Talking about Security with Professional Developers

Conference or Workshop Item

Talking about Security with Professional Developers

Tamara Lopez*, Helen Sharp*, Thein Tun*, Arosha Bandara*, Mark Levine† and Bashar Nuseibeh*‡
*School of Computing & Communications, The Open University, Milton Keynes, UK
†Department of Psychology, University of Exeter, Exeter, UK
‡Lero - The Irish Software Research Centre, University of Limerick, Limerick, Ireland
Email: *firstname.lastname@open.ac.uk, †firstname.lastname@exeter.ac.uk, ‡firstname.lastname@lero.ie

***Abstract*—This paper describes materials developed to engage professional developers in discussions about security. First, the work is framed in the context of ethnographic studies of software development, highlighting how the method is used to explore and investigate research aims for the *Motivating Jenny* research project. A description is given of a series of practitioner engagements, that were used to develop a reflection and discussion tool using security stories taken from media and internet sources. An explanation is given for how the tool has been used to collect data within field sites, offering a way to clarify and *member check* findings, and to provide a different view on practice and process. The report concludes with observations and notes about future aims for supporting and encouraging professionals to engage with security in practice.**

***Index Terms*—secure software development, collaborative environments, empirical studies**

I. Introduction

Given the availability of tools, accepted process models, and the vast coverage of security incidents in the media, one might expect that developers would adopt secure practices as a matter of course [1]. Surprisingly, however, many professional software developers do not consistently and comprehensively make use of security tools, models and practices. Why is this?

It may be that writing secure code requires effort be made by developers at every step in the development process. However, it may be that security in software development is also driven by intrinsic factors that can be supported through social interactions in the community and culture of software development.

Starting from this premise, the Motivating Jenny project[1] is conducting a series of ethnographic studies that build upon frameworks of personal motivation and team culture [2], [3].

The project aim is examining what security in practice is like in the professional world, with the aim to find ways to better support and engage developers with security [1]. The workshop described in this report is one example of how this project is meeting this aim, by developing a set of materials that are being refined for dissemination to practitioners working in the field.

II. Background

The ethnographic method is used to study peoples' actions and accounts of actions. The method allows researchers to develop understanding about what practitioners working in socio-technical environments do and why they do it [4]. Ethnography's distinguishing feature is that it allows researchers to consider experience from the perspective of the insider [5]. Motivating Jenny has an interest in understanding the point-of-view professional software developers have about security. A particular focus is given to "ordinary organisational insiders" [6], that is, to developers who are not specialists in security.

Ethnographic research often includes exploration of open ended questions that are used to guide individual study designs and to support interactions in the field. This paper opened with one such problem, noting the discrepancy between existing methods to secure software and adoption among practitioners, asking:

Why don't developers adopt secure practices and technologies as a matter of course?

In engaging with questions like these, researchers are able to develop and maintain a critical stance toward a topic [5]. In studies for the Motivating Jenny project, this entails listening to what developers say, but also considering how what developers say might be shaped by their interactions with the researchers, and with other aspects in their working environment, including other developers, the workplace, within the broader software development profession, and beyond.

The project seeks to understand how to motivate professional developers to engage with security in software engineering practice. The project does not intend to assess the quality or quantity of the security information developers possess. These two statements elide aspects inherent in conducting field studies in professional settings.

First, it is necessary to establish trust and to engage with professionals in a way that is non-judgmental. This is particularly important in investigations that involve sensitive concerns. In the same way that companies, and developers, don't want to be perceived as releasing buggy code [7], neither wants to be perceived as releasing insecure code.

Related to this, it is often necessary to quickly get under the skin of a topic during interactions in the field; access to professionals is difficult to get and is often sporadic or constrained [8]. In software engineering, this means it can be challenging to get past the codified knowledge [9] that professional developers are "supposed" to know. In the context of this research, early interactions confirmed that it is difficult to establish what developers think about security. In conversation, many developers are able to quickly name

[1]https://www.motivatingjenny.org

common vulnerabilities or to describe in broad terms aspects of threat modelling, but this doesn't reveal very much about their interest in security or how important they believe it is.

One way to explore a topic like security is to undertake activities that run parallel to studies conducted in the field. These activities provide opportunities to build up understanding about the topic under investigation [10]. They also serve as an additional, if informal source of information by which to confirm or refute meanings collected in other data [8]. Finally, they serve the practical purpose of raising fluency with the topic, making it easier for researchers to conduct studies in the field.

The workshop described in the following sections is an example of an activity that can run in parallel to field studies, and can be adapted for use with participants from field studies or taken into professional settings for independent use.

III. Design

The design for the workshop grew out of observations made in early interactions with practitioners in a meetup[2] and a preliminary study of conversation in an on-line Q&A environment [11]. It should be noted that the first interaction did not include formal data collection; the second was conducted with approval of the first author's university ethics committee.

Through these activities, conversations among developers about security were shown to include technical advice and guidance that include established practices and principles. They also include statements about personal values and attitudes like responsibility, trust, and fear. This point stood out: similar attitudes have been shown to determine or influence security behavior in the general public [12]. It seemed reasonable to assume that they also influence developers, and that finding ways to harness talk about values and attitudes might be a way to positively influence secure coding behaviour.

A. Aims and Objectives

This workshop was developed as a part of research that is examining the role motivation plays in the production of secure code and how practitioners can initiate and sustain a secure software culture. The workshop meets two needs for the larger research program.

First, as noted above, the workshop supports and strengthens our research activities in field sites [4]. It meets three research aims:

1) It increases fluency with the topic, better equipping the researchers to interact with developers in the field, and to "get under the skin" of security.
2) It provides a secondary source of information that can be compared with evidence gathered in formal studies.
3) It is a way evaluate the project's growing sense about "what security is" against theories and conceptual frameworks in fields other than software engineering.

Second, ethnography can be used to inform the design of software engineering tools and improve process development [4]. Though Motivating Jenny is not specifically focused on tools or process, one aim of the project is to develop guidelines and materials that professional developers can use to engage with security in practice. To this end, a series of objectives were defined for the workshop structure and content.

1) Encourage attendees to primarily engage with each other, in order to promote a more naturalistic environment, similar to those in which developers normally work.
2) Provide an opportunity for attendees to engage with scenarios drawn from actual security events.
3) Establish a space for non-confrontational interaction about security.

B. Related Work

The following sections describe related work that informed the workshop design.

1) Supporting interaction through talk: Unscheduled talk in the workplace is integral to software development. Conversations between developers include sharing war stories about past experiences but also provide a narrative for one another in the midst of practice, a "summing up" [13] that workers use to develop confidence, to circulate "community memory" and to learn [14]. Talk is used to generate understanding of what the software is and needs to be, and of what developers need and would like to make. This kind of "code talk" is often "snatched" or serendipitous, but lends structure to decisions about work that will be undertaken at the desk [15].

2) Security events in the media: The public sphere has been identified as one of the ways workers develop awareness of security [16]. In the current climate, security incidents are widely reported in media sources. It is also possible to find accounts of developers who have been affected by high-profile breaches. Like war stories [13], these personal accounts provide insight into how developers solve security related problems. They also give additional perspective about the far reaching impact of security incidents on developers and companies.

3) Taking a positive approach toward security: When developers talk about code, it is value laden and dynamic [15]. It is also supportive. The approach used was influenced, in part, by *The Envisioning Cards*, a set of cards and exercises designed to help designers think broadly about how technologies are used, and to consider their long-term effects on societies [17]. In taking a positive, value-oriented approach toward security [18], developers are permitted to identify what they believe is important in their work, a point that has connections with research in motivation [19]. The approach taken is to position security as a quality to be striven for [20].

IV. An Overview of the Workshop

Using recent tabletop security games [21], [22] as a guide, materials were developed that would evoke a sense of play and engage attendees [23]. The workshop is structured to last 90 minutes. Participants are divided into groups of 5 or 6 (see

[2]https://www.meetup.com/Extreme-Programmers-London/events/245075051/

group work in Figure 2). Activities are undertaken in three parts:

Part 1 Compromised Software (30:00) In this part, attendees read the report of the compromise given by HandBrake (read an overview of the incident in Section IV-B). A set of cards prompts discussion about different aspects of the security incident (see Figure 1), including stakeholders and impact of the incident.

Part 2 Another Point of View (30:00) In part two, each group works through a second version of the HandBrake incident. This version of the incident is an account told from the perspective of a developer (See Figure 1) who was directly affected by the compromise reported in part one. Prompting questions aid discussion about how perceptions about stakeholders and impact change when the focus of the incident is oriented toward a developer.

Part 3 Group Discussion (30:00) In Part 3, groups are brought together for facilitated discussion. Because this workshop is employed in different kinds of environments, the content of this section has been tailored to individual interactions.

A. Instructions

The instructions given for parts one and two are the same:

Open the envelope marked Part1/Part 2. Inside, you will find a report of a security incident and a set of cards.

- Use 10 minutes to read the reports.
- Spend 15 minutes discussing the questions on the cards in relation to this story.
- For the last 5 minutes, make a note about two or three points that stood out in discussion. Include notes about why each point stood out.

As you work through each part, take notes about key points on sheets of paper or directly on the cards themselves (for examples of notetaking, see Figures 3 and 4).

B. The Incident: HandBrake

In May 2017, attackers compromised a download server for the open-source media-encoding software HandBrake. An infected version of the software was placed on the compromised server that included the malicious software Proton. Once installed on Mac computers, Proton allows unauthorized access to the affected machine [3]. Reports were collected about this incident from general news websites, technical news websites, news aggregators, and the notice of the compromise posted by the software company[4]. An account of a software developer working at Panic Software who infected his computer with the compromised software[5] was also collected to provide a perspective on the story to which participants could more directly relate. For representative screenshots of these reports, see Figure 1.

[3]https://www.symantec.com/security-center/writeup/2017-050811-5239-99

[4]https://forum.handbrake.fr/viewtopic.php?f=33&t=36364

[5]https://panic.com/blog/stolen-source-code/

TABLE I
FIRST SET OF VALUES, ADAPTED FROM [24]

Value	Description
Trust	Expectations and confidence in the reliability of social interactions with others, for example, good will.
Privacy	Right of an individual to determine what information be communicated to others.
Autonomy	The ability to act in ways that will help one achieve goals.
Ownership	A right to possess information, use it, manage it, & bequeath it.
Human Welfare	Physical, material and psychological well-being.
Fairness	Impartial and just treatment or without favouritism or discrimination.
Accessible	Ensuring all people are able to access information technology.
Informed Consent	Permission for something to happen, granted in full knowledge of the possible consequences.
Accountable	Properties ensuring that actions of a person, people or institution are able to be uniquely traced, explained or understood.
Identity	A persons understanding of who he or she is over time, embracing continuity and discontinuity.

C. Value Cards

Part two includes prompting cards and also a set of cards that name a series of values with suggested definitions. The prompting cards ask attendees to consider the perspective of the developer affected by the compromise in light of the values.

Two sets of values have been adapted and trialled, including one set drawn from value-centred design (See Table I) and one from social psychology (See Table II).

V. DISCUSSION

The workshop has been run with slight variations three times. This section gives a synopsis of each event, with notes about variations.

A. Summer 2018: Practitioner Conference

The workshop was first given in a 75 minute session at a practitioner conference in London, UK. Attended by between 30 and 35 people, goals for this event were to trial the work shop design, and to define the security dimensions. Due to constraints in the room, attendees worked in groups of between 6 and 8 people.

Each group was given a report from different media sources for Part 1. Each account reported basic details about the incident, but provided different kinds of background information and different commentary. In Part 2, each group was given the same account of the compromise at Panic software. The first set of values was used to support discussion for this part (See Table I).

Part 3 was used to talk over a story about a security incident told by a participant in the group. Other participants were

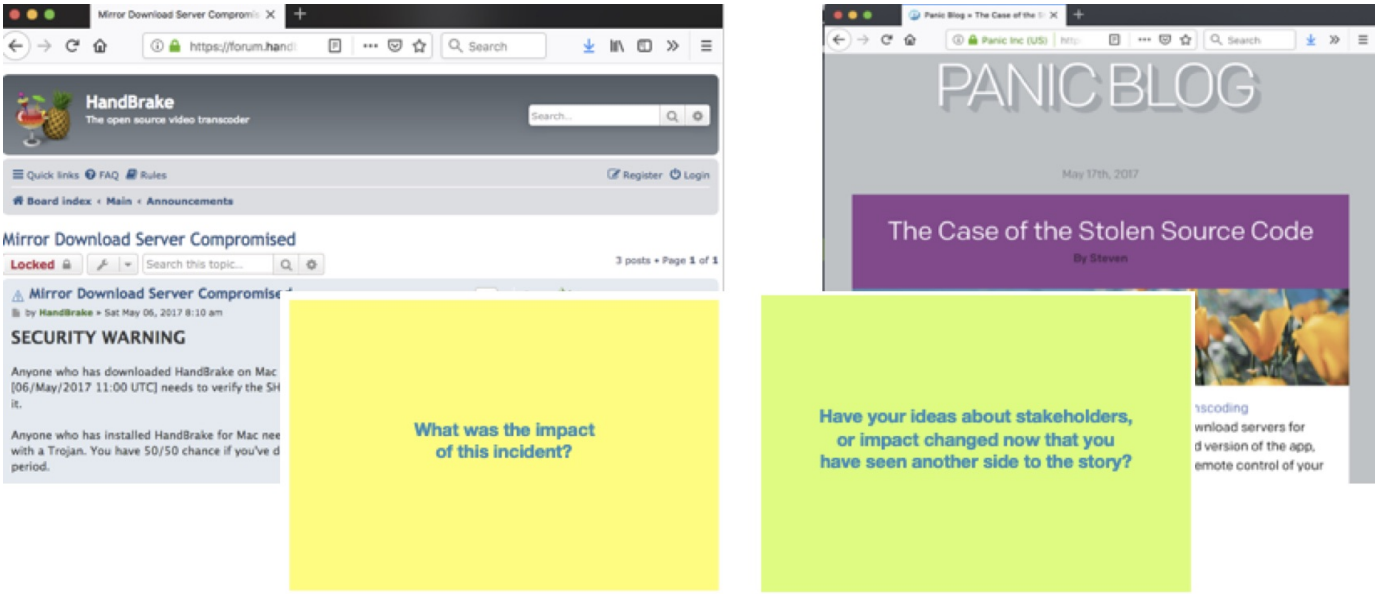

Fig. 1. Stories and Cards

Fig. 2. Group Work

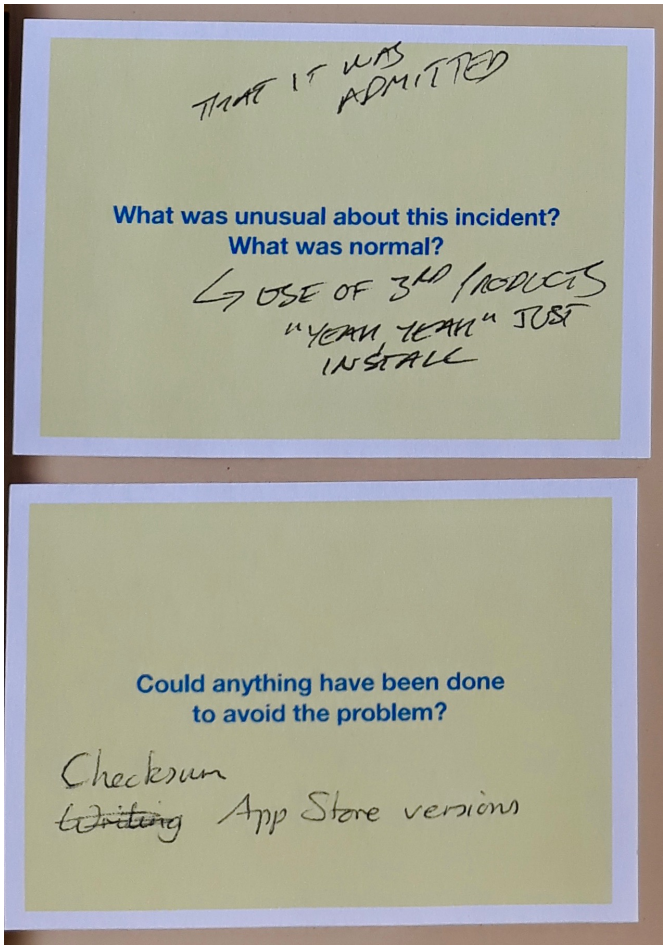

Fig. 3. Notes on cards

invited to use a prompting question or a value card from Part One or Part Two within the discussion.

In running the session, the materials and instructions worked well, however there was not enough time in the schedule to permit a full discussion in part 3. In spite of this, written feedback provided to conference organisers after the event was overwhelmingly positive. 17 attendees provided conference organisers with written feedback after the session. Ranked along a scale with five indicating an Excellent session, all respondents found the session to be either a 4 or 5. Likewise, feedback indicated that the respondents felt that they learned a lot, and that the session was well led, a point we to took as affirmation that the design of the workshop was effective.

Here are some positive comments from the feedback:

Great team discussion!

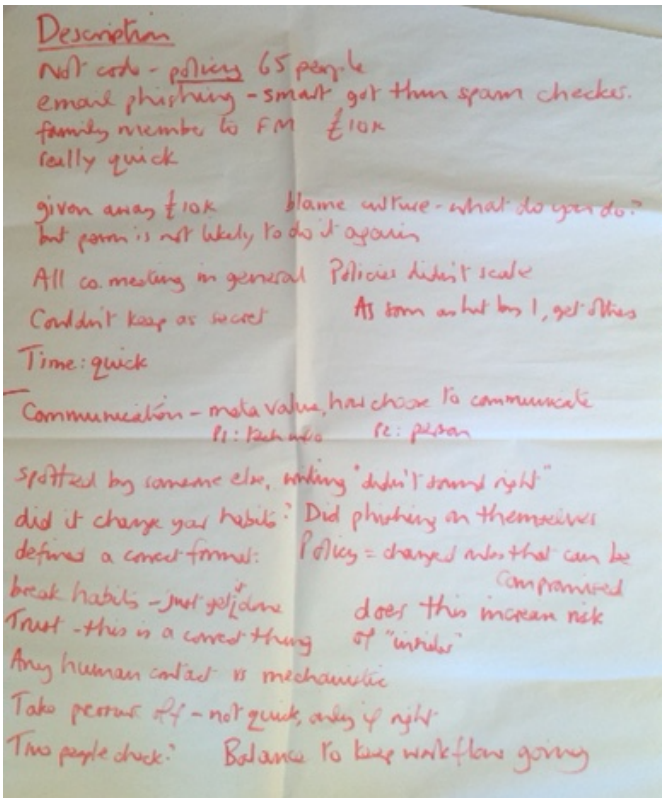

Fig. 4. Notes from a session

TABLE II
SECOND SET OF VALUES, ADAPTED FROM [25]

Value	Description
Self-direction	Freedom to cultivate ones own ideas and abilities. Freedom to determine ones own actions.
Safety	Safety in ones immediate environment. Safety and stability in the wider society.
Conformity	Compliance with rules, laws and formal obligations. Avoidance of upsetting or harming other people.
Benevolence	Being a reliable and trustworthy member of the group. Devotion to the welfare of group members.
Universalism	Commitment to equality, justice and protection for all. Acceptance and understanding of differences in people.
Power	Exercising control over people. Control of material and social resources.
Stimulation	Excitement, novelty and change.
Achievement	Success according to social standards.
Tradition	Maintaining and preserving cultural traditions.
Humility	Recognising ones insignificance in the larger scheme.
Hedonism	Pleasure and sensuous gratification.
Face	Maintaining ones public image and avoiding humiliation.

I liked the way the two parts were from different perspectives.

A refreshingly different look at security issues in software.

Really enjoyable, thought provoking and participatory! Thank you!

The feedback also included critical comments. Several attendees noted that the session was too short to allow for comprehensive group discussion. One person noted that how the value cards were to be used was not clear. We used these points to refine the second and third events.

B. Autumn 2018: Field Site

With this workshop, the primary aim was to observe how the participants from one project field site interact with one another when talking about security. Prior observations of group work had taken place during everyday practice in a large open plan office. Though a few instances of practice were observed that included security elements, this was the first time developers at this company were observed openly conversing about security. This session was also used to clarify findings generated in earlier visits, and to elicit further comments about the attendees' experiences and attitudes.

The workshop was attended by 6 developers, one tester that had been interviewed and one developer new to the company. The attendees worked in groups of four. The second set of values was used (See Table II). In Part 3, a group discussion was facilitated. The questions for Part 3 are given below:

1) Is talking about security incidents in this way helpful?
   - What did you like?
   - What didn't make sense?
   - Are there other approaches you have used that were helpful?
   - How were they like or different than this exercise?
2) Does a consideration of stakeholders, and impact come into software development at this company? How often, what prompts it?
3) How about a consideration of values in the context of security? Does it come up? When?
4) Can you think of recent examples from your experience at this company where similar kinds of talk have taken place?
   - If so, where was that? What context, e.g. a meeting or in the kitchen?
5) Could an approach like this be used at this company if so where, how?
6) Who is responsible for security in code?
   - What role do/can developers play in this?

As in the first session, attendees at this field site were engaged and focused in the activities for the workshop. Feedback gathered in the third part has not been fully analyzed, however,

the response to the first question, *Is talking about security incidents in this way helpful?* was a pronounced "Yes."

C. Autumn 2018: Seminar form

For this invited 6o minute talk, the workshop was presented to a room in which participants were sitting in groups of two or three at small tables. Approximately 30 people attended. In this environment, the second set of values was used (See Table II). Each table was given a random selection of three or four value cards from the set of 12.

Drawing on elements of the case-based [26] and peer instruction [27] teaching methods, the accounts of the software compromise were presented to the room as a case. After presenting the incident, attendees were asked to give a show of hands for these questions:

- Is this incident unusual?
- Is it likely to happen again?

Following this, attendees were asked to discuss with each other who might be affected by a compromise of this type. Impressions were shared afterward around the room. After presenting the developer-centred version of the incident, attendees were asked to discuss with their partners how the values they found on their table figured into the case.

After a period of five minutes, attendees were asked to volunteer information for the following questions:

- Which of the values from the table was the most important? For whom?
- When did this value come to the fore? Before the incident, after, or long after?

For the third part of the discussion, the session leaders facilitated a brief discussion around the room around the following two points:

- Security should be handled through tooling.
- Developers should be thinking about security all the time.

VI. Lessons Learned

It has been suggested that developers need to be taught new attitudes toward security that will encompass technical knowledge and security analysis, but also skills in communicating about security issues beyond engineering teams. What is needed are "engaging" interventions that will appeal to programmers [28].

Experiences to date with this workshop suggest that a positive approach that connects security to developer experiences is useful in engaging professionals in discussion, but there are some areas that can be improved.

A. Developers engage with personal stories.

Talking around two perspectives for a single security incident is an effective way to strike a balance in discussion between technical and security detail, and personal values. The "Panic" story is close enough to the developers' own experience to engage them, but not close enough to inhibit participation. The workshop has been using reports printed in full from the internet. While this worked in the first two sessions, we have now gathered enough information about how developers engage with these sources to refine the stories into shorter cases.

B. Interaction through play is effective.

Participants enjoy the physical aspects of the game, including setting the timers and working with cards. Though the sets of prompting cards have been effective in the workshops, informal feedback suggests that there are too many questions for each section. Likewise, asking only two or three questions per section worked well in the seminar. Going forward, the sets of questions for each part will be refined and streamlined.

C. Values support conversation.

Informal feedback suggests that developers like taking a positive, value-oriented approach toward security. Two sets of values drawn from different fields of research have been trialled. The workshops in which the individual sets were used were similar in character; attendees did not appear to have difficulty in talking about the values in either set. However, it is not clear exactly what role the values play in the process, particularly in relation to the incident that was used. Formal evaluation of the workshop must be made to clarify these points.

D. Public incidents facilitate information trading.

Informal observation suggests that different kinds of information are traded among developers in the midst of these conversations. Developers were observed to expand on technical information included in the reports, providing additional scenarios and examples from personal experience, but also drawing in terminology associated with the security mindset including threats, attacks and technology specific security facts. Fuller analysis of the workshop data gathered in the field site to catalog information trading is underway.

VII. Conclusions

The question of why professionals don't adopt secure practices and technologies as a matter of course remains open, and is a part of continuing investigations in the Motivating Jenny project. However, with this workshop, the project has identified ways to support professionals in talking about security.

The workshop developed here includes a set of working materials that can be employed with practitioners in a variety of settings. It uses narrative and storytelling to connect developers with security incidents, and to encourage talk about security implications and impacts.

These activities have a place in professional settings. In structuring activities around an incident taken from the public sphere, professionals are shown techniques for critical engagement with sources that are known to influence security awareness. The non-confrontational space for talk about security stands to positively affect security problem solving, confidence and knowledge.

Looking forward, the materials will continue to be used and developed within the project to support interactions with developers in field sites and community engagements. Several additional research and practical aims have been identified:

- Investigate in more detail what values bring to talk within security discussions. The set depicted in Table II will be explored in more detail, as they more closely reflect qualities related to software developer characteristics and motivation.
- Conduct a formal evaluation of the materials. This will begin with a structured examination of feedback given from the first engagement, and an analysis of workshop data gathered as a part of the field study.
- Develop the materials into a set of cards for production and dissemination.

ACKNOWLEDGMENT

We thank the professional developers who participated in our workshops. The work was supported by the National Cyber Security Centre (NCSC). Nuseibeh thanks SFI, EPSRC and ERC for financial support.

REFERENCES

[1] C. Weir, A. Rashid, and J. Noble, "How to improve the security skills of mobile app developers? Comparing and contrasting expert views," in *Twelfth Symposium on Usable Privacy and Security (${$SOUPS$}$ 2016)*, 2016.

[2] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, "Motivation in Software Engineering: A systematic literature review," *Information and software technology*, vol. 50, no. 9, pp. 860–878, 2008.

[3] H. Sharp, H. Robinson, and M. Woodman, "Software engineering: community and culture," *IEEE Software*, vol. 17, no. 1, pp. 40–47, Jan. 2000.

[4] H. Sharp, Y. Dittrich, and C. R. B. d. Souza, "The Role of Ethnographic Studies in Empirical Software Engineering," *IEEE Transactions on Software Engineering*, vol. 42, no. 8, pp. 786–804, Aug. 2016.

[5] M. Hammersley and P. Atkinson, *Ethnography: Principles in practice*, 3rd ed. Routledge, 2007.

[6] C. Posey, T. L. Roberts, P. B. Lowry, and R. T. Hightower, "Bridging the divide: A qualitative comparison of information security thought patterns between information security professionals and ordinary organizational insiders," *Information & Management*, vol. 51, no. 5, pp. 551–567, Jul. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378720614000421

[7] D. Perry, "Where Do Most Software Flaws Come From?" in *Making Software: What Really Works, and Why We Believe It*, A. Oram and G. Wilson, Eds. O'Reilly Media, Inc., 2010, pp. 453–494.

[8] S. Easterbrook, J. Singer, M. Storey, and D. Damian, "Selecting empirical methods for software engineering research," *Guide to advanced empirical software engineering*, pp. 285–311, 2008.

[9] M. Eraut, "Non-formal learning and tacit knowledge in professional work," *British Journal of Educational Psychology*, vol. 70, no. 1, pp. 113–136, 2000.

[10] B. Anderson, "Work , Ethnography and System Design," in *The Encyclopedia of Microcomputers*, A. Kent and J. G. Williams, Eds. Marcel Dekker, 1997, vol. 20, pp. 159–183.

[11] T. Lopez, T. T. Tun, A. Bandara, M. Levine, B. Nuseibeh, and H. Sharp, "An Investigation of Security Conversations in Stack Overflow: Perceptions of Security and Community Involvement," in *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, ser. SEAD '18. New York, NY, USA: ACM, 2018, pp. 26–32.

[12] L. Coventry, P. Briggs, J. Blythe, and M. Tran, "Using behavioural insights to improve the publics use of cyber security best practices," Government Office for Science, Tech. Rep., 2014.

[13] J. E. Orr, "Narratives at work: Story telling as cooperative diagnostic activity," in *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*. ACM, 1986, pp. 62–72.

[14] P. Duguid, "What talking about machines tells us," *Organization Studies*, vol. 27, no. 12, pp. 1794–1804, 2006.

[15] A. Higgins, "'Code talk' in soft work," *Ethnography*, vol. 8, no. 4, pp. 467–484, Dec. 2007.

[16] S. Furnell and K.-L. Thomson, "From culture to disobedience: Recognising the varying user acceptance of IT security," *Computer Fraud & Security*, vol. 2009, no. 2, pp. 5–10, Feb. 2009.

[17] B. Friedman and D. Hendry, "The Envisioning Cards: A Toolkit for Catalyzing Humanistic and Technical Imaginations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1145–1148.

[18] P. Roe, "The value of positive security," *Review of International Studies*, vol. 34, no. 4, pp. 777–794, Oct. 2008.

[19] H. Sharp, N. Baddoo, S. Beecham, T. Hall, and H. Robinson, "Models of motivation in software engineering," *Information and software technology*, vol. 51, no. 1, pp. 219–233, 2009.

[20] B. McSweeney, *Security, identity and interests: a sociology of international relations*. Cambridge University Press, 1999, vol. 69.

[21] T. Denning, B. Friedman, and T. Kohno, "University of Washington," 2013. [Online]. Available: https://securitycards.cs.washington.edu/

[22] B. Shreeve and A. Rashid, "The University of Bristol Cyber Security Group." [Online]. Available: https://sites.google.com/view/decisions-disruptions/

[23] M. Gondree, Z. N. Peterson, and T. Denning, "Security through play," *IEEE Security & Privacy*, no. 3, pp. 64–67, 2013.

[24] B. Friedman, P. H. Kahn, A. Borning, and A. Huldtgren, "Value Sensitive Design and Information Systems," in *Early engagement and new technologies: Opening up the laboratory*, ser. Philosophy of Engineering and Technology. Springer, Dordrecht, 2013, pp. 55–95.

[25] S. H. Schwartz, J. Cieciuch, M. Vecchione, E. Davidov, R. Fischer, C. Beierlein, A. Ramos, M. Verkasalo, J.-E. Lnnqvist, K. Demirutku, O. Dirilen-Gumus, and M. Konty, "Refining the theory of basic individual values." *Journal of Personality and Social Psychology*, vol. 103, no. 4, pp. 663–688, 2012.

[26] A. Yadav, M. Vinh, G. M. Shaver, P. Meckl, and S. Firebaugh, "Case-based instruction: Improving students' conceptual understanding through cases in a mechanical engineering course," *Journal of Research in Science Teaching*, vol. 51, no. 5, pp. 659–677, 2014.

[27] C. H. Crouch and E. Mazur, "Peer Instruction: Ten years of experience and results," *American Journal of Physics*, vol. 69, no. 9, pp. 970–977, Aug. 2001.

[28] C. Weir, A. Rashid, and J. Noble, "Reaching the masses: A new subdiscipline of app programmer education," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2016, pp. 936–939.