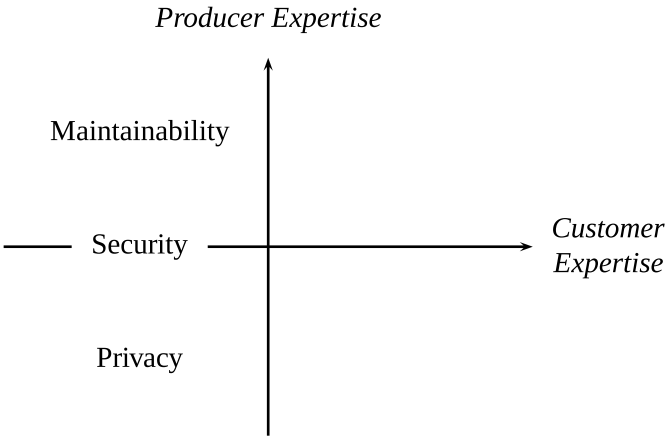

Figure 2. Quadrant diagram visualizing producer expertise within three non-functional requirements categories. Producers typically fulfill unspecified maintainability needs but not unspecified privacy needs.

Compared to security there seems to be a much higher degree of over-delivery in maintainability. One of the reasons for this is that maintainability requirements such as system documentation, regression test suits, and coding standards typically are general, i.e. not specific to the customer domain. Therefore they are part of the producer expertise and is a potential candidate for over-delivery—a mature producer doesn't skip documentation just because the customer failed to require it.

We can place the three non-functional requirements categories in another quadrant diagram (see Figure 2). Maintainability needs are part of general software engineering, (technical) security needs are more and more becoming part of software engineering, whereas privacy needs do not seem to be part of the software engineering domain yet.

7 Related Work

Several research studies have investigated security and privacy requirements (presented below). Unfortunately most of them treat the underlying problem as based on experience or as anecdotal. We hope to fill that gap, but our studies are of course related.

Alderson discusses the fact that vague or underspecified requirements (defined by him as *false requirements*) often express real customer needs [1]. His findings support that requirements specifications often leave out or fail to properly specify customer needs.

Anderson gives an example where a British bank system did not log customer address changes and a clerk abused the system by changing a chosen customer's address to her own, issuing a new ATM card and PIN, and then changing the address back [4]. This was possible since the bank's clerks had privileges to change both customer addresses and issue new ATM cards. A risk analysis would have had to include people with domain-specific knowledge of privileges and procedures at the bank to detect this security threat.

McDermott and Fox note that the security engineering process is complex and hard to understand even for skilled software engineers [19]. As an example they mention theoretical models for RBAC. This supports our conclusion that domain-specific needs will not be covered by software engineers even though they are skilled and have the best intentions.

Alexander shows by example that complex requirements problems can only be solved by the "combined domain knowledge and skill of the stakeholders" [2]. This pinpoints the fact that only the combination of various domain expertises can cover all the customer needs and formulate them as requirements.

Firesmith, Sindre, and Opdahl have shown that security and privacy requirements are often similar or exactly the same across various systems, at least when compared on an abstract, goal-oriented level [10, 11, 20]. This suggests that typical security requirements can be listed and reused in future projects (potentially in a refined form). To prove their point they provide examples of such reusable requirements.

Firesmith et al's results potentially conflict with ours since they seemingly suggest that there are no domain-specific security or privacy requirements. But, as they note themselves, careful elicitation is necessary to be able to *choose* among the reusable requirements. And as mentioned, their reusable requirements are all on an abstract, goal-oriented level, and thus need to be *refined* to comply with the customer's domain-specific needs.

Liu, Yu, Mylopoulos, and Cysnerios have presented a framework for modeling security and privacy requirements using the agent-oriented *i** language [17, 18, 24]. Traditionally, *i** iteratively models *actors*, *actor goals*, and *actor dependencies*. In the context of security and privacy they add models of the counterpart, i.e. *attackers*, *attacker goals*, *vulnerabilities*, and *countermeasures*.

Use cases, introduced by Jacobson [14], have been used to model (mostly functional) requirements for long, for instance within RUP [15]. But security and privacy are often conceived as non-functional requirements [5, 6, 7, 9] and therefore not suited for use cases [21]. To elicit such requirements McDermott and Fox proposed *abuse cases*, which model interactions between systems and one or more actors, where the results of the interactions are harmful to systems, actors, or system stakeholders [19]. Very similar to abuse cases are *misuse cases* [2, 3, 21], *abuse frames* [16], and *anti-requirements* [8]. Since abuse/misuse cases identify and model threats they have much in common with *threat modeling* [12]. But threat modeling involves analysis of data flow and is therefore typically carried out later when there is a high-level design of the system.

8 Conclusions

Current practice in security requirements is in many ways poor. Customers tend to rely on local competence to specify security and privacy requirements, and tend to rely on the software producers to cover up for unspecified requirements.

Mature software producers seem to cover up fairly well for unspecified requirements that are part of the general software engineering domain. But our in-depth interviews with customer representatives show a severe impact of neglecting the specifics of the customer domain in eliciting security and privacy requirements. Unspecified requirements specific to the customer domain are unlikely be fulfilled by software engineers even though the engineers are skilled and have the best intentions.

According to our results, privacy requirements seem especially prone to being domain-specific. All three customers we interviewed had privacy needs that were never specified as requirements and never fulfilled by their producers. Customers need to cooperate with both producers and domain experts such as security specialists to be able to identify their needs and formulate them as requirements. Only the combination of various domain expertises has the potential to cover all the customer needs. This cooperation could be performed within the scope of risk analyzes.

9 Acknowledgments

We would like to sincerely thank the previewers of this paper and of course our interview participants.

References

[1] A. Alderson. False requirements express real needs. *Requirements Engineering*, 4:60–61, May 1999.

[2] I. Alexander. Initial industrial experience of misuse cases in trade-off analysis. In *Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE'02)*, pages 61–68, Essen, Germany, September 2002.

[3] I. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20:58–66, January/February 2003.

[4] R. J. Anderson. *Security Engineering—A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, 2001.

[5] J. Burge and D. Brown. NFRs: Fact or fiction? Computer Science Technical Report, Worcester Polytechnic Institute, WPI-CS-TR-02-01 `ftp://ftp.cs.wpi.edu/pub/techreports/pdf/02-01.pdf`, November 2002.

[6] L. Chung, B. A. Nixon, and E. Yu. Using quality requirements to systematically develop quality software. In *Proceedings of the Fourth International Conference on Software Quality*, McLean, VA, USA, October 1994.

[7] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

[8] R. Crook, D. Ince, L. Lin, and B. Nuseibeh. Security requirements engineering: When anti-requirements hit the fan. In *Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE'02)*, Essen, Germany, September 2002.

[9] P. T. Devanbu and S. Stubblebine. Security and software engineering: A roadmap. In *Proceedings of the Twenty-second International Conference on Software Engineering, ICSE*, Limerick, Ireland, June 2000.

[10] D. G. Firesmith. Engineering security requirements. *Journal of Object Technology*, 2(1):53–68, January–February 2003.

[11] D. G. Firesmith. Specifying reusable security requirements. *Journal of Object Technology*, 3(1):61–75, January-February 2004.

[12] M. Howard and D. LeBlanc. *Writing Secure Code, 2nd Edition*. Microsoft Press, 2002.

[13] International Organization for Standardization. ISO/IEC 17799:2000 information technology – code of practice for information security management. `http://www.iso.org/iso/en/prods-services/popstds/informationsecurity.html`.

[14] I. Jacobson. Object oriented development in an industrial environment. In *Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA'87)*, pages 183–191, Orlando, Florida, USA, 1987.

[15] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley Professional, December 2003.

[16] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett. Introducing abuse frames for analysing security requirements. In *Proceedings of 11th International IEEE Requirements Engineering Conference (RE'03)*, pages 371–372, Monterey Bay, CA, USA, September 2003.

[17] L. Liu, E. Yu, and J. Mulopoulos. Security and privacy requirements analysis within a social setting. In *Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03)*, pages 151–161, Monterey Bay, CA, USA, September 2003.

[18] L. Liu, E. Yu, and J. Mylopoulos. Analyzing security requirements as relationships among strategic actors. In *Proceedings of the 2nd Symposium on Requirements Engineering for Information Security (SREIS'02)*, Raleigh, North Carolina, USA, October 2002.

[19] J. McDermott and C. Fox. Using abuse case models for security requirements analysis. In *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99)*, Scottsdale, AZ, USA, December 1999.

[20] G. Sindre, D. G. Firesmith, and A. L. Opdahl. A reuse-based approach to determining security requirements. In *Proceedings of the 9th International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'03)*, Klagenfurt/Velden, Austria, June 2003.

[21] G. Sindre and A. L. Opdahl. Eliciting security requirements by misuse cases. In *Proceedings 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS Pacific)*, Sydney, Australia, November 2000.

[22] I. Sommerville and P. Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, 1997.

[23] J. Wilander and J. Gustavsson. Security requirements—a field study of current practice. In *E-Proceedings of the Symposium on Requirements Engineering for Information Security, in conjunction with the 13th IEEE International Requirements Engineering Conference*, Paris, France, `http://www.sreis.org`, August 2005.

[24] E. Yu and L. M. Cysneiros. Designing for privacy and other competing rerquirements. In *Proceedings of the 2nd Symposium on Requirements Engineering for Information Security (SREIS'02)*, Raleigh, North Carolina, USA, October 2002.