Open Research Online

The Open University's repository of research publications
and other research outputs

Security Thinking in Online Freelance Software Development

Conference or Workshop Item

How to cite:

Rauf, Irum; Petre, Marian; Tun, Thein; Lopez, Tamara and Nuseibeh, Bashar (2023). Security Thinking in Online Freelance Software Development. In: IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), 14-20 May 2023, Melbourne, Australia.

For guidance on citations see FAQs.

© [not recorded]

https://creativecommons.org/licenses/by-nc-nd/4.0/

Version: Accepted Manuscript

Link(s) to article on publisher's website:
https://conf.researchr.org/home/icse-2023

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data policy on reuse of materials please consult the policies page.

oro.open.ac.uk

Security Thinking in Online Freelance Software Development

Irum Rauf, Marian Petre,
Thein Tun, Tamara Lopez
The Open University, Milton Keynes, UK
firstname.lastname@open.ac.uk

Bashar Nuseibeh
The Open University, UK
Lero, Republic of Ireland
firstname.lastname@open.ac.uk

***Abstract*—Online freelance software development (OFSD) is a significant part of the software industry and is a thriving online economy; a recent survey by Stack Overflow reported that nearly 15% are independent contractors, freelancers, or self-employed. Although security is an important quality requirement for the social sustainability of software, existing studies have shown differences in the way security issues are handled by developers working in OFSD compared to those working in organisational environments. This paper investigates the security culture of OFSD developers, and identifies significant themes in how security is conceived, practiced, and compensated. Based on in-depth interviews with 20 freelance (FL) developers, we report that (a) security thinking is evident in descriptions of their work, (b) security thinking manifests in different ways within OFSD practice, and (c) the dynamics of the freelance development ecosystem influence financial investment in secure development. Our findings help to understand the reasons why insecure software development is evident in freelance development, and they contribute toward developing security interventions that are tailored to the needs of freelance software developers.**

***General Summary*– Online freelance software development (OFSD) is a significant part of the software industry and is a thriving online economy. Although security is an important quality requirement for the social sustainability of software, existing studies have shown differences in the way security issues are handled by developers working in OFSD compared to those working in organisational environments. Based on in-depth interviews with 20 freelance developers, this paper investigates the security culture of OFSD developers, and identifies significant themes in how security is conceived, practiced, and compensated.**

***Index Terms*—freelance software development, payment for security, security, developer, software development in society, societal challenges of secure software development**

I. Introduction

Professional software developers work in environments characterised by factors such as geographic location, organisational culture, and employment status. In the 2022 Stack Overflow survey of over 70,000 developers worldwide, nearly 15% of developers were independent contractors, freelancers, or self-employed [1]. With a 5% increase since the previous year, and with a sharp rise in demand during the Covid pandemic period [2], online freelance software development (OFSD) is now a significant part of the software industry. OSFD is performed by self-employed workers who have a task-based association with the employer [3], often made through online commercial platforms such as *Toptal* and *Upwork*.

Software development is considered to be among the most demanding jobs in the online freelance community [4]. Although most of the highly-skilled workforce – including software developers – in online freelance communities comes from developing countries, the demand is global; SMEs and individual contractors from advanced economies often outsource relatively small to medium projects and tasks to developing countries in order to benefit from a lower-paid and skilled workforce [4]. Freelance software developers work in diverse socio-economic [5] and socio-technical [6] environments – a diversity that is at the heart of a range of innovations in the digital economy. Given the general shift toward a self-employed work model through freelancing [7], and the growing recognition that software vendors share responsibility with developers [8] for countering cyber-attacks, understanding how security is handled within software produced by the OFSD community has increasing societal importance.

Security is identified as an important quality requirement for the social sustainability of software [9]. Developer-centered security focuses on understanding how developers write secure code and facilitates the development of tools and techniques for the production of secure code [10]. However, security in freelance software development has received little attention. Existing work on security behavior of freelance developers [11], [12] and on understanding security in the freelance development ecosystem [13] notes that freelance software developers produce more insecure code and holds them accountable for it [13].

Existing studies have reported that freelance software developers often fail to pay full attention to security issues in their code (e.g., [11], [14], [12] – discussed more fully in Related Work). However, while the OFSD population is identified as less attentive to security in code, the different circumstances under which freelance developers produce code, compared to employee developers, and nuances in their security thinking are less well explored. Following the premise that developers' approaches to software security depend on their culture, priorities, values and practices [15], our research questions in this work are:

- How do freelance developers think about security in their code? Is security a concern they address or not? Why?
- How do individual freelance developers operationalise their security thinking in practice? Do they have common meth-

ods/strategies for handling security?
- How do freelance developers expect to be compensated for their effort in improving security?

To address these questions, we conducted video-conference interviews with 20 online freelance software developers and a thematic analysis [16] focusing on how security is understood and handled in OFSD.

II. Related Work

This section reviews existing empirical studies of the freelance software community (clients and/or developers) that reported secure coding.

A. Characterising freelancers

Freelancers work outside the commitment of a full-time office worker, and can be seen as having the freedom to choose their job, but little or no job security. Shevchuk and Strebkov [17] identify different types of freelancers based on their working patterns: *genuine freelancers* who work exclusively as freelancers, *moonlighters* who also hold regular jobs, and *entrepreneurs* who also run small businesses. Their work studies value-shifts in freelancers based on their working patterns. Sison and Lavilles [18] identify different stages of online freelance software development which include *noob* (i.e., beginners who are starting to venture in OFSD), *rockstar* (i.e., an experienced online freelance software developer), and *super-rockstar* (i.e., those who maximize the available opportunities and offer work to others and gain profit from such offerings). In another paper, Lavilles and Sison [19] report six main concerns of online freelance software developers in the Philippines, which include "uncertainty and transitions, trust and work agreements, reputation and client relationships, accomplishing tasks, platforms and software support, and work practices" (p.6). These working patterns, work stages and concerns of online freelance software developers clearly distinguish the demographics of *freelance* developers from those of *employee* developers.

B. Security in freelance development

Security is often neglected if not reflected properly "in the communicative and collaborative structures of the organization" (p.1, [20]). Haq et al.'s [21] survey of 162 clients from freelancing platforms showed that security has the highest impact on client satisfaction – but did not report whether freelance developers are (made) aware of this.

Studies on how freelance software developers address security looked at how developers code and how they review code. Naiakshina et al. [11] and Bau et al. [12] studied how developers write code. Naiakshina et al. [11] hired 49 freelance developers for a web development project and reported that "a number of freelance developers did not feel responsible for security" (p.1) and did not attend to security. Their findings showed that freelance developers tended not to store passwords securely until they were asked to do so explicitly. They also reported that payment levels had no significant effect on the security of code. This confirmed findings reported earlier by Bau et al. [12], which highlighted that the price ranges of project-completion had no significant effect on projects' vulnerability rates. Freelancers, compared to startups, produced significantly more vulnerable code and tended to be less reliable in delivering projects on time [12].

Some studies (e.g., [22], [14] and [23]) asked developers to review code. Edmunson et al. [22] recruited 30 freelance developers to conduct a manual security review of a small web application with the aim of identifying metrics useful in predicting effectiveness of the reviewers. The study did not find any significance related to developers' years of experience, education, or opinion of how well they performed on the code reviews.

Danilova et al. [14] recruited freelancers to review password-storage code snippets. The study reported similar behavior to Naiakshine et al. [11], i.e., most of the freelance developers did not report security issues with password storage. The participants who reported security issues were prompted to think about security. When participants were asked explicitly if they felt responsible for end-users' security when writing or reviewing code, most agreed that they did, but the evaluation of their code review report showed that they did not notice security issues in vulnerable code snippets.

Rauf et al. [23] conducted an empirical study with 124 freelance developers to investigate secure engagement when participants reviewed code. The study found that freelance developers think about security when engaging with code, but that each thinks differently from others.

Inspired by this prior work, the study reported here interviewed participants in order to understand the freelance software developers' perspectives on security and to identify evidence and characteristics of security thinking, that is, of their attitudes and practices regarding secure coding.

III. The Study

The study, approved by the OU's ethics committee, involved semi-structured interviews with 20 active online freelance software developers. Interviews are often used to study developers' practices and opinions [24]. For example, Xie et al. [25] interviewed "15 professional software developers to understand their perceptions and behaviors related to software security" (p.1), and Balebako et al. [26] conducted "a series of interviews with 13 app developers to obtain rich qualitative information about privacy and security decision-making" (p.1). The study was coordinated and conducted online using tools such as email, Qualtrics (www.qualtrics.com) and Zoom (https://zoom.us) and consisted of a brief (5-minute) background questionnaire, and a 30- to 45-minute interview.

The study was conducted in three steps. First, those who expressed interest in participating in the study were sent a link to the information sheet and consent form. Second, participants who completed the consent form were sent a link to a 5-minute background questionnaire (concerning programming experience, work background and general demographics) along with time-slots for the interview. Third, a one-to-one online interview was conducted with each of the participants.

All the interviews were audio-recorded using Zoom and stored anonymously on secure servers. Participation in the study was voluntary. Each participant was given a £10 Amazon voucher, and entered in a prize draw for an additional £50 Amazon voucher for one participant.

A. Interview structure

At the outset, we encouraged participants to think of the last project they completed and delivered, and to answer every question first thinking about that project and then whether this is what their general work practice is in other projects. Encouraging the participant to think about a specific project first served as an anchor point to help them recall their working practices. We started with general questions about the types of projects on which the participant works and the software qualities the developer prioritizes. We enquired about how the developer generally negotiated with clients. This was followed by more specific questions on how security is handled in projects and in interactions with the client.

The questions on each topic were structured using the funnel technique [27], which offers a general-to-specific approach for acquiring information from the interviewee in a systematic manner. The interview follows a sequence of questions for each topic: "*Open questions* are used to obtain a general description of the expert's approach to the task. They are followed by *probing questions* which delve into a specific section of the experts description. If necessary, the knowledge engineer elicits even more specific information by using *closed questions* to verify and clarify the knowledge base. The information is summarized and a transition moves the interview to the next subject area." (p. 34) The intention is to exhaust each area of enquiry before moving on to the next area.

The interview protocol is available online at: http://bit.ly/3YxLcdS All of the participants were debriefed at the end of the interview regarding our interest in understanding security culture in the freelance software development community.

B. Recruitment

We aimed to recruit 20 freelance software developers for the 30- to 45-minute interviews online. We considered this a reasonable sample size. First, as found by Guest et al. [28], data saturation is typically reached within the first 12 interviews in qualitative research. Second, longer interviews are impractical because freelance developers consider performing online jobs as a source of income [29]. Finally, analysing recorded interviews that are significantly longer than 900 minutes becomes prohibitive. We recruited participants via convenience sampling [30]. The study was advertised through social networks and personal contacts, including relevant professional groups on LinkedIn and Facebook. Snowballing was also used; participants were requested to ask other freelancers to participate without disclosing interview details.

Participants were told that the aim of the interview was to understand how software development is carried out in OFSD communities. Participants were not informed *a priori* that the interview sought to understand their security thinking and practices. This provided an unprompted view of how security manifests in their conversations. However, if participants did not mention security in the first 15 minutes, they were asked explicitly if security is a concern for them.

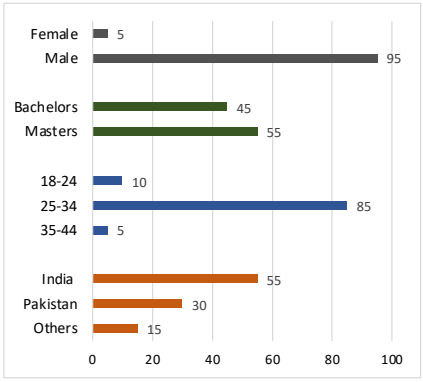

Fig. 1. Demographics - top: Gender, top (2nd): Education, bottom (2nd): Age, bottom: Location - All values are Percentages

C. Demographics

As shown in Figure 1, participants were predominantly male (95%), and all the participants had graduate or post-graduate degrees. Participants were primarily between 25 and 34 years old. Geographically, the participants were mostly South Asians. Other countries included Egypt, Kyrgyzstan and France. These demographics are in line with the profile of freelancers identified earlier in the literature with respect to sex [4], age [31] and geography [4].

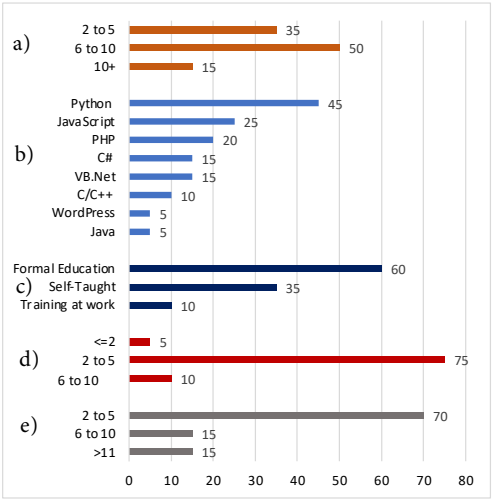

Fig. 2. Developer Profile (Part 1) (a) Years Programming (b) Programming Language (c) Learning Background (d) Year Freelancing (e) FL Projects - All values are Percentages

Figures 2 and 3 show that the majority of the participants are highly experienced developers; Python and JavaScript are their primary programming languages. All were educated formally and had been working freelance for 2 to 5 years, and they had worked on 2 to 5 freelance projects in the previous

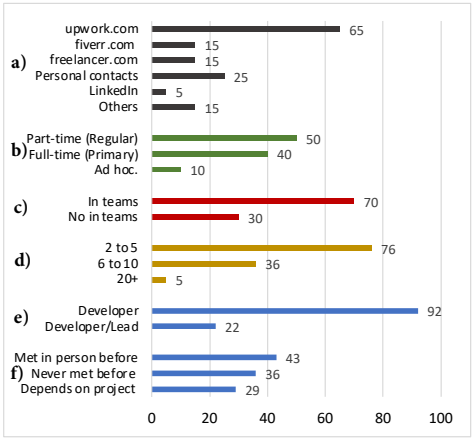

Fig. 3. Developer Profile (Part 2) (a) Freelance Platform (b) Income (c) Team (d) Team Size (e) Role in team (f) Knowing team members - All values are Percentages

year. upwork.com was their most-frequent freelancing platform. The majority regarded income from freelance work as either a primary or regular source of income. We consider this an indication that the participants took their freelance projects seriously. Most participants had worked in teams, mostly small teams. Almost half had worked with developers they had met in person before. Overall, the participant demographics represent a range of experience and backgrounds.

D. Work Profile

Figure 4 shows that participants' projects were mainly web and app development. Most participants identified functionality, efficiency and usability among the most important qualities of the software they develop.

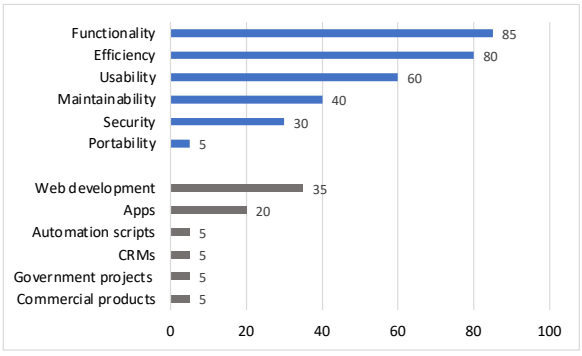

Fig. 4. Preferred Quality Attributes (top), Types of Projects (bottom)

IV. Analysis of Security Themes in OFSD

We conducted thematic analysis [16] of the open responses using Atlas.ti (https://atlasti.com). All 20 interviews were first transcribed in Microsoft Word and then checked and corrected by the first author, who also carried out all the interviews, and was therefore well-prepared to analyse the transcripts qualitatively, as suggested by Braun and Clarke [32]. In the initial phase of analysis, the interview transcripts were read repeatedly in an active manner, i.e., text was coded and re-coded as the understanding of the text became clear. Initial themes were identified and refined. Themes were then grouped semantically. The analysis was reviewed periodically by one of the co-authors working independently. She looked at the categorised data (returning to the raw data as necessary) and raised any concerns, which were discussed and resolved before the next iteration of the analysis.

After this inductive analysis, all of the authors considered what the identified themes revealed with respect to the three research questions. That meta-grouping was then used as a structuring device for reporting the findings. Hence, the three sections that follow align with the three research questions, although the underlying findings were reached inductively.

V. Presence of Concern for Security

This theme investigates whether security is a concern for freelance developers, and do they act on it? In cases where security is recognised as an explicit concern, we identified the possible reasons for the developers' concern, the situations in which security was treated as low-priority despite their concern for it, and the ways in which they attended to security if they considered it a concern. In cases where participants said explicitly that security was *not* a concern for them, we examined the possible reasons for that too. Fig. 5 summarises this theme and its sub-themes.

A. 'Security is a concern'

15 of the 20 participants stated explicitly that security concerns arise in their work. Without being prompted by the interviewer, 9 participants recounted that they brought up security and code quality when negotiating with clients. The other 6 participants (of the 15) were asked about security when they did not mention it during the first 15 minutes of the interviews. They stated that security is a concern for them in their projects and discussed how they address it. The 5 participants who said explicitly that they are *not* concerned about security are discussed in sub-section B.

1) Reasons for security concern among FL developers: Participants spoke about different reasons that motivate them to take security seriously.

Internal Motivation: Some developers are motivated by their concern for users of the applications (e.g., *FL1*) while others previously had a bad experience with insecure software they developed (e.g., *FL15*). *FL1* showed concern for his users saying: "*So I'm also thinking what happens to my users...when they will start using the platform? So when we are designing the architecture at that time, we give a huge priority to the security. It is the main concern, and it is a strict instruction to our quality assurance in testing team that you need to try to break the things as much as possible*." *FL15* previously had a DDoS attack on his application and had to troubleshoot and fix the security loophole. Due to this, he said, he attends to security in his software projects.

External Requirements: Some developers showed concern for security due to the requirements set out by their clients

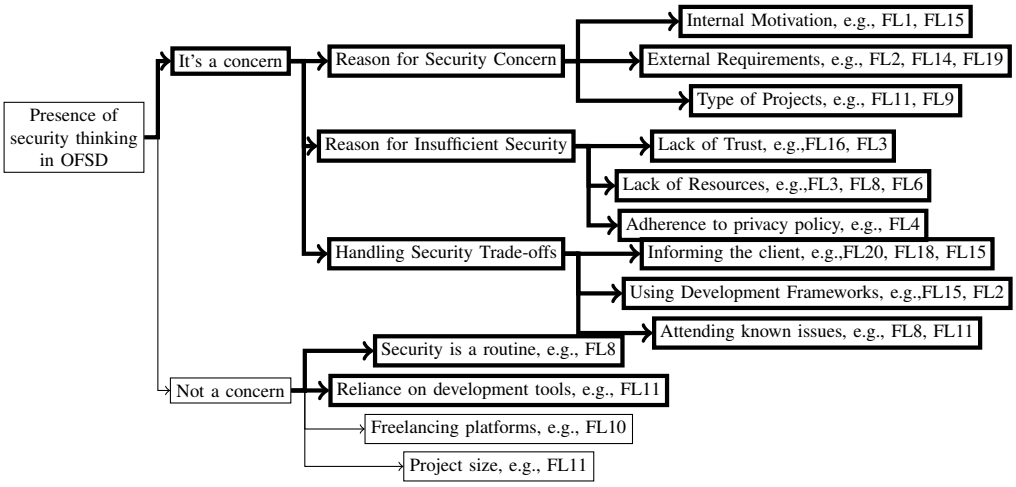

Fig. 5. Theme 1: Presence of concern for security in OFSD: Bold nodes show evidence of explicit or implicit security concern; non-bold nodes show absence of security concern. Some developers who responded that security is "not a concern" then elaborated with elements of practice that reflected security thinking.

(e.g., *FL2*, *FL14*, *FL18*, *FL19*). Others paid attention to security because of app store policies. *FL18* said: "*Security is the main thing, like we do prioritise it, so we are never going to compromise security – and even we can't, because see whenever you have to upload any app on Android Store or App Store, so the rules and regulations are also strict nowadays. So we must follow that rules*." *FL12* considers security an important concern, because he works on medical apps that need to be HIPAA-compliant to be in app stores. On the other hand, *FL19* works in government projects in which independent auditors are hired to ensure projects comply with different standards and security requirements. The project *FL2* worked on also has independent code reviewers who may identify security loopholes.

Type of Project: Some participants attend to security if they are working on security-critical projects such as antivirus projects (*FL11*) or projects that have large financial penalties if the system is compromised (*FL9*). Both *FL11* and *FL12* said that security issues are not a concern for small projects and noted that clients start thinking about security only when projects have large numbers of users (millions).

2) Reasons for insufficient security: Participants identified various reasons why security measures turn out to be insufficient, including lack of trust by the clients, their unwillingness to pay the time and financial costs, and their apparent lack of concern for data privacy.

Lack of Trust: Participants often feel a lack of trust especially when communicating with clients who may not be aware of security risks and implications. *FL16* pointed out that non-technical clients do not talk about security issues and *FL3* said: "*We let them know that it's insecure in this many ways. And if the [non-technical] clients find this useful, they can understand it. But they do not often agree with us, because they [think] we are finding ways for earning money*."

Lack of Resources: Many participants considered lack of money and time as important reasons why they often have to compromise on security. Some participants stated that security requires extra effort (e.g., *FL3*, *FL8*, *FL6*), and hence it costs more – which clients are often unwilling to pay (e.g., *FL3*, *FL13*, *FL16*, *FL9*). *FL10* believed that clients' priorities are time and money, which can compromise quality. *FL10* compared his full-time office job to freelance projects, saying that he has more time in his office job to write good code. On the other hand, if he has an approaching deadline in his freelance project, then he "*[doesn't] really see or care about the structure of the code*". Similarly, *FL3* said that security needs more time but can be ignored in rapid development, as it is not the main concern of the client.

Adherence to privacy policy: Many participants mentioned that their advice on security is often not heeded by their clients, due to apparent concerns about sharing data with third parties. *FL4*: "*...in the previous project, ... we ask the client to deploy this solution [on] AWS [as] you know, we have the inherited security in the AWS cloud. But the client was not ready to do that because they thought that there will be some kind of privacy issue. So they wanted to deploy on their own server. So we had to accept it. In that case we compromise on security*."

3) Handling security trade-offs: In cases where participants had concerns for security but they were not discussed or prioritized during development, participants did what they regarded as fulfilling basic responsibility for ensuring security.

Informing the client: Some freelance developers (*FL13*, *FL20*, *FL18*, *FL15*, *FL14*, *FL3*) expressed their concern for security by initiating discussions about security with non-technical clients who may not be aware of security concerns.

Using development frameworks: Some developers changed their development framework for security reasons. *FL2* said: "*I started my freelance development with WordPress, and when there were security concerns, I actually drifted away from it because it is open source and the community keeps updating it regularly. It was very difficult for me to keep up with it.*" Thus FL2's concern for security was coupled with his concern

about being able to use the development framework securely, leading him to drift away from WordPress. In contrast, *FL20* uses the WordPress framework as it is easy to use and does not require programming expertise, yet he is aware that Wordpress has vulnerabilities that hackers exploit. His concern for security is evident in how he makes up for using an insecure framework by adopting working practices that circumvent the framework's weaknesses, i.e., he stays up-to-date on security plugins and, if the client asks for secure software, he installs them on extra payment.

Attending to known issues: Some freelance software developers attend to security by following secure coding practices with which they are familiar. *FL8* thinks that he does not compromise on security intentionally and will not leave a dangerous vulnerability in code knowingly. *FL11* also thinks that good coding habits come with experience and then one "*cannot write bad code intentionally*" if one has experience.

B. 'Security is not a concern'

Some participants (e.g., *FL10*, *FL11*, *FL8*, *FL9*) expressed clearly that they do not worry about security when developing software projects. Their reasons are discussed below.

1) Security is a routine: Some participants (e.g., *FL8*, *FL11*) who said when asked that they are not concerned about security, seem to have a nuanced perception of security. They see some aspects of security as being routine, but other aspects that require adopting particular security protocols and standards are treated differently.

2) Reliance on development tools: Some participants are satisfied they are already doing secure development which comes from using the right tools. *FL11* said: "*I do not really care about security ... software development is more about choosing the right libraries and technologies – if a library is not used by many developers, this means that it is not reliable*".

3) Freelancing platforms: Freelance developers' perceptions of platforms often shape their prioritisation of security. *F10* said: "*I'm working on the fiver [platform], so the top priority is always about the money and the time; [clients] don't really care about the quality*".

4) Project size: Some freelance developers (e.g., *FL11*) suggested that security is a concern only in bigger projects.

In summary, freelance developers are not oblivious to security concerns in their software projects. The concern for security manifests itself in their interaction with the clients, in the development tools they choose, and in the coding habits they practice. However, security may require extra effort and time, and thus cost more. Freelance developers often struggle to sell security to their clients. Still, security measures may turn out to be insufficient.

VI. Operationalise of Security Thinking in OFSD

This theme identifies how freelance developers operationalise security thinking. It shows different perceptions of security by freelance developers and hence why it is difficult to get a common expectation of security from them. Although one participant, *FL11*, explicitly made a distinction between

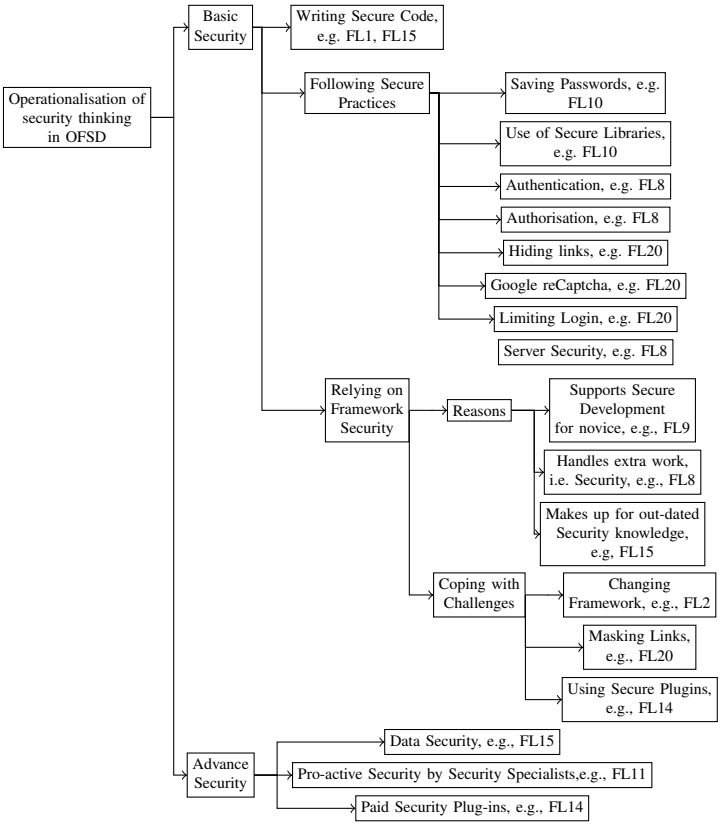

Fig. 6. Theme 2: Operationalising Security Thinking in OFSD

what he called *basic* and *advanced* security to explain how security is perceived, we identify this pattern of operationalising security in responses of other participants as well. Hence, we have adopted this distinction in explaining this theme. *Basic security* encompasses routine secure coding practices that are practiced without explicit requirements. On the other hand, *advanced security* requires specialist security expertise, where requirements need to be defined explicitly. Figure 6 shows this theme; its sub-themes are discussed below.

A. Basic security

This sub-theme discusses what participants considered as basic security, i.e., security that comes without explicit requirements. While some participants emphasize that freelance developers should ensure basic security, they often differ in what they mean by it. The perceptions of basic security that emerge from data are: *writing secure code*, *following secure practices* and *relying on the security of development framework*. We discuss these perceptions below.

1) Writing secure code: Some participants (e.g. *FL1*, *FL11*, *FL15*, *FL16*, *FL19*) expressed that secure development comes with experience and should not need separate discussion with the client. These participants did not consider security as something separate from development. *FL11* said: "*Like more years you get into the code, uh, your coding habits change... If your coding habit is good enough, like you have spent years of experience in doing the development, then you are,*

in any case, going to write the right code and just avoid those vulnerabilities." *FL11* also thought that secure code development is a habit that comes with experience, and that developers who write good and secure code over the years do not need extra effort and time to address basic security, because "*your mind doesn't let you do that*."

FL1 said that security does not need a separate discussion with the client, as it should be present in all the development modules: "*Actually, [security] is not a separate topic ... Let's say we have 10 modules for each of the products. So [security] is not going to be discussed with the client or anybody because it should be available with all the possible modules*." *FL19* worked in a small team and considered security part of their development activities. *FL16* said that security requires just writing the same code with some extra checks and conditions, which does not increase the cost of the project much.

2) Following secure practices: Some participants defined basic security as following secure practices in software development. However, when asked, they varied in what they considered basic security to be.

Some participants (e.g., *FL10*, *FL11*) thought that saving passwords securely is a basic security practice. *FL11* considered encryption of passwords as an example of basic security practice that, like other basic security parameters, should be implemented by the freelance developer without explicit security requirements[1].

Other secure practices identified by participants included the use of secure libraries (*FL10*), implementing authentication and authorization (*FL8*), masking links to frameworks that the developer considers vulnerable to attacks (*FL20*), applying Google reCaptcha (*FL20*), limiting logins (*FL20*), and considering how secure the server is (*FL8*).

3) Relying on framework security: For many participants, basic security was only about using the right tools and development frameworks; they relied on the security of development frameworks they use in order to ensure they comply with basic security. Their reasoning and strategies are discussed below.

Reasons to rely on framework security: It is a recommended practice for novice developers to rely on framework security rather than writing their own code [34]. Inexperienced developers can rely on popular, established tools and frameworks to address security concerns in their projects. *FL9* thought that using the most common technologies like Django can help less-experienced developers to implement securely. *FL11* thought that security is about choosing the right libraries and technologies, based on the standards they follow.

Many developers rely heavily on built-in security provided by development frameworks, because doing both functional development and ensuring security of applications is double the work. *FL8* said that, as Django web developers, they rely on the framework since it is hard to both write code and ensure security, which is a job of testers. Since *FL8* considered himself a professional Django web developer, he and other developers that he works with "do really not care about security". They just take care of user permissions and prioritization and also about understanding malicious requests.

Some participants find it difficult to stay up-to-date with the changing security landscape. They thus rely on frameworks to ensure they are handling basic security. In order to stay up-to-date with security, *FL15* keeps himself updated on how the new version framework should be used: "*...nowadays, every framework provides security measures. You have to take care about how to use that concept*." FL12 also relies on framework security. FL8 acknowledged that his security knowledge was not up to date but seeks refuge in knowing that Django takes care of most of the security vulnerabilities.

Coping with security challenges of frameworks: While many participants relied on framework security, we note that they made conscious decisions about choice of frameworks. Freelance developers replaced the development framework if they realized that it was vulnerable (e.g., *FL2* and WordPress); or, if they could not replace it, they found a way to cope with its security challenges. Thus, most of the participants have trust/mistrust relationships with frameworks. *FL20* preferred WordPress because it does not require programming knowledge to develop web applications. However, to deal with security issues, *FL20* said that he masked WordPress links with their custom names to hide the WordPress framework from common attackers. Similarly, *FL14* considered WordPress to be secure enough, but uses plugins if advanced security or security testing is required by the client.

B. Advanced Security

Participants thought that advanced security involves concerns that need to be discussed explicitly with the client and requires extra effort, payment, time and consent of the client.

FL15 thought advanced security is about data security. It comes with technologies that make data storage and transition secure, which should be addressed explicitly with the clients. *FL14* used paid security plugins for WordPress in order to offer advanced security to clients, if they demanded it. *FL11* thought advanced security requires the role of security specialists in a project and more proactive development.

In summary, freelance developers expressed highly personal views of what *basic security* is and how security is manifest in their development practice. Even participants who stated explicitly that security is not their concern, when asked, showed some attention to security in their projects. They made conscious decisions on how they chose and used development frameworks. Moreover, some security practices were described as part of their routine activity. In the absence of adequate guidance on what basic security is in the OFSD context, freelance developers have formed their own understanding. This can lead to an erroneous belief that security is addressed adequately in their projects, a perception that may fall short of the basic security threshold maintained by other practicing freelance developers.

[1]It is worth noting that hashing, rather than encrypting, passwords is considered a safer design [33]. The issue of developers' security knowledge and how it leads to insecure code is also reported in earlier studies [15].

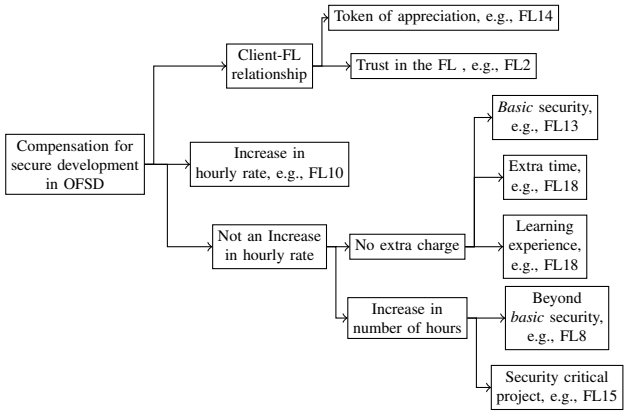

Fig. 7. Theme 3: Compensation for security in OFSD

VII. Compensation for Secure Development

This theme (shown in Figure 7) concerns the dynamics of financial investment in secure development in OFSD: the relationship between developer and client, and freelancers' perceptions of how they should be compensated for security. The sub-themes are discussed below.

1) Client and freelancer relationship: The payment for secure programming is dependent on what kind of relationship exists between the freelance developer and the client.

Trust in the freelancer: Some freelance developers prefer working with the same client and build a trusting relationship. Hence, if they inform the client of security features that they need to implement, the client pays for it. For example, *FL2* said: "*Initially we had issues on price. If I suggested 200 dollar and he would say, no, it's 50 or 60. As the time passed he understand I never demanded extra money. I'm not sure it's the quality or trust. He just accept my proposal now.*" When asked about how security is handled in his projects, he said: "*I think I had to do all of that myself, and the client was fine with that. They just trusted me to deliver the right thing*."

However, in cases where freelance developers work with different clients, they find it hard to win the client's trust. *FL3* said: "*If [we] find something, we let them know that it's insecure in this many ways. And if the client find this useful, they can understand it... But they do not often agree with us, because they knew we are [also] finding ways for earning money*."

Token of appreciation: Freelance developers expect to be acknowledged for doing a good job. *FL14* considered extra payment for security as a token of appreciation from the client: "*And if someone who focuses on that [security] and emphasize that he should be paid, yes, he should be appreciated at least*."

A. Increase in hourly rate

A few freelance developers said they would increase their hourly rate if they learned secure programming. And a client who wants secure code should hire skilful developers and pay more. For example, *FL10* works on Fiverr, as he looks for small, quick jobs that are paid less. He thought that security in software projects always comes down to money. If clients are on a freelancing platform where clients are charged less, then the client is expecting less quality, which comes with a compromise on security. *FL10* also thought that, if he learns security skills, then he will definitely increase his hourly rate: "*You [client] have to find a resource or find a freelancer who does...secure programming on its own. He doesn't charge extra for it. He is more expensive than me...*"

B. Alternatives to an increase in hourly rate

Some developers thought that clients should not be charged extra for security, as it is an inbuilt practice. For example, *FL8* said that he would change his rate only if he were doing completely different development like machine learning, but not for security. As shown in Figure 7, participants who do not increase their hourly rate, either do not charge extra for security or bill for extra hours. These are discussed below.

1) No extra charge: FL1 said: *"Actually, it's a part of the development. So whenever we [are] just architecting the project in the beginning, we try to keep everything in that.*" Freelancers may not charge extra for implementing security if they have time and want to learn something new. *FL18* said: *"If we are having enough free time ... and we are not having any other projects to handle, then yes, we do agree [to do security] that we will be gaining experience from that*." Freelancers may not charge extra if it is a security practice with which they are comfortable, i.e., they consider it *basic* security. *FL13* said: "*Actually, it totally depends. If that is something in my hands, which can be implemented on the go, [I] don't charge extra*."

2) Increase in number of hours: Participants charge for extra hours if they need to implement security beyond their usual (*basic* security) practices, such that it requires extra effort and time, or for a security-critical project that requires skills of a security specialist.

Beyond basic security: *FL8* said that if he adds security, it is like adding new features, and he will just charge for more hours, as it would require extra effort and time. *FL13* said: "*[For security] I just charge for some extra hours, which I'm going to work on*." *FL14* also charges extra for security, depending on what features a client asks for beyond the basic security that he usually incorporates in his price at the start of the project. *FL14* thought that clients should accept freelancers' demand for extra payment to implement security: "*You know, security is the prime thing in development; if someone is not focusing on that too, it is a real threat*." *FL16* suggested that secure development does not double the price of the project, but rather increases it by 10%-20%.

Security-critical project: Freelancers may charge extra for secure coding that requires the skills of security specialists for security-critical projects. *FL15* considered secure programming as good programming practice that doesn't need additional payment, as it should be normal practice. However, if it is a security-critical project, then *FL15* said that extra charges would apply.

In summary, participants had different views on compensation for secure software development in the freelance environment. Time, money, and security skills are important resources that determine how secure programming is compensated. Some FL developers who do not have security skills and work on small, quick tasks categorise security as a specialised skill, and indicate that they will increase their hourly rate when they learn how to apply secure practices in software development. Others who practice basic security skills only charge for extra hours for adding advanced security features which they do not usually implement and which would require extra time and effort. Freelance developers feel that they need to make a convincing case for security to ask for additional compensation from the client. Most often clients agree to give time, however, freelance developers report that it is necessary to build a trust relationship with the client in order to convince clients to pay for security-related tasks. Participants expressed an ethical approach toward charging for security. If implementing security is part of their routine work and does not require extra effort and time, they do not charge extra for it. However, when it requires them to move out of their primary skillset and to work extra hours, they charge for it.

VIII. Limitations

This paper presents a qualitative data analysis of interviews with 20 freelance developers. This was a convenience sample, but we ensured that participants met basic criteria as freelance developers. Although overall the participant demographics demonstrate a range of experience and backgrounds, it is inherently a limited sample. For example, the majority of the participants were men. Earlier studies (e.g., [4]) on freelance developers report similar demographics, i.e., male developers are over-represented in the online job marketplaces. Another limitation is that participants are predominantly from India and Pakistan, which happen to be where many freelancers are based [4].

Nevertheless, we consider the participants to be a reasonable sample population of freelance developers, in the sense that they represent different scenarios in the freelance environment, i.e., freelance developers working alone or in teams of different sizes and types, with different programming and learning backgrounds, on different types of projects, using different freelance platforms, and with varying experience (number of years and projects) and income levels.

We sought primarily to understand the perspectives and practices of individual freelance software developers. Although we asked questions on whether the participants had worked in teams to understand their working dynamics, we did not investigate how security is handled in the software teams, nor look at how their collaboration models work to ensure that security is handled at all layers. This is a subject for future research.

Moreover, although our sample population represents freelance developers working in different scenarios, it is possible that freelance developers working with different programming languages and frameworks have different security practices and concerns. An interesting line of research is to compare security perspectives and concerns for different OFSD demographics, for example, NodeJs developers vs. Django developers, female developers vs. male developers and developers working in teams vs. developers working alone.

IX. Discussion and future work

The work reported in this paper provides a descriptive account of the nuances of security perceptions held by freelance developers in order to better understand factors that may compromise their attention to secure software development. This section highlights broader themes that cut across our findings, situating each in relation to relevant literature, and suggesting implications for future research and activity.

A. Payment considerations in OFSD

Freelance software development encourages fixed-term and project-based association with the project owner. The project's cost model defines what a developer is required to do and, by extension, how much effort a developer is willing to invest.

Costing in secure software development is an active area of research [35], with research in recent years focusing on the economics of secure software development, i.e., analyzing "software security in the context of business, management and finances" (p.5, [35]). Although, earlier studies (e.g., [11], [12]) with freelance developers reported that payment levels for security do not have significant effect on the security of code, our study addresses in more detail how developers' own security perceptions relate to payment. Developers were found to vary in how they believed secure software development should be compensated, and in how they charged their clients.

Following this, we postulate that online freelance development offers additional socio-technical dynamics that should be considered in developing cost models for OFSD. Within socio-technical systems "tasks are distributed among human and technological components, which mutually affect each other in contingent ways" [6] (p. 1). In OFSD, different experience levels [17] and different working patterns [18] influence how freelance developers view compensation for security. Future research needs to address how social dynamics and economic instability of freelance developers influence the cost estimation models for security and propose possible solutions for handling any negative impacts on software security.

B. Client-freelancer relationship

Clients and freelance developers work together to produce secure software in a task-based relationship. Multiple engagements between a client and developer were reported to be accompanied by a growth in trust.

Establishing trust in online communities is an active area of research (e.g., [36], [37]). Toth et al. [38] conducted a survey with 127 freelancers to explore the relationship between trust and work engagement in a virtual community. *Work-engagement* has a strong link to meaningful work [39] and is defined as "a positive, fulfilling work-related state of mind that is characterized by vigor, dedication and absorption"

[40] (p. 74). *Person-job fit* is described as "a match between personal abilities and demands of the job" [38]. They suggest that trust in digital communities positively affects both work-engagement and person-job fit.

Many participants in this study reported experiences in interacting with non-technical clients, who – in the absence of an established trust relationship – interpreted efforts by the developer to improve the security of the software as an attempt to extract more money. However, we also noted that freelance developers see a change in clients' attitudes and willingness to include payment for security once a relationship develops; the developers are perceived by clients to be trustworthy, and clients are willing to accept suggestions by the developer for additional work. We suggest that, in cases in which trust has not developed, the transactional, task-based association between freelance developers and clients may lead to security compromises. To counter this risk, in engagements in which *trust* has not been established, the freelance developer may require direct compensation for security tasks. Alternatively, the freelance community may benefit from the establishment of mechanisms within OFSD platforms that can promote *trust* between clients and developers.

C. Security interventions for freelance workers

Freelance developers come from a range of educational backgrounds and have different perceptions about what security implies in software development. This can have negative consequences. Prior work has shown that acquaintance with a security concept does not necessarily translate to meaningful engagement with code. For example, Acar et al. reported that 13% of developers studied perceived their code to be secure but actually produced insecure code [41]. Although variation in developers' security knowledge is common in 'employee' developers as well, the company environment can be harnessed to support learning/development of its developer community. Research on different types of security intervention on 'employee' developers have explored how security behavior of developers can be improved over time through security awareness workshops [42]), how developers can be encouraged to adopt security tools through peer networks [43]) and whether security audits influence developers' security behavior [44].

In contrast, while freelance software developers constitute an important segment of the developer community, there is not a corresponding motivation for companies to invest in or develop secure coding skills or security awareness within this group of workers. Interventions and training that will improve security practices within freelance communities are lacking. To help the broader developer community build secure, and hence more sustainable projects for society [9], security interventions and training must be tailored to support professional development within the rapidly growing online freelance software development industry.

D. Ambiguous security obligations

Freelance developers vary in whether they consider secure development to be a part of software development or whether it needs to be specified separately in project requirements. Moreover, those who consider it a part of software development, vary in which aspects of secure development 'go without asking'. Due to the difference in how freelance developers' interpret responsibility, and their varied security knowledge and skillsets, they look for explicit client *requests for* [11] and *investment in* secure development (i.e., payment, time) beyond their normal coding practices.

The responsibility of freelance software developers for producing secure software is important, as they use their skills and knowledge to develop applications which have direct or indirect impact on society [13]. However, they cannot be held responsible if they do not have control over their actions [45] – which are often constrained by the transactional nature of their work.

The analysis presented here suggests that freelance developers are not oblivious to their responsibility to write secure code. However, security is multi-valent in nature [46]; understood and implemented at various architectural levels (e.g., code-level, design-level); and has different aspects (e.g., password storage, authentication, secure data transmission). In the absence of explicit security requirements, developers address security in different ways [23]. The subtleties of obligations around security held by freelance developers in relation to companies must be better understood before freelance developers can be held accountable for producing insecure software. Like workers in other contexts [47], these developers require support to ensure that code is secure and sustainable.

X. Conclusion

The multivalent nature of security, when combined with unique dynamics of online freelance software development, raises issues which – if not attended to – lead to security compromises in freelance software development. To summarise our findings:

- Security thinking is evident in freelance developers' descriptions of their work, and is both motivated internally and demanded externally.
- When security thinking is operationalised, there are key differences in how security issues are scoped by different developers. There is a lack of common understanding of what security implies.
- Finally, financial investment influences freelancers' attention to secure coding. There is some expectation about compensation for secure software development, but only if it is seen as requiring specialist knowledge by the freelance developers.

These findings suggest some differences in the way *freelance* developers and *employee* developers handle security issues, and thus may require different approaches (such as tailored security interventions) to support their security practices – and help the developer community overall build more sustainable software for society.

XI. Acknowledgements

This work was supported, in part, by UKRI/EPSRC grant EP/R013144/1 SFI grant 13/RC/2094 P2.

References

[1] "2022 developer survey," https://survey.stackoverflow.co/2022, accessed: 2022-09-27.

[2] J. Younger, "The big freelance skills needed as companies rebuild after covid 19," *Forbes*, Apr 2020. [Online]. Available: https://www.forbes.com/sites/jonyounger/2020/04/19/the-big-freelance-skills-needed-as-companies-rebuild-after-covid-19/

[3] V. Gupta, J. M. Fernandez-Crehuet, and T. Hanne, "Freelancers in the software development process: A systematic mapping study," *Processes*, vol. 8, no. 10, p. 1215, 2020.

[4] N. Beerepoot and B. Lambregts, "Competition in online job marketplaces: towards a global labour market for outsourcing services?" *Global Networks*, vol. 15, no. 2, pp. 236–255, 2015.

[5] B. Galobardes, M. Shaw, D. A. Lawlor, J. W. Lynch, and G. D. Smith, "Indicators of socioeconomic position (part 1)," *Journal of Epidemiology & Community Health*, vol. 60, no. 1, pp. 7–12, 2006.

[6] M. Noorman, "Computing and Moral Responsibility," in *The Stanford Encyclopedia of Philosophy*, Fall 2020 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020.

[7] S. Johal and G. Anastasi, "From professional contractor to independent professional: The evolution of freelancing in the uk," *Small Enterprise Research*, vol. 22, no. 2-3, pp. 159–172, 2015.

[8] "National security agency. securing the software supply chain: Recommended practices for suppliers," https://media.defense.gov/2022/Oct/31/2003105368/-1/-1/0/SECURING_THE_SOFTWARE_SUPPLY_CHAIN_SUPPLIERS.PDF, accessed: 2022-10-31.

[9] N. Condori-Fernandez and P. Lago, "Characterizing the contribution of quality requirements to software sustainability," *Journal of systems and software*, vol. 137, pp. 289–305, 2018.

[10] M. Tahaei and K. Vaniea, "A survey on developer-centred security," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2019, pp. 129–138.

[11] A. Naiakshina, A. Danilova, E. Gerlitz, E. von Zezschwitz, and M. Smith, "" If you want, I can store the encrypted password": A Password-Storage Field Study with Freelance Developers," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 140.

[12] J. Bau, F. Wang, E. Bursztein, P. Mutchler, and J. C. Mitchell, "Vulnerability factors in new web applications: Audit tools, developer selection & languages," *Stanford, Tech. Rep*, 2012.

[13] M. A. Ahmed and J. van den Hoven, "Agents of responsibility—freelance web developers in web applications development," *Information Systems Frontiers*, vol. 12, no. 4, pp. 415–424, 2010.

[14] A. Danilova, A. Naiakshina, A. Rasgauski, and M. Smith, "Code reviewing as methodology for online security studies with developers-a case study with freelancers on password storage," in *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, 2021, pp. 397–416.

[15] I. Rauf, M. Petre, T. Tun, T. Lopez, P. Lunn, D. Van der Linden, J. Towse, H. Sharp, M. Levine, A. Rashid, and B. Nuseibeh, "The Case for Adaptive Security Interventions," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2021.

[16] V. Clarke, V. Braun, and N. Hayfield, "Thematic analysis," *Qualitative psychology: A practical guide to research methods*, vol. 222, p. 248, 2015.

[17] A. Shevchuk and D. Strebkov, "Heterogeneous self-employment and work values: The evidence from online freelance marketplaces," in *Contemporary entrepreneurship*. Springer, 2016, pp. 141–158.

[18] R. Sison and R. Lavilles, "Software gigging: A grounded theory of online software development freelancing," in *International Conference on Information Systems 2018, ICIS 2018*, 2018.

[19] R. Q. Lavilles and R. C. Sison, "A thematic analysis of software developers' experience in online sourcing marketplaces," p. 1–12, 2017.

[20] A. Poller, L. Kocksch, K. Kinder-Kurlanda, and F. A. Epp, "First-time security audits as a turning point?: Challenges for security practices in an industry software development team," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2016, pp. 1288–1294.

[21] N. U. Haq, A. A. Raja, S. Nosheen, and M. F. Sajjad, "Determinants of client satisfaction in web development projects from freelance marketplaces," *International Journal of Managing Projects in Business*, 2018.

[22] A. Edmundson, B. Holtkamp, E. Rivera, M. Finifter, A. Mettler, and D. Wagner, "An empirical study on the effectiveness of security code review," in *International Symposium on Engineering Secure Software and Systems*. Springer, 2013, pp. 197–212.

[23] I. Rauf, T. Lopez, H. Sharp, M. Petre, , M. Levine, , J. Towse, T. Tun, , D. Van der Linden, , , A. Rashid, and B. Nuseibeh, "Influences of developers' perspectives on their engagement with security in code," in *Accepted at International Conference on Cooperative and Human Aspects of Software Engineering (CHASE '22)*.

[24] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[25] J. Xie, H. R. Lipford, and B. Chu, "Why do programmers make security errors?" in *2011 IEEE symposium on visual languages and human-centric computing (VL/HCC)*. IEEE, 2011, pp. 161–164.

[26] R. Balebako *et al.*, "The privacy and security behaviors of smartphone app developers," 2014.

[27] V. R. Waldron, "Interviewing for knowledge," *IEEE Transactions on Professional Communication*, no. 2, pp. 31–34, 1986.

[28] G. Guest, A. Bunce, and L. Johnson, "How many interviews are enough? an experiment with data saturation and variability," *Field methods*, vol. 18, no. 1, pp. 59–82, 2006.

[29] S. C. Kuek, C. Paradi-Guilford, T. Fayomi, S. Imaizumi, P. Ipeirotis, P. Pina, and M. Singh, "The global opportunity in online outsourcing," 2015.

[30] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[31] T. Rankin, "13 of the latest freelance statistics you as a developer should know," 2020. [Online]. Available: https://torquemag.io/2020/10/13-of-the-latest-freelance-statistics-you-as-a-developer-should-know/

[32] V. Braun and V. Clarke, *Successful qualitative research: A practical guide for beginners*. sage, 2013.

[33] OWASP Secure Coding Practices - Quick Reference Guide, https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide, [Accessed July-2019].

[34] Secure Code Training for Developers, http://www.securecodewarrior.com/, [Accessed January-2023].

[35] E. Venson, X. Guo, Z. Yan, and B. Boehm, "Costing secure software development: A systematic mapping study," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–11.

[36] X. Ma, J. Cheng, S. Iyer, and M. Naaman, "When do people trust their social groups?" in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[37] S. Sharma, P. Menard, and L. A. Mutchler, "Who to trust? applying trust to social commerce," *Journal of Computer Information Systems*, vol. 59, no. 1, pp. 32–42, 2019.

[38] I. Toth, S. Heinänen, and K. Blomqvist, "Freelancing on digital work platforms–roles of virtual community trust and work engagement on person–job fit," *VINE Journal of Information and Knowledge Management Systems*, 2020.

[39] W. H. Macey and B. Schneider, "The meaning of employee engagement," *Industrial and organizational Psychology*, vol. 1, no. 1, pp. 3–30, 2008.

[40] W. B. Schaufeli, M. Salanova, V. González-Romá, and A. B. Bakker, "The measurement of engagement and burnout: A two sample confirmatory factor analytic approach," *Journal of Happiness studies*, vol. 3, no. 1, pp. 71–92, 2002.

[41] Y. Acar, C. Stransky, D. Wermke, M. L. Mazurek, and S. Fahl, "Security developer studies with github users: Exploring a convenience sample," in *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*, 2017, pp. 81–95.

[42] C. Weir, I. Becker, J. Noble, L. Blair, M. A. Sasse, and A. Rashid, "Interventions for software security: Creating a lightweight program of assurance techniques for developers," *Software: Practice and Experience*, vol. 50, no. 3, pp. 275–298, 2020.

[43] S. Xiao, J. Witschey, and E. Murphy-Hill, "Social influences on secure development tool adoption: why security tools spread," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014, pp. 1095–1106.

[44] S. Türpe, L. Kocksch, A. Poller, S. Türpe, L. Kocksch, and A. Poller, "Penetration tests a turning point in security practices? organizational challenges and implications in a software development team." in *WSI-WSOUPS*, 2016.
[45] J. R. Lucas, "Responsibility," 1995.
[46] I. Rauf, T. Lopez, H. Sharp, and M. Petre, "Challenges of recruiting developers in multidisciplinary studies," *In: 1st International Workshop on Recruiting Participants for Empirical Software Engineering (RoPES'22)*, 2022.
[47] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.