Achieving “Good Enough” Software Security: The Role of Objectivity

Inger Anne Tøndel
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
inger.anne.tondel@ntnu.no

Daniela Soares Cruzes
SINTEF Digital
Trondheim, Norway
daniela.s.cruzes@sintef.no

Martin Gilje Jaatun
SINTEF Digital
Trondheim, Norway
martin.g.jaatun@sintef.no

ABSTRACT

Today’s software development projects need to consider security as one of the qualities the software should possess. However, overspending on security will imply that the software will become more expensive and often also delayed. This paper discusses the role of objectivity in assessing and researching the goal of good enough security. Different understandings of objectivity are introduced, and the paper explores how these can guide the way forward in improving judgements on what level of security is good enough. The paper recommends adopting and improving upon methods that include different perspectives, support the building of interactive expertise, and support confirmability by keeping documentation of the basis on which judgements were made.

CCS CONCEPTS

• **Security and privacy → Software security engineering**; • **Software and its engineering → Agile software development**; **Risk management**; *Requirements analysis.*

KEYWORDS

software security, objectivity, security level, good enough security, security priorities, agile software development

ACM Reference Format:
Inger Anne Tøndel, Daniela Soares Cruzes, and Martin Gilje Jaatun. 2020. Achieving “Good Enough” Software Security: The Role of Objectivity . In *Evaluation and Assessment in Software Engineering (EASE 2020), April 15–17, 2020, Trondheim, Norway*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3383219.3383267

1 INTRODUCTION

In today’s interconnected and digitized world, a large portion of the software that is developed needs to consider security. This is the case not only for software that is considered security critical (e.g., military systems), but also for the more “normal” type of software (e.g., web applications, mobile apps). The goal and focus of these software development approaches is to deliver value to the customers, and security is commonly seen as a secondary goal; something that must be considered but usually would imply extra development time and costs. Viewed this way, security could have a negative impact on the output of functionalities, and over-spending on security may make the software less competitive or successful. However, not putting in the necessary security is not a good option, as this may cause severe problems later on.

Security experts in general would agree that perfect or total security is an illusion [13, 24], this is e.g. a foundation for a risk-management approach to security [13] - an approach taken in major security standards such as ISO/IEC 27001 [14]. Determining what is good enough is however hard [24]. Beznosov [5] suggests not defining what is “good enough”, but rather letting the customer define and adjust security needs as the project progresses, utilizing the customer involvement built into agile development approaches. Sandhu [24] suggests two design principles: “Designing with the application in mind”, and viewing security as being “about trade-offs, not absolutes” [24]. Hurlburt [13] points to the human factor as a reason why “systems security will never be better than good enough” [13]. He acknowledges the importance of investing in a robust upfront security design, but argues that this will not solve the problem completely. He points out that with today’s distributed attacks there is a need for more overarching approaches, not only considering whether an attacker might be discouraged from attacking one particular system. He suggests instead, “an objective, consensus-based rating system” [13] that companies can use to rank risks of different products and organizations, and claims that through the use of such a rating system one may establish a kind of working threshold that defines what is considered good enough when it comes to security.

Objectivity is a key characteristic for research in general, and even a part of research ethics [21]. Objectivity can be understood in different ways, but is often considered as a striving towards avoiding bias [9, 21] to ensure adequate “standing of our judgements and interpretations” [9]. Thus, objectivity is related to a confidence that multiple observers could come to similar judgements [22]. We agree with Hurlburt that there is a need to objectively assess what is good enough when it comes to security. We additionally agree with Sadhu that “good enough” is not something that can be defined outside of the context of a particular project, organization or product. In this emerging results paper we continue the discussion on objectivity in relation to security, exploring what it would mean to have an “objectively correct” level of security, and which security analysis approaches can support objective judgements about “good enough security”. We build on insights from our own empirical studies on software security performed over a period of several

EASE 2020, April 15–17, 2020, Trondheim, Norway
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7731-7/20/04…$15.00
https://doi.org/10.1145/3383219.3383267

years. Additionally, we introduce theory on objectivity and examine what this type of theory may bring to security practice and research.

The paper is organised as follows. Section 2 introduces the concept of objectivity in more detail, and different ways to understand objectivity. Section 3 explains the research approach taken in the empirical research that underlie the claims we make in this paper about software security practice in agile development projects. Then Section 4 exemplifies challenges in reaching objective evaluations of software security. In Section 5 we propose and discuss strategies that could help increasing objectivity both in software security analysis and in research on "good enough" software security. Section 6 concludes the paper.

2 AN INTRODUCTION TO DIFFERENT UNDERSTANDINGS OF OBJECTIVITY

In this paper we use a set of different philosophical understandings of objectivity, as described by Gaukroger [9], to structure our introduction to objectivity. Additionally, we bring in understandings and practical considerations on how to achieve objectivity from research methods literature.

Objectivity can be understood as "a judgement that is free of prejudice and bias" [9], or even as "a judgement which is free of all assumptions and values" [9]. These understandings are both describing a "particular state of mind" [9], and are both negative theories of objectivity, stating what should be removed in order to be objective. Aiming to be free of prejudice and bias can be a challenging endeavor, but Gaukroger argues that it is a sensible goal [9]: "Objectivity requires us to stand back from our perceptions, our beliefs and opinions, to reflect on them, and subject them to a particular kind of scrutiny and judgement. Above all, it requires a degree of indifference in judging that may conflict with our needs and desires." [9]. Removing all assumptions and values is however not possible [9, 20]; nothing is a "view from nowhere" [9], all beliefs are socially situated [11].

Quantitative methods have long been associated with objectivity. There is a seeming neutrality that comes from having numbers - "The numbers speak for themselves!" [9]. Gaukroger argues that practices that could fall into the term 'number crunching' "are not necessarily subjected either to reasoned judgement or to the empirical evaluation of particular cases, but typically bypass any form of independent or objective reasoning at all" [9]. Aiming for a judgement free from prejudice and bias does not mean eliminating judgement. Instead of letting the numbers be "a substitute for decision making" they could be used as "an aid to decision making" [9], or even as illustrations allowing for a more persuasive argument [7].

Objectivity can be understood as consisting of "accurate representations" [9]. What constitutes an accurate representation is however subject to judgement, and can be considered differently depending on what one wants the representations for. Objectivity is not an absolute - you are not either objective or not objective - but rather there are degrees of objectivity. The understanding of objectivity as accurate representations points to objectivity as "something that can be learned and improved upon through practice" [9]. "Trained judgement" as well as "identification and elimination of arbitrary judgement" [9] becomes important traits of objectivity. Objectivity as accurate representation however is costly and needs to be balanced against other concerns.

Objectivity can be understood related to the procedure used, viewing an objective procedure as "one that allows us to decide between conflicting views of theories" [9]. Objectivity is seen as a core aspect of science [9], and the methods used in science is expected to support objectivity. The scientific method and the progress towards better and better theories rely on theories being falsifiable and that scientists do serious attempts at refuting theories [20]. It has however been argued that the way objectivity is used in science does not necessarily fit other contexts, e.g. the needs when studying human behaviour [9] or when using other research paradigms [17] than those using conventional scientific methods. The types of methods used in e.g. sociology have some fundamentally 'subjective' traits where research approaches to eliminate subjectivity, such as double-blind testing, do not work [7]. This however does not mean that qualitative studies cannot strive for and demonstrate objectivity. In the following we introduce two examples of this: the suggestion to replace objectivity with confirmability in naturalistic studies [17], and the concept of strong objectivity from feminist studies [11].

Lincoln and Guba [17] argue that confirmability is a preferable concept to objectivity within the naturalistic paradigm, a primarily qualitative research paradigm where studies are performed in natural settings and researchers avoid manipulating the research outcomes a priori [17]. A move towards confirmability removes the issue from "the investigator's characteristics" to "the characteristics of the data: Are they or are they not confirmable?" [17] According to Lincoln and Guba, confirmability is tightly linked with auditability, and they argue that research studies must establish an audit trail consisting of (e.g.) raw data, data reduction and analysis products, data reconstruction and synthesis products, process notes, materials relating to intentions and dispositions, and instrument development information [17]. Confirmability thus requires that the research design is constructed in such a way that the audit trail is preserved, and Lincoln and Guba also state that an actual audit must take place.

Similar thoughts to that of Lincoln and Guba can be found in several other qualitative methods textbooks. Examples include Miles and Huberman [19] who proposed a set of questions to ask of a qualitative study about objectivity. These questions cover to what extent the methods are described explicitly and in detail, whether there is a record of the study detailed enough to be considered an audit trail, whether it is possible to follow the sequence of data collection, processing and presentation, whether researcher assumptions, values and biases are made explicit, whether competing hypotheses are considered, and whether study data is retained and available for re-analysis by others. Additionally, Collins in his introduction to sociology research affirms that "if qualitative research is to deserve the label of "science" it should be conducted in such a way that it could be replicated *in principle*" [7]. Collins however does not only link replicability to method concerns, but also to the ability to generalise from the results - "as the significance broadens, there are more and more ways of checking" [7]. On a more practical level, confirmability seems analogous to accountability of (e.g.) service providers, since both concepts aim to verify that the researchers cf. service providers are "doing the right thing". In security research, the concept of accountability came into prominence with the introduction of the EU General Data Protection Regulation, where the ability to demonstrate that handling of sensitive personally identifiable information is performed in a compliant manner became an explicit requirement. An accountable organization must define what it does when it handles personal data, monitor how it acts, remedy any discrepancies between the definition of what should occur and what is actually occurring, and explain and justify any related action [16].

The concept of strong objectivity stems from feminist standpoint theory [11]. Standpoint theory points out that all knowledge arises in particular social situations with people with particular social positions, and thus is not value-free. The concept of strong objectivity brings the assumptions and agendas of the researchers into the research as part of what is investigated, acknowledging that these are not easily detected by individuals. It claims that by taking the perspectives of the marginalized or oppressed one can achieve more objective knowledge. Marginalised individuals are “outsiders within” [11] and are thus able to understand both their own position and that of the dominant culture.

Gaukroger additionally points out another possible understanding of objectivity, namely that “something is objective if it leads to conclusions which are universally accepted” [9]. Similarly, Robson [22] claims that ‘objective’ can be taken to refer to what multiple observers agree to as a phenomenon, in contrast to the subjective experience of the single individual. However, Gaukroger warns that “we should not assume that there is a correlation between degree of agreement and degree of objectivity” [9].

To sum up, objectivity can be understood in different ways. There is a need to consider for particular cases “what we want out of objectivity” [9] and how objectivity can be secured. In the remaining parts of this paper we look more closely at objectivity related to judgement about the security level, and in particular what level of security is “good enough”. We make use of the understandings of objectivity as 1) procedures that allow one to decide between conflicting views, 2) accurate representations, and 3) freedom from prejudice and bias, and we touch upon the understanding of objectivity as universally accepted conclusions (see Table 1).

3 RESEARCH METHODOLOGY

Our claims about agile software security in this paper are based on involvement over several years with companies on the topic of software security in agile development. This includes interview studies involving about 20 public companies with the aim to identify practices and challenges in agile development [15, 28] and action research involving several companies, as part of the SoS-Agile research project [8]. In this project we have investigated how to meaningfully integrate software security into agile software development activities. The companies we have worked with are varied in their size, the type of software they develop and their organization, and include smaller development departments, distributed development teams, and larger development organisations. We have studied individual projects, as well as overall organizational approaches to software security in agile development. Our major involvement has been with three companies, and these have been studied over several years. In addition to those, we have had shorter collaborations with five companies on more specific issues.

Table 1: Understandings of objectivity used in the examples

Understanding of objectivity	Example from software security
Objectivity as a procedure that allows one to decide between conflicting views – with the scientific method and its focus on falsification as an example of such a procedure.	To what extent “we have good enough software security” is a falsifiable claim, i.e., whether one is able to identify cases of security levels that are too low as well as too high.
Objectivity as accurate representations, and objectivity as universally accepted conclusions.	The different roles/stakeholders involved in judgements about software security, and their varying viewpoints and understanding about software security.
Objectivity as freedom from prejudice and bias.	The prejudice and bias commonly found among security experts when approaching a development project.

In action research, the aim is to merge theory and practice in such a way that real-world problems are solved by theory-informed actions in collaboration between researchers and practitioners [10]. In our research, the “action” has been the introduction of various security practices; threat modeling, static analysis tools, self management for security and security requirements work. To obtain a wide understanding of the transformation phenomenon, various data collection mechanisms have been applied, including observations, interviews, questionnaires and document analysis, and we have built a close relationship with the software companies. Building a close relationship is important in any action research study. Aspects of security work however makes it even more important, in our experience. This is due to the secrecy and sensitivity of the information and artefacts that are dealt with in the organization, but also that security requirements are mostly non-functional and not really the focus in the daily activities of software teams.

4 OBJECTIVITY CONCERNS IN JUDGEMENTS ABOUT “GOOD ENOUGH” SOFTWARE SECURITY

In the following we exemplify challenges in reaching objective evaluations of the level of software security, and in particular whether the level of software security is “good enough”. We draw upon different understandings of objectivity, as shown in Table 1.

4.1 Is “we have good enough security” a falsifiable claim?

There are a variety of ways one may go about evaluating the level of software security in a project. Still, judging whether the security level is too high or too low – currently and in the near future – is not straightforward. We illustrate this with some examples.

A na’́ive evaluation of the claim “we have good enough security” would be to consider whether the system experiences any security incidents; thus a system that is successfully attacked would not be satisfactorily secure. There is some merit to this, however, there is a need for ways to evaluate security that are less reactive. Results of code review, static code analysis and security testing offer a more proactive evaluation of security, and can provide some assertion that one is on the right track. This does not mean, though, that it is clear how much analysis and testing is enough. Additionally, all security issues identified using such approaches do not necessarily need to be addressed to achieve good enough security. Another seemingly straightforward way of addressing whether the software security is good enough, is to consider whether it meets legislative and customer requirements on security. However, customers often consider security as an implicit requirement that should be taken care of by the developers [2], and legislative requirements and their concrete implications for the project can be a case of debate and negotiation between different types of experts [28].

It has been argued that it may be easier to evaluate software security based on the processes performed, rather than the software itself [18]. In our research and interaction with software development companies we have often used the Building Security In Maturity Model (BSIMM) [18, 31] as a tool to help companies identify practices that they want to apply or improve [15]. All software security activities included in the BSIMM are activities that are performed in real companies. The BSIMM additionally identifies which activities are most commonly adopted by software companies, and, by extension, most companies would probably benefit from doing. However, companies are of different size and develop software with different security requirements. In our work with small and medium sized software development companies, we find that it is not trivial to know which activities to recommend to a software company, despite the knowledge of which activities are most common. A BSIMM evaluation would give you knowledge about what activities you do and don’t do, but not whether you have adopted a set of practices that fits your needs. Additionally, doing a full scale BSIMM evaluation is costly. In our use of BSIMM as a research tool we have relied heavily on self-evaluation, and thus on the company’s own understanding of their own practices. Interestingly, we have observed that in one company their overall BSIMM self-evaluation scores actually dropped after investing in several improvements in their software security practices. This was because they now had a better understanding of the implications of the BSIMM activities and the limitations of their own practices.

A mantra in most security work is that the approach should be risk based, meaning that one is aware of what the main risks are, and targets those in a strategic manner. Risk assessments should then ideally help software projects identify this “good enough” level of security. There exists a variety of methods for performing risk assessments related to information or software security, some highly detailed and quantitative in nature, others less formal and qualitative in nature. There is research showing that software projects often do not have a risk-based approach to software security [28]. Still, it is safe to assume that agile software projects would typically rely on qualitative risk analysis with expert evaluations for security risk assessments, as this is a cost-effective approach and can be done in a relatively short amount of time. Such analysis has been critiqued for not measuring risk, but rather “human judgement about security risk” [12], and that though this judgement can be useful, it comes with its limitations. It is an open question how much effort needs to be put into a risk analysis for the result to be reliable. Note also that security is in many ways a moving target – new vulnerabilities and attacks can invalidate previous assumptions and thus demand a new risk assessment to be performed.

The kinds of security analysis introduced above would mainly be able to identify lacks in security. Indications of too much security would probably come in other forms, e.g. through loss in competitiveness and broken deadlines. These are however very indirect measures of the security level, and these problems may stem from sources not related to security as well. There are many approaches to evaluate cyber security investments, with the Return on Security Investments being one example [4]. In practice however, we find that companies we interact with only discuss this in informal terms.

Based on the above we would claim that though it may be possible to state after-the-fact that the security at some point was too low, it is very difficult to know if a project invests more than necessary on security, or if the same investments could be more efficiently used in a different way. Indicators may be introduced throughout, but this does not necessarily increase objectivity if not paired with proper judgment about what kind of decision support they provide.

4.2 Can you see something you don’t have knowledge about?

In ongoing research on software security requirements and priorities in agile projects, we have used interviews as one of the data collection methods, and have among other things asked different actors in a software development project to what extent they believe they ended up with good enough security in the project; not too high or too low. In a project where we asked a security champion (SC) of a team, a product owner (PO) and a technical product owner (TPO) this question, they all agreed the security was good enough, but they had different explanations as to why that was the case.

The SC is a developer that is a regular part of the development team but has been assigned some responsibility for security. The SC thought security had been given the right priority. The project could have done more on security, but then the SC believed they would have had problems finishing. The SC had observed a raised security awareness in the development team compared to previous projects, and that this had impacted the software developed, thus the security level was not considered too low either.

The TPO has a background as a developer and software architect and brings his technical background into strategic discussions and priorities in the project. The TPO believed the level of security was about right, but that the sense of responsibility for security should be different so that everybody took responsibility for security without having to make a big process around it. The TPO believed security was important but did not have capacity to take this on as yet another task.

The PO is responsible for prioritising the requirements and represents the customer interests. In this project the PO had focus on following up the formal security requirements towards the customer to ensure the contractual obligations were met. The PO trusted the

TPO and the SC, was aware that the SC and TPO spent some time on security, and consequently trusted that the level of security was about right.

We bring out this example to illustrate that the security level and priorities are viewed differently based on the position and the competence of the one making the evaluation; in this case from a perspective of how security is perceived by the team (SC), the sense of responsibility (TPO) and trust in others (PO). The PO that does not have that deep technical knowledge, including on security, is not in a position to see security problems if not told about this, and is not aware of security issues that have come up along the way and have been decided upon by the team (while the SC and TPO are aware of this).

With such varying viewpoints, a relevant question is 'What is an accurate representation?' The common understanding is not necessarily the most objective one, as this is a highly specialized topic and objectivity in representations can require skills and training [9]. However, is it also possible that individuals with a lot of security awareness and knowledge may see "too many" security issues or at least more than what you, from a business perspective, would want to invest on fixing?

4.3 Is prejudice and bias a driver for software security assessments and research?

Security analysis and research is often done based on the assumption that the security work currently is not good enough and needs to be improved. This can be considered a form of prejudice and bias – the state of software security is, before any study has been performed, considered to be too low. Adding to this potential challenge is what has been characterised as a "disconnect between security and development" [29] stemming from these expert communities traditionally being isolated from each other. Thus, security experts commonly lack an adequate understanding of development [29].

The role and mindset of a security expert is often to identify problems; that lies in the nature of the field and the tasks. This could be represented by the auditor role. Security experts could however assume another role, that of the guide or the supervisor, providing support to developers and strengthening what they are already doing that is good. This requires another skill set [29] and possibly a more positive attitude. It is an open question to what extent the mindset of the security expert, it being that of auditor or support, affects how they assess the security level.

5 WAY FORWARD

The previous section exemplified challenges of making objective judgements of whether the security is good enough. In the following we suggest ways in which practitioners and security researchers can draw on the different understandings of objectivity to improve their judgements of security. We discuss the following strategies: including a variety of perspectives, building interactional expertise, and supporting confirmability.

In security analysis, it is quite common to aim to "think like an attacker", e.g. as is done in threat modeling [25]. We would claim that bringing in more perspectives is one way of increasing accuracy of representation. Relevant perspectives include not only that of the attacker, but that of developers, operations, customers, users, managers, etc. When taking the attacker's perspective this is done as an exercise in thinking, but other types of actors may even be invited to take part in the analysis. Several security techniques support this kind of involvement; risk analysis can be done with a wide range of participants, games such as Protection Poker [30, 32] invite broad attendance though mainly from the development team, and techniques such as the Security Intention Meeting [27] aim to include the management level regularly in high-level security analysis and decisions. The cost of such involvement, however, needs to be taken into account.

In addition to improving accuracy of representation, bringing in representatives of different perspectives can potentially help flash out prejudices and bias of the researcher or security practitioner doing the security analysis. The concept of strong objectivity [11] points to this potential role of being "gazed back" at from the objects of study, and through this being able to gaze back at oneself, one's own socially situated beliefs and practices, from a location further away from daily work [11]. The true effects of strong objectivity come with practices that are too extensive and thus out of the scope for the topic of this paper. Still, one may likely experience some of these effects simply by including and truly listen to other perspectives on the security level. Experiences from using the concept of strong objectivity in a transdisciplinary research project [23] point to benefits of being open and transparent about own positions and standpoints; "there *always* exist value judgements in science. Reaching objectivity requires not only making these transparent and accessible, but also necessitates submitting those judgements to an open and rational debate" [23]. This goes beyond just engaging different stakeholders and includes such things as addressing power imbalances.

In literature, the PO role has been found to commonly limit the priority given to security [1, 26]. The business case for security is often considered unclear [26] while the push for functionality is strong, and this results in less focus on security [1, 26, 28]. In our experience with companies, we see that the PO role can act as a hindrance for security in many cases. Additionally, we observe that POs often have limited competence about software security. Building security competence at the PO level could be a way to increase the PO's ability to make good judgements related to security priorities. This is however not a one way street. There is a need for security experts, as well as security concerned developers, to understand the perspectives of the PO so that these roles can have fruitful discussions about security and project priorities. To cite Sandhu: "We are completely clueless about what is good enough. [...] Business people cannot tell us because they don't understand security and security people cannot tell us because they don't understand business. We must close this divide" [24].

We exemplify this need by addressing an underlying assumption in this paper, namely that too much security will lead to drawbacks such as increased cost, reduced usability, etc. Though this is commonly considered to be the case, it is not necessarily always true. The Privacy by Design [6] initiative does in one of its principles encourage the move away from zero-sum thinking about privacy, to positive-sum, looking for win-win solutions that is good for privacy as well as other goals of the system. In the same way, if the thinking about security is mainly that it hampers other system goals, one may miss solution alternatives where security can help achieving other types of goals in the system as well. Having security experts with a better understanding of the project goals is one possible step towards a kind of thinking that may lead to positive-sum solutions. The need for competence lies at the level of interactional expertise [7], that is, there is a need for being able to understand each other, speaking the language, but not being able to perform or directly contribute to each others tasks. Meeting and talking together can help build the necessary trust and understanding [7].

This paper has informally discussed several types of indicators to approach an evaluation of what is good enough, including the time spent on security, security testing results, what other companies do (BSIMM), what some experts assess to be the need (risk analysis), the security awareness in the development team, etc. More research is needed to know to what extent there is a good correlation between any of these indicators and the security of the final system. We do not in this paper make any claims as to what types of security analysis methods would best support objective security evaluations, apart from our recommendation to include different viewpoints in the analysis. More research is needed in order to make such claims. However, we draw on the concept of confirmability in recommending that the judgements as well as the reasons behind any judgements are kept so that assumptions and decisions can be revisited at a later stage. This is important in case there is a security incident, but also in order to deal with changing threat landscapes and project goals. Note however that though this is to some extent in conflict with the agile manifesto and its emphasis on "[w]orking software over comprehensive documentation" [3], agile is not about no documentation, and it is possible to document these types of issues as part of e.g. the software's structure, commit messages, unit tests and comments.

6 CONCLUSION

This paper has used theory on objectivity to see how it can improve both researchers' and practitioners' assessment of what is good enough when it comes to software security. More research is needed in order to provide agile-friendly and concrete method support on achieving objective judgements about what is "good enough". As a way forward, this paper suggests researching and adopting practices that include different perspectives, supports the building of interactive expertise among key actors, and that support confirmability by documenting judgements about security and their rationale.

ACKNOWLEDGMENTS

This work was supported by the SoS-Agile: Science of Security in Agile Software Development project, funded by the Research Council of Norway (grant number 247678, https://www.sintef.no/en/digital/sos-agile/). Thanks to Prof. Jonathan Knowles and Prof. Colin Boyd for commenting on this paper at various stages.

REFERENCES

[1] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. 2017. Quality requirements in large-scale distributed agile projects–a systematic literature review. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 219–234.

[2] S. Bartsch. 2011. Practitioners' Perspectives on Security in Agile Development. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 479–484. https://doi.org/10.1109/ARES.2011.82

[3] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. 2001. Manifesto for agile software development. *Online at http://www.agilemanifesto.org* (2001).

[4] Stefan Beissel et al. 2016. *Cybersecurity investments*. Springer.

[5] Konstantin Beznosov. 2003. Extreme security engineering: On employing XP practices to achieve'good enough security'without defining it. In *First ACM Workshop on Business Driven Security Engineering (BizSec). Fairfax, VA*. Citeseer.

[6] Ann Cavoukian et al. 2009. Privacy by design: The 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada* 5 (2009).

[7] Harry Collins. 2019. *Forms of Life: The Method and Meaning of Sociology*. MIT Press.

[8] Daniela S. Cruzes, Martin G. Jaatun, and Tosin D. Oyetoyan. 2018. Challenges and Approaches of Performing Canonical Action Research in Software Security: Research Paper. In *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HoTSoS '18)*. ACM, New York, NY, USA, Article 8, 11 pages. https://doi.org/10.1145/3190619.3190634

[9] Stephen Gaukroger. 2012. *Objectivity: A very short introduction*. Oxford University Press.

[10] Davydd J Greenwood and Morten Levin. 2006. *Introduction to action research: Social research for social change*. SAGE publications.

[11] Sandra Harding. 1991. Strong objectivity" and socially situated knowledge. *Whose science* (1991), 138–163.

[12] Lance Hayden. 2010. *IT security metrics: A practical framework for measuring security & protecting data*. Vol. 396. McGraw Hill New York.

[13] George Hurlburt. 2016. " Good Enough" Security: The Best We'll Ever Have. *Computer* 49, 7 (2016), 98–101.

[14] ISO. 2013. Information technology – Security techniques – Information security management systems – Requirements. ISO/IEC Standard 27001:2013. https://www.iso.org/standard/54534.html

[15] Martin Gilje Jaatun, Daniela S. Cruzes, Karin Bernsmed, Inger Anne Tøndel, and Lillian Røstad. 2015. Software Security Maturity in Public Organisations. In *Information Security*, Javier Lopez and Chris J. Mitchell (Eds.). Lecture Notes in Computer Science, Vol. 9290. Springer International Publishing, 120–138.

[16] Martin Gilje Jaatun, Siani Pearson, Frédéric Gittler, Ronald Leenes, and Maartje Niezen. 2016. Enhancing Accountability in the Cloud. *International Journal of Information Management* (2016). https://doi.org/10.1016/j.ijinfomgt.2016.03.004

[17] Yvonna S Lincoln and Egon G Guba. 1985. *Naturalistic inquiry*. Sage Publications Inc.

[18] Gary McGraw, Sammy Migues, and Jacob West. 2018. *BSIMM 9*. Technical Report. Synopsys, Inc.

[19] Matthew B. Miles and A. Michael Huberman. 1994. *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Sage.

[20] Karl R Popper. 1972. *Objective Knowledge: An Evolutionary Approach*. Oxford University Press, Chapter The bucket and the searchlight: Two theories of knowledge.

[21] David B Resnik et al. 2011. What is ethics in research & why is it important. *National Institute of Environmental Health Sciences* 1, 10 (2011), 49–70.

[22] Colin Robson. 2011. *Real World Research* (3 ed.). John Wiley & Sons.

[23] Judith Rosendahl, Matheus A Zanella, Stephan Rist, and Jes Weigelt. 2015. Scientists' situated knowledge: Strong objectivity in transdisciplinarity. *Futures* 65 (2015), 17–27.

[24] Ravi Sandhu. 2003. Good-enough security. *IEEE Internet Computing* 7, 1 (2003), 66–68.

[25] Adam Shostack. 2014. *Threat Modeling: Designing for Security*. Wiley.

[26] Evenynke Terpstra, Maya Daneva, and Chong Wang. 2017. Agile Practitioners' Understanding of Security Requirements: Insights from a Grounded Theory Analysis. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. IEEE, 439–442.

[27] Inger Anne Tøndel, Daniela Soares Cruzes, Martin Gilje Jaatun, and Kalle Rindell. 2019. The Security Intention Meeting Series as a way to increase visibility of software security decisions in agile development projects. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. ACM, 59.

[28] Inger Anne Tøndel, Martin Gilje Jaatun, Daniela Soares Cruzes, and Nils Brede Moe. 2017. Risk Centric Activities in Secure Software Development in Public Organisations. *International Journal of Secure Software Engineering (IJSSE)* 8, 4 (2017), 1–30.

[29] K. R. van Wyk and G. McGraw. 2005. Bridging the gap between software development and information security. *IEEE Security & Privacy* 3, 5 (2005), 75–79.

[30] Laurie Williams, Michael Gegick, and Andrew Meneely. 2009. Protection poker: Structuring software security risk assessment and knowledge transfer. In *International Symposium on Engineering Secure Software and Systems*. Springer, 122–134.

[31] Laurie Williams, Gary McGraw, and Sammy Migues. 2018. Engineering Security Vulnerability Prevention, Detection, and Response. *IEEE Software* 35, 5 (2018), 76–80.

[32] Laurie Williams, Andrew Meneely, and Grant Shipley. 2010. Protection poker: The new software security game. *IEEE Security and Privacy* 8, 3 (2010), 14–20.