The Journal of Systems & Software 195 (2023) 111548

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

The PREVENT-Model: Human and Organizational Factors Fostering Engineering of Safe and Secure Robotic Systems[☆]

Christina Glasauer

University of Klagenfurt, Universitaetsstraße 65–67, 9020, Klagenfurt, Austria

ARTICLE INFO

Article history:
Received 9 March 2022
Received in revised form 2 June 2022
Accepted 12 October 2022
Available online 27 October 2022

Keywords:
Robotic systems
Human factors
Safety
Security
Software engineering
Human resource management

ABSTRACT

The rise of robotic systems demands increasingly elaborated safety and security measures in order to prevent damage to property as well as human and economic harm. While technical expertise of engineers is a prerequisite and international standards constitute an overall frame for safety and security by design, human and organizational factors also shape the development of responsible technology. The current study derives an overarching model covering both human and organizational factors facilitating the development of safe and secure robotic systems. In a qualitative interview study, we conducted three focus groups with experts from robotics, software engineering, and related domains. The interviews were analyzed by three independent coders using a qualitative content analytical approach. The resulting integrative PREVENT-Model comprises 17 individual factors and 13 organizational factors that enhance safety and security by design. We embed each factor into existing research and derive implications for practitioners and possible courses of action. The PREVENT-Model serves as a roadmap to develop tailored measures. Based on this framework, practitioners can improve personnel development programs, organizational structures, working environments, and other aspects crucial for the development of safe and secure robotic systems.

1. Introduction

Robotic systems and automated processes are becoming increasingly prevalent in industry, health care and the medical domain, as well as in people's private lives. Simultaneously, the distance between humans and robots decreases (e.g., "collaborative robotics"; Tantawi et al., 2019). Regardless of whether it is due to close proximity, constant presence, or sensitivity of tasks, safety and security (S&S) are of utmost importance in most of the domains robots are used in today (Kirschgens et al., 2019). However, vulnerabilities still occur (Taurer et al., 2019; Mayoral-Vilches et al., 2020a; Denning et al., 2009; Mayoral-Vilches et al., 2019) even though common safety issues, security risks, and corresponding mitigation measures are well known and best practices, adequate processes, and security tools exist.

As vulnerabilities can lead to serious financial losses and severe human harm (Yaacoub et al., 2021), S&S already have to be considered during the developmental process of a robot. Hence, in order to minimize negative consequences, S&S need to be implemented *by design* (Howard and Lipner, 2006) instead of *by disaster*. Besides general technical standards on safety and security (e.g., IEC 61508; International Electrotechnical Commission, 2010; and IEC 62443; International Electrotechnical Commission, 2003), several standards are specifically tailored to the development of robotic systems (e.g., IEEE's standards for robotics and automation; Institute of Electrical and Electronics Engineers [IEEE], 2015, 2021; or standards developed by the ISO Technical Committee 299 for robotics; International Organization for Standardization, 2021). These include elaborated requirements and recommendations on structuring and implementation of workflows, procedures, and documentation. Standardized software development processes, such as Microsoft's Security Development Lifecycle (SDLC; Howard and Lipner, 2006), aim at improving *security by design*. Besides DevOps (Virmani, 2015; Ebert et al., 2016), DevSecOps found its way into robotics development (Myrbakken and Colomo-Palacios, 2017; Mayoral-Vilches et al., 2020b), depicting security-wise advancement in the robotics domain. Still, there is a variety of possible sources of S&S issues. Research shows team composition, engineers' personalities, and organizational culture impact team performance and the quality of developed products (Hartnell et al., 2011; De O. Melo et al., 2013; Purna Sudhakar et al., 2011; Yilmaz et al., 2017). On these grounds, a *comprehensive* approach to improve the developmental process of robots is required. However, although research on organizational behavior, human resource management, and the role of human factors in software engineering is bountiful, it is hard to determine which of the factors that are beneficial for

[☆] Editor: Raffaela Mirandola.
E-mail address: christina.glasauer@aau.at.

https://doi.org/10.1016/j.jss.2022.111548

other domains and outcomes also positively impact S&S in robotic systems.

The present study identifies S&S-wise influential human factors in robotic systems engineering that go beyond developers' technical skills. We address factors on both the individual engineers' as well as on an organizational level that influence a robot's S&S levels during development. A qualitative interview study yielded 17 factors on the individual engineers' level and 13 organizational factors. Taken together, these factors constitute the PREVENT-Model for S&S by design. Besides scientific evidence and theoretical foundation, we present possible courses of action to improve each aspect.

With this information, managers can provide organizational structures, processes, and working environments that facilitate the development of responsible robotic systems. Knowledge about S&S-wise beneficial individual soft skills enables to optimize team composition, targeted personnel development, and strategic task allocation. The PREVENT-Model provides a holistic viewpoint on possible weaknesses and valuable strengths in the entire process of robotic systems engineering; thus, it helps to improve to-be-developed robots' S&S and to prevent severe human harm, damage to property, and economic losses.

2. Background & related work

2.1. Safety & security in robotic systems

As humans and robotic systems increasingly tend to work collaboratively (Tantawi et al., 2019), physical safety has to be warranted. Appropriate consideration of safety as a non-functional requirement (Leveson, 2016; Gall, 2008; Jamont et al., 2014; Valori et al., 2021; Lutz, 2000) is essential to prevent danger for humans and assets. Still, even if safety is addressed perfectly, all of these efforts might be levered out if a system's security measures are insufficient (Kirschgens et al., 2019; Mayoral-Vilches, 2020). Given a robotic system with elaborated safety measures, an attacker who succeeds in entering the system could easily disable existing safety controls if security barriers are low. For instance, researchers were able to deactivate a robotic mobile platform's emergency stop remotely via Wi-Fi (Taurer et al., 2019) and to compromise a robotic arm with the robotic ransomware "Akerbeltz" (Mayoral-Vilches et al., 2020a). Although these examples luckily were performed for research purposes and with no malicious intent, these case studies alarmingly illustrate the importance of security as a prerequisite for physical safety.

2.2. Roots of vulnerabilities

In the case of Taurer et al. (2019)'s research project, deactivation of the mobile platform's emergency stop was enabled because (1) the system's programmable logic controller (PLC) was connected to the internal network of the robot and (2) the manufacturer used standard passwords (included in the robot's manual) – therefore, access to the network was easily possible. In the example of Akerbeltz (Mayoral-Vilches et al., 2020a), researchers took control of the system by exploiting *known but unpatched* vulnerabilities. In both cases, several instances in the developmental process seem to have failed. While insecure programming occurred in the first place, lack of procedures such as penetration testing and threat modeling (as Taurer et al., 2019 discuss) and lack of ambitions on behalf of the producers (as discussed by Mayoral-Vilches et al., 2020a) eventually allowed for the intrusions into the systems. Evidence of software engineering research suggests that engineers bear the most blame for vulnerabilities. The occurrence of bugs is attributed to the engineers' lack of knowledge, attention, or motivation (Assal and Chiasson, 2018a). For example, Oliveira et al. (2014) found that specific security vulnerabilities might be "blind spots" that engineers easily overlook if they do not explicitly pay attention to them. Human errors in software development and requirements analysis have also been classified (Anu et al., 2018, 2020) and there is evidence that better understanding of possible human errors results in fewer mistakes when writing requirements documents (Hu et al., 2018).

It would be the easy way out to shift the responsibility for security issues upon engineers. However, if the management neglects security due to competing priorities (Assal and Chiasson, 2018b) and engineers are not given sufficient time and resources to care about security, these persons may not be perceived as the root of vulnerabilities. Across a variety of factors mostly concerning individual engineers' motivations, Song et al. (2018) identified *perceived organizational support* to have the greatest effect on engineers' intentions to assimilate secure software development innovations. Interview studies identified bad management processes, competing priorities, the way an organization is treating non-functional requirements, and lack of structured procedures as reasons for security issues (Assal and Chiasson, 2018b, 2019; Tahaei and Vaniea, 2019).

2.3. Research objectives

Although software security research provides valuable indications, scientific evidence on aspects affecting S&S in *robotic systems* is not established yet. Therefore, an exploratory study is needed to ascertain human and organizational factors that influence S&S when developing robotic systems. Our study addresses factors on both the individual engineers' as well as on an organizational level. Regarding the individual level, we examine which soft skills like personality traits, behavioral/cognitive patterns, or attitudes of individual engineers impact S&S of the system under development. In the present work, we will refer to these as *individual factors*. Further, we investigate *organizational factors*, which we define as influences on S&S on a managerial level, such as aspects of organizational culture, leadership behaviors, or managerial actions. The study addresses the following research questions (RQ):

RQ1 *Which of engineers' individual soft skills facilitate or impede consideration of safety and security during robotic system development?*

RQ2 *Which organizational factors facilitate or impede consideration of safety and security during robotic system development?*

Identification of these influential non-technical factors enables to address them and therefore improves S&S by targeting essential nodes (e.g., persons, teams, instances) and edges (e.g., processes, channels of communication) in the developmental network.

3. Method

3.1. Study design & sample

An exploratory qualitative interview study served to answer the two RQs. We identified professionals in industry and research who have devoted themselves to at least two of the following domains for several years: robotic systems engineering, cybersecurity, safety, software engineering, or ethics. We reached out to these professionals in stages via e-mail (Overall, we contacted 21 professionals, of whom 13 agreed to participate in the study. One person in the last interview had to cancel last-minute, resulting in $n = 12$ participants in total).

We conducted three focus group (FG) interviews (Gawlik, 2018) consisting of $n_1 = 5$, $n_2 = 4$, $n_3 = 3$ experts respectively

Table 1
Fields of expertise of focus group participants (P).

Group	P	Robotics	Security	Safety	Ethics	Softw. Eng.
1	1	×			×	
	2	×			×	
	3	×	×			
	4	×				×
	5	×	×			
2	1	×	×	×		×
	2	×		×		
	3		×	×		×
	4	×		×		×
3	1	×	×	×		×
	2	×	×	×		×
	3		×			×

(see Table 1 for their fields of experience). While in the first FG all participants were in academia, we strove to compose groups of practitioners from industry and non-academic institutions in the subsequent FGs. Since no new topics emerged in the third FG, the category system was saturated (Saunders et al., 2018) and we stopped data collection after the third FG. Experts participated voluntarily and did not receive any compensation for taking part in the interview. Except for one person from an extra-European region (participating in the first FG), all respondents came from Austria and Germany. The ethics committee of the University of Klagenfurt has approved the conduct of this study on June 15, 2021.

3.2. Focus groups

Due to the SARS-CoV-2 pandemic, the three FGs took place on an online meeting platform. While the first group interview was conducted in English, the other two were conducted in German. Each FG lasted for 90 to 110 min and they were recorded with written consent of participants. A predefined interview guide set the key questions to be discussed (see Table 2). We adapted the first question (Table 2, 1a/1b) for the second and third FG, as it showed to provoke elaborations on mostly technical solutions for S&S in the first group, which we did not intend. Instead, we wanted to encourage participants to talk about their personal experiences. The complete interview guide, including subquestions, is available in the supplementary material.

Prior to the FGs, a moderator welcomed the interviewees and gave a short impulse presentation on S&S in robotic systems. The moderator highlighted the complexity of robotic systems development to attune the group to the contents of the discussion. For the first key question (see Table 2, questions 1a/1b), respondents were given two to four minutes to reflect before they presented their answers to the group. Participants then responded and discussed each others' thoughts. The moderator led through all the key questions to ensure that all topics were covered and all respondents had equal opportunity to speak. In case of ambiguous answers, the moderator posed deepening follow-up questions to facilitate additional explanations. During the discussion, the key questions addressed at the current point of conversation were permanently visible to the participants on a screen.

3.3. Qualitative analysis

Recordings of the FGs were transcribed verbatim. We conducted Qualitative Content Analysis (Mayring, 2000, 2014) using the open-access software QCAmap (Mayring and Fenzl, 2013) to identify the relevant factors the experts addressed in the discussions. Due to the exploratory character of the study, we chose an inductive category formation approach, that is, we formed the categories ("factors") based on analysis of the interview transcripts.

Three independent coders separately analyzed the data for the two RQs (see Section 2.3). Finally, the three coders met in an extensive intercoder meeting to collate their resulting category systems and to achieve communicative validation (Kvale, 2007, p. 125). We merged corresponding categories (e. g., categories that were named differently but expressed the same meaning), in order to obtain an overall category system.

Several measures served to assess intercoder agreement and consistency. On a category-system level, we assessed percentage agreement between coders on existence/non-existence of each category across the entire transcript. To measure agreement on the level of coded passages, four different coefficient estimates were examined: Gwet's AC_1 (Gwet, 2008), Krippendorff's α (Krippendorff, 2004), Conger's κ (Conger, 1980), and simple (percent) agreement (Lombard et al., 2002). These measures are suitable for multiple coders and multiple categories. Measures of intercoder agreement were calculated in R (R Core Team, 2021) using the package `irrCAC` (Gwet, 2019).

4. Results: The Prevent-Model

The resulting 290 recorded minutes of interview yielded 2209 lines of transcript. Analyses of three independent coders revealed 17 individual factors (RQ1) and 13 organizational factors (RQ2) considered relevant by experts in our FGs. Table 4 depicts the overall category system with factors grouped into five individual and three organizational clusters. For the majority of factors, the three coders agreed 100% on existence/non-existence (column $r(C)$ of Table 4). Examination of intercoder agreement on codings of passages yielded coefficients ranging from 0.698 to 0.733 for the individual factors and from 0.785 to 0.811 for the organizational factors (Table 3), indicating substantial agreement of coders (Landis and Koch, 1977). The entirety of these factors constitutes our model of individual and organizational factors that facilitate the engineering of safe and secure systems, which we refer to as the Prevent-Model (*Precautionary Engineering: Variables Enabling Nonhazardous Technology*).

In the remaining part of this section, we describe the identified Prevent-factors. We cite our experts' most representative, exemplary quotes for each factor. Note that the statements from interviews 2 and 3 were initially phrased in German, and we, therefore, translated them to English. Each quote is referenced with a group index (G), referring to the group the statement came up in, and an index (P) labeling the participant who expressed it (e. g., [G3-P2] referring to participant 2 in the third FG). We corroborate the identified factors with evidence from psychology, organizational behavior, and software engineering research. Lastly, we summarize implications for practitioners ("*Take-aways*") derivable from our results.

4.1. Individual factors

Regarding RQ1, the factors concern individual engineers' soft skills (personality traits, attitudes, potentials, ...) positively influencing S&S of the robotic systems they develop. Engineers can increase their consciousness of these aspects and pay special attention to behave in these ways according to their capabilities. Leaders and managers in turn might facilitate these behaviors and attitudes by making use of the possible courses of action provided for each factor.

Table 2
Key questions of the interview guide of the FG interviews.

1.	(a)	Think about actions an engineer can take and organizational actions, that contribute to S&S in robotic systems. Which actions do you consider most effective?
	(b)	Think about a safety-/security-related occurrence you experienced in the past. Which qualities helped individual persons to contribute by detecting and fixing the issue?
2.	What are individual engineers' soft skills that contribute to S&S being considered during the development process: Why are they important?	
3.	What are structural/organizational circumstances that foster development of safe and secure robotic systems?	

Note: Question (1a) only was used in the first FG and replaced by question (1b) in the following FGs. Questions (2) and (3) were the same in all of the three interviews.

Table 3
Coefficients of intercoder agreement for the coding of individual factors (RQ1) and organizational factors (RQ2).

Coefficient	Estimate	$S.E.$	p-value
Individual factors			
Gwet's AC_1	0.717	0.065	<0.001
Krippendorff's α	0.698	0.043	<0.001
Conger's κ	0.706	0.065	<0.001
Simple agreement (%)	0.733	0.064	<0.001
Organizational factors			
Gwet's AC_1	0.796	0.061	<0.001
Krippendorff's α	0.785	0.038	<0.001
Conger's κ	0.791	0.061	<0.001
Simple agreement (%)	0.811	0.061	<0.001

Note: Estimates, their standard errors ($S.E.$), and p-values of testing the hypothesis that the respective coefficient equals zero for each measure of intercoder agreement.

4.1.1. *Safety & security assertiveness*

Highly assertive individuals take a stand for S&S by communicating the need for these qualities. They interfere if S&S are neglected, address the topic in team meetings, or engage in other forms of promoting S&S. Asserting S&S is a behavior that prevents these aspects from being neglected. All three coders strongly agreed on the existence of this factor. The need for S&S advocates was expressed several times (coded 8.67 times on average) and in each of the three FGs.

> *I think it is really important to have self-confidence to really advocate the topic. Because, if there is a super-specialist on security but he is unable to speak up and he does not claim it from the project management [. . .], then it's not going to happen. But if there is somebody who also has the confidence to advocate the topic, to really claim it, and repeatedly point it out, then [security] is something, that actually will happen.* [G2-P3]

Safety & Security Assertiveness was the most prominent factor of the *Safety & Security Appreciation* cluster. Assertiveness also is known as the key characteristic of "security champions" (Jaatun and Soares Cruzes, 2021; Boström et al., 2006; Migues et al., 2020). Evidence on the positive relationship between assertiveness and job and team performance (Pearsall and Ellis, 2006; Judge et al., 2013) underpins the validity of this factor. *Take-away: Encourage employees to speak up if they have concerns and listen to their demurs. Do not oppress their objections and misgivings. As an engineer, voice your concerns, if you apprehend possible vulnerabilities and safety issues.*

4.1.2. *Awareness towards safety, security, and risks*

This specific type of awareness comprises sense for, appreciation of, and knowledge about the role of S&S, especially about related risks and impacts. Aware engineers know about the consequences of S&S-issues and bear in mind that vulnerable human beings will work close to the robot they develop. They do not overlook or ignore the dangers related to their work. Participant G2-P4 concluded: "[A robot] just looks inconspicuous, but it really is a machine and not a toy". Although frequencies of coded passages vary among coders for this factor, all of them identified this category at least once.

> *It is important that there is awareness in the first place and that S&S is an important topic and that, if you do not consider it, it can lead to problems — that this exists within developers. Because if I am not aware of that, even if I am the greatest perfectionist, then I will not take notice of this topic and I will not spend a lot of time on it.* [G2-P2]

The importance of awareness also is appreciated within Microsoft's SDLC (Howard and Lipner, 2006), in which "Education and Awareness" constitute Step 0. While the SDLC proposes information and (videotaped) lectures to raise awareness among engineers (Howard and Lipner, 2006; Gardner, 2014) lists a variety of other possible training types to increase awareness for information security, including posters, web-based courses, or informal "Lunch and Learn Sessions". *Take-away: Raise awareness through information (e.g., lectures, training, informal sessions, brochures). Engineers may reflect on risks and possible negative consequences of unsafe and unsecured robots ("what could happen?"). Awareness towards S&S might already be raised in the course of education.*

4.1.3. *Questioning actions & information*

Critically thinking people do not take guidelines and tasks for granted unquestionedly. They reflect and (re-)appraise assigned tasks, possible issues, and consequences. They scrutinize procedures and tasks and do not implement things unreflectedly — thereby they can prevent S&S flaws from occurring. The importance of questioning instructions emerged 5.33 times on average among coders. Participant G2-P4 emphasized this trait's importance by explaining that "as a project leader you are not perfect either and sometimes you will forget about things. And these things simply have to pop up and have to be questioned by the project team".

> *These are persons who have a closer look, or people who rather ask again instead of carrying on regardless.* [G2-P3]

Bermudez (2015) discusses approaches towards the similar concept of *reflective skepticism* and conflates existing definitions into the key intellectual operations of "the methodic scrutiny of arguments and thinking procedures, the examination of underlying assumptions, and the disclosure and correction of distortions" (Bermudez, 2015, p. 108). It is argued that critical thinking can be trained (Halpern, 1998) and experimental studies show better decision process outcomes for individuals encouraged to think critically than for control groups (Helsdingen et al., 2010). *Take-away: Engineers may strive to consider underlying structures of problems, reassess whether information is sufficient or conflicts, or reflect on alternative ways to phrase a problem. Managers may promote critical thinking skills among employees, encourage critical reflection through positive remarks and feedback, and give task instructions that enhance critical thinking.*

Table 4
Factors of the PREVENT-Model and frequencies of coded passages per factor and coder.

Level	Cluster	Factor	C1	C2	C3	$\bar{x}$ (SD)	$r(C)$
Individual factors	Safety & Security Appreciation	Safety & Security Assertiveness	5	13	8	8.67 (4.04)	1.00
		Awareness towards Safety, Security, & Risks	1	16	6	7.67 (7.64)	1.00
		Questioning Actions & Information	3	12	1	5.33 (5.86)	1.00
		Intrinsic Motivation (Passion)	3	5	1	3.00 (2.00)	1.00
		Sense of Responsibility	3	4	1	2.67 (1.53)	1.00
	Perceptual Range	Bigger Picture	12	30	12	18.00 (10.39)	1.00
		Receptiveness	12	19	14	15.00 (3.61)	1.00
		Creativity	8	0	6	4.67 (4.16)	0.67
		Learning Receptivity	0	3	2	1.67 (1.53)	0.67
	Structuredness	Conscientiousness	17	17	13	15.67 (2.31)	1.00
		Analytical Thinking	4	8	7	6.33 (2.08)	1.00
		Feasibility Orientation	1	1	1	1.00 (0.00)	1.00
		(Self-)Management	1	0	0	0.33 (0.58)	0.33
	Dealing with Mistakes	Acceptance of Mistakes	6	8	7	7.00 (1.00)	1.00
		Awareness towards Fallibility	2	3	2	2.33 (0.58)	1.00
	Teamwork	Ability to Work in a Team	9	9	7	8.33 (1.15)	1.00
		Compliance	3	5	1	3.00 (2.00)	1.00
		Σ	90	153	89	110.67 (50.46)	–
		$\bar{x}$	5.3	9.0	5.2	6.51 (2.97)	0.92
		SD	4.8	8.1	4.6	5.31 (2.77)	0.19
Organizational factors	Corp. Safety & Security Culture	Implementation of Structured Processes	15	34	15	21.33 (10.97)	1.00
		Resources	15	18	11	14.67 (3.51)	1.00
		Safety & Security as Corporate Values	7	22	9	12.67 (8.14)	1.00
		Explicit Roles	7	8	7	7.33 (0.58)	1.00
		Customer Consideration	0	13	0	4.33 (7.51)	0.33
	Team Assembly & Skills	Team Diversity	7	11	10	9.33 (2.08)	1.00
		Employee Training & Development	5	11	9	8.33 (3.06)	1.00
		Task Allocation: Matching Tasks & Strengths	10	10	2	7.33 (4.62)	1.00
		Degrees of Freedom	3	2	2	2.33 (0.58)	1.00
	Collaboration	Feedback & Error Culture	9	6	11	8.67 (2.52)	1.00
		Intraorganizational Exchange	9	9	2	6.67 (4.04)	1.00
		Fostering & Facilitating Collaboration	3	7	5	5.00 (2.00)	1.00
		Participation	3	6	4	4.33 (1.53)	1.00
		Σ	93	157	87	112.32 (51.14)	–
		$\bar{x}$	7.2	12.1	6.7	8.64 (3.93)	0.95
		SD	4.5	8.4	4.6	5.10 (3.15)	0.19

Note: Number of coded passages per factor and coder (C1, C2, C3) including means ($\bar{x}$) and standard deviations (SD) of the frequencies across coders and the proportion of coders who identified each factor $r(C)$. Total means, standard deviations and sums (Σ) across factors are depicted in rows for both the individual as well as the organizational level.

4.1.4. Intrinsic motivation (passion)

Intrinsically motivated persons pursue their task of implementing S&S driven by their own internal interests, values, and goals. In their daily work, they are motivated to do well, because they find it meaningful, exciting, and valuable. The benefits of intrinsic motivation of engineers emerged in every single FG.

> *So when you're really invested into the action [. . .] because you personally are invested in it, the outcome in regards to S&S can be more prominent because you really want to do it. You don't just have to do it, but you want to do it.* [G1-P1]

The positive effect of intrinsic motivation on task performance is well established in research (Ryan and Deci, 2000; Herlambang et al., 2021). Intrinsic motivation was found to influence the relationship between job autonomy and job performance (Joo et al., 2010) and work quality: When intrinsic motivation is high, perceived job autonomy has a greater positive effect on work quality than when intrinsic motivation is low (Dysvik and Kuvaas, 2011). In Li et al. (2015), being passionate was explicitly stated as a key characteristic of "great" software engineers. According to the Job Characteristics Model (JCM; Hackman and Oldham, 1976), targeted job design can increase employees' intrinsic motivation. In JCM, skill variety, task identity, task significance, feedback and autonomy lead to perceived meaningfulness of work, perceived responsibility for tasks, and to knowledge about actual results of the work. These in turn lead to higher intrinsic motivation, work performance/quality, job satisfaction, and lower absenteeism (Hackman and Oldham, 1976). Job rotation, job enrichment, or relational job design are strategies organizations can take to improve their employees' intrinsic motivation (Grant, 2007; Herzberg, 1968). *Take-away: Design job positions with sufficient autonomy, social relations, requirements of solving "complete" tasks with identifiable pieces of work, promote task variety, and emphasize the significance of the job for others. Engineers may actively envision the impact of their work for other people, even if it is just a small part of a bigger picture.*

4.1.5. Sense of responsibility

Responsibly minded persons feel responsible for the outcome of their work and feel in charge of the developed product's impact on users. They perceive their work as something serious as they know about potential dangers. Therefore, responsible individuals try their best to mitigate flaws in order to develop a safe and secure robotic system. A sense of responsibility differs from awareness towards safety, security, and risks in a way, that engineers are not only sensible of possible (negative) outcomes, but even feel responsible to prevent these. This factor was present in every FG.

> *I would like to have a person that does not say "this is not my sphere" or "this is not my responsibility". That's the wrong person.* [G1-P2]

The theoretical foundation of this factor's validity is given within the "Professional Social Responsibility Development Model"

(PSRDM; Canney and Bielefeldt, 2015). In the realm of personal social awareness, PSRDM's dimension of *Connectedness* is defined as "A feeling of moral obligation, responsibility, or social requirement to help others" (Canney and Bielefeldt, 2016, p. 455). According to software development research, perceived individual and collective responsibility is a main motivator to address software security (Assal and Chiasson, 2019). Unfortunately, in some cases the feeling of social responsibility seems to decrease during education (Bielefeldt and Canney, 2016, 2014). Therefore, as argued by many, consciousness on social responsibility might already be raised in education by including more non-technical, responsibility-raising classes in engineering curricula (Bielefeldt and Canney, 2014; Rulifson and Bielefeldt, 2015; Trbusic, 2014). Psychological research on responsibility aversion suggests that people tend to prefer options with outcomes highly influenced by chance over other options in which the outcome is directly influenced by them in order to avoid responsibility (Leonhardt et al., 2011). Given this human tendency, it seems even more vital that engineers genuinely feel responsible for the outcomes of their work as, otherwise, S&S outcomes might be left to chance. Organizations might increase employees' sense of responsibility by providing them with autonomy (Hackman and Oldham, 1976) and relational job design. Giving employees the chance to build and keep up relationships with customers and to collaborate in teams enables them to make a positive difference in other people's lives and tends to increase motivation (Grant, 2007). *Take-away: Implement relational job design, high degrees of freedom in how employees fulfill their tasks (autonomy), and already start to inform about social impacts and possible consequences of engineering during education.*

4.1.6. Bigger picture

Engineers who have a view of the "bigger picture" know the entire process of development and have an understanding of how the end users of the robot are affected by their decisions. Within their daily work, they consider the future work environment of the robot, customers, and their (sometimes risky) behavior. They do not perceive their task (e.g. programming) as an isolated entity, disconnected from the other development steps. Rather, they understand how all steps interrelate and understand their work's impact on other areas within the process. Experience in various backgrounds might contribute to understanding of the entire developmental process. Participant G2-P3 suggested that engineers should spend days out in the field once or twice per year in order to prevent them from getting too focused on the mere task they receive. He emphasized that "everybody has to make decisions at some point and these [decisions] affect another sphere". Coders identified this call for the need to have a view of the bigger picture 18 times on average, which makes this factor the most prominent individual factor mentioned.

> *Wherever I'm sitting as an engineer, I should have high knowledge of [. . .] the whole [developmental] cycle. And then I thought it might be useful to have people who have quite process-orientated thinking and acting to understand the overall context in the process of robotic systems engineering.* [G1-P2]

The importance of engineers to have a view of the *Bigger Picture* is illustrated best by examining its absence – a state which can be referred to as "silo mentality". Presence of this silo phenomenon is discussed as one of the reasons for the Boeing 737 MAX aircraft crashes (Englehardt et al., 2021). It is related to negative work climate and even hostility within an organization (Cilliers and Greyvenstein, 2012). The "silo mentality" was identified as inhibiting implementation of DevOps-Processes (Hüttermann, 2012; Kamuto and Langerman, 2017) and is argued to introduce a variety of further negative consequences (Mohapeloa, 2017). *Take-away: Introduce job rotation programs, exchange meetings, and "days in the field" in other departments in order to familiarize engineers with the whole development process. As an engineer, talk to colleagues and users and think about how your decisions may affect others' tasks.*

4.1.7. Receptiveness

Receptiveness is characterized by openness, curiosity, adaptability, and interest. These persons are receptive to new methods, tools, topics, and processes that they might not have used in the past, but seem more suitable for specific purposes they encounter. It includes not restricting one's bounds to the known and only sticking to conventional methods but being willing/able to stretch one's horizons. The bounds of reception are not narrow and fixed but agile and permeable. Statements on the importance of receptiveness included stories of people avoiding a problem, because it was not solvable with their most favorite tools (G1-P3) and a personal story of participant G3-P2, who explained his initial incitement to devote himself to robotic security was "curiosity combined with general interest". Receptiveness came up in every FG and was coded 15 times on average per coder with only minor standard deviation ($SD = 3.61$).

> *In my opinion it's also highly important to be at least content to acquire new [knowledge] and to implement new things. Because if you hire a new employee who says by default "We have always done it like that and I will keep doing it like that in the future", I think you will encounter problems fast. So I think openness to the new is really important.* [G3-P1]

Different aspects of *Receptiveness*, such as seeking for improvement, being open-minded, curious, and adaptable, were identified as key success factors for engineers in software development (Li et al., 2015; Groeneveld et al., 2020). This factor may be perceived as partly related to the trait *openness to new experiences* of the Five-Factor-Model (FFM) of personality (Fiske, 1949; Goldberg, 1993; McCrae and John, 1992), with overlaps in the dimensions of wide interest, imaginativeness, originality, and curiosity. Research on the relationship between personality and job outcomes is extensive but ambiguous (Cruz et al., 2015). Studies found openness to correlate with academic performance of programming students (Salleh et al., 2014). Openness also was identified as an important characteristic for most of the roles a software engineer can take (Rehman et al., 2012), and the decline in job performance over time was found to be decreased by high openness (Minbashian et al., 2013). Parker (2000) found change receptiveness to predict goal ownership and role breadth self efficacy. Evidence regarding receptiveness' curiosity facet suggests high operational validity for task performance (Mussel, 2013). Individuals' receptiveness might be enhanced by transformational leaders, who intellectually stimulate and personally challenge their followers (Bass, 1985; Tang, 2019). *Take-away: Engineers should try to open up to new ideas, tools, and methods and may not refuse these just because they are not within their routine yet. Intellectual stimulation and challenge by leaders might foster employees' potential to go beyond their routines.*

4.1.8. Creativity

Creative engineers have the ability to come up with new solutions that have not been used routinely in the past. They can think beyond the borders of conventional methods and can come up with more useful, innovative ideas. Creativity was acknowledged as a factor on its own only by two coders, while one coder included this factor in the category of receptiveness. Also, creativity only was mentioned in the first and second interview but did not come up in the third FG. For these two reasons, reliability of this factor might be considered lower.

> *[. . .] you need the visionary, the creative, who can think laterally and introduce new ideas.* [G2-P2]

Creativity is a skill already being discussed within software engineering literature (Mohanani et al., 2017). Qualitative interview studies identified creativity as a valuable factor in software engineering (Groeneveld et al., 2020, 2021). Especially in the testing phase (e. g., in penetration testing), the ability to come up with novel and uncommon test cases could enlarge the variety of tested cases and therefore prevent systems from being tested in "fair weather mode", as G2-P4 termed it. While a variety of creativity-enhancing tools exist, only some of them are commonly used in development Mohanani et al., 2017; Groeneveld et al., 2021; Vieira et al., 2012. Taking advantage of creativity-enhancing tools (decision upon which tool is appropriate might be supported by the "Creativity Patterns Guide" Vieira et al., 2012) might improve successful problem solving in the developmental process (Groeneveld et al., 2020). *Take-away: Introduce idea boxes/boards, competitions, or reward systems for good ideas. Especially for engineers with high creative potential, reserve extra think-out-of-the-box time to be creative. Make sure your organization's error culture is benign and the climate is shaped by psychological safety to enable that novel ideas can be expressed without fear.*

4.1.9. *Learning receptivity*

Engineers keep up with current topics and new S&S developments. They inform themselves about new vulnerabilities, exploits, deprecation of certificates and protocols, new possibilities and methods, etc. They continuously stay on track, improve their knowledge and needed skills, and they are willing to participate in training. Although *Learning Receptivity* was mentioned in all of the three FGs, only two coders identified it as a factor.

> *Of course you have to keep yourself current, because the topic does not rest. Yes, continuously new protocols and systems are published. Then a vulnerability is being patched and a new one appears — you have to keep in touch, you also have to be content to continuously engage with the topic.* [G3-P2]

Every year thousands of new security vulnerabilities are detected and published. Given the 18 375 records of vulnerabilities listed in the Common Vulnerabilities and Exposures database (CVE Mitre Corporation, 1999) in 2020 (Mitre Corporation, 2021), the demand for keeping oneself up to date as a security engineer is self-evident. In 2019, Mayoral-Vilches et al. (2019) introduced the Robotic Vulnerability Database (RVD), which contains 238 vulnerabilities and 1778 recorded bugs at the end of 2021 and thereby illustrates the need to stay current as an engineer in robotics. Exemplified by the robotic ransomware Akerbeltz (Mayoral-Vilches et al., 2020a), intrusion took place by exploiting publicly known but unpatched vulnerabilities, which could have been prevented if engineers had been up-to-date. To "devote oneself to continuous learning" and to "be a fast learner to keep up with the pace of new technologies" were ranked among the most important personal skills for exceptional software engineers (Groeneveld et al., 2020, p. 1099). *Take-away: As an engineer, stay up to date and inform yourself about new developments. Managers and leaders should grant time during working hours to do so and might introduce central information platforms containing necessary news to keep employees informed.*

4.1.10. *Conscientiousness*

When working on a task, the engineer pays attention to details, is concentrated, and works meticulously and conscientiously. Conscientiousness is the opposite of working sloppy, without focus, and only considering the surface of the task but not the details. Participant G2-P2 emphasized issues that can arise from a lack of conscientiousness as he stated: "So if you are working imprecisely [. . .], that will lead to problems fast." With a mean of coded passages $\bar{x} = 15.67$ and a small standard deviation across coders ($SD = 2.31$), coders highly agreed on this factor. The importance of conscientiousness came up several times in each group.

> *I mean it is really hard to always think about everything and to fill every possible security gap. [. . .] But something that could help with that is to have actual perfectionists in the team who strive to make everything 100% perfect on their own accord.* [G3-P1]

The factor *Conscientiousness* is most closely related to the conscientiousness factor of the FFM (Fiske, 1949; Goldberg, 1993; McCrae and John, 1992), which is characterized by orderliness, dutifulness, and self-discipline. In the Prevent-Model, this factor additionally strongly emphasizes detail-mindedness and meticulousness. Research on the related FFM conscientiousness factor reveals medium sized but consistent effects on occupational variables and it suggests this trait to be the best predictor for work performance among the FFM factors (Wilmot and Ones, 2019; Zell and Lesick, 2021), especially when it occurs in combination with agreeableness (Witt et al., 2002). *Take-away: Avoid putting time pressure on engineers whenever possible — making them hurry could inhibit them to work in sufficient detail.*

4.1.11. *Analytical thinking*

Analytically thinking engineers can dissect information and infer proper conclusions based on the given elements of evidence while following logical rules. This skill includes taking into account a variety of informative elements in order to come up with a logically correct conclusion and/or solution. Analytically thinking engineers like to puzzle over problems and tasks and take great effort to solve them. Participant G3-P1 explained why he primarily engaged with S&S in robotic systems with his proclivity to solve problems: "For me, the matter was really, really exciting. In general. Analyzing an entire system and puzzling over how to make it secure." Analytical thinking was coded by every coder and it emerged in every FG.

> *You really have to proceed analytically, that is highly important.* [G2-P2]

In literature, this conglomerate of problem solving, logical thinking, and reasoning skills sometimes also is referred to as critical thinking ("the use of those cognitive skills or strategies that increase the probability of a desirable outcome. It is used to describe thinking that is purposeful, reasoned, and goal directed" Halpern, 2014, p. 8). In problem solving skills is expressed repeatedly (Groeneveld et al., 2020; Sedelmaier and Landes, 2014; Serna and Serna, 2015). Certain questioning techniques seem to improve this higher-order thinking skill (Snyder and Snyder, 2008; Buchanan Hill, 2016). By asking independent, cognitively demanding questions prior to the actual task one can effectively activate analytical thinking (Baghaei Lakeh and Ghaffarzadegan, 2015). *Take-away: When instructing tasks, administer them with a variety of questions to enhance analytical thinking (e. g., What does this problem imply? What is the core structure of this problem? Why do you solve that task in that exact way?).*

4.1.12. *(Self-)Management*

(Self-)Management is the ability to manage one's tasks, even if no structured plan is enforced by leaders. This factor was coded only once in the first FG and was identified only by one coder, while the other two coders did not acknowledge (self-)management as a factor on its own. Therefore, the credibility of this factor is constrained.

> *[. . .] to make their time schedule and they're planning for themselves.* [G1-P3]

Still, the positive effect of self-management is already being discussed for decades (Manz and Sims, 1980; Snyder et al., 1983). Research suggests that self-management abilities are positively related to job performance (Gerhardt et al., 2009) – an effect which also was shown to be induced by self-management training (Frayne and Geringer, 2000). The time management aspect of this factor was found to become increasingly relevant: The association between time management skills and job performance has increased over the past 20 years (Aeon et al., 2021). This evidence suggests that time management is advancing as a key factor for occupational performance in nowadays' working environments. *Take-away: If structured working is required, but employees' self-management skills are low, invite them to participate in self-management training or provide them more structure and directives as a leader.*

4.1.13. Feasibility orientation

Feasibility orientation describes an accurate perception and judgment of reality. Real-life conditions, actual limitations, and feasibilities are assessed and judged validly. Practicability is neither under- nor overestimated; the realistic engineer does not daydream on unimplementable solutions. Although all of the coders perfectly agreed 100% on the coding of the corresponding text passage, this factor was mentioned only once by a single participant in group 1. Therefore, reliability of *Feasibility Orientation* should be interpreted with caution.

> *I would like to add: realistic. Actually more realistic. As a personality trait. [. . .] Yeah, and not being in a daydream.* [G1-P4]

Still, Groeneveld et al. (2021) discuss "the right combination between creativity and critical thinking, taking into account the context and constraints of the problem" [p. 6] in the course of their research on creativity in software engineering. In their interviews, participants emphasized the importance of solutions being *realistic and feasible*, even if they are creative. *Take-away: Especially engineers who have highly innovative, creative, and novel solutions also should take the approaches' feasibility under consideration. Even the most valuable ideas might have limitations; examine their practicability carefully.*

4.1.14. Acceptance of mistakes

To accept mistakes means to perceive past mistakes not negatively and to learn from them. It includes being able to take criticism as something constructive and appreciating other people pointing out these mistakes. Engineers who accept their mistakes are not ashamed or afraid of making errors; they perceive feedback on errors as a chance to improve in the future. They are honest about past mistakes. Misdoings are not being hidden but communicated in order to solve the problem. Respondent G3-P2 related his experiences of his own mistakes: "Often enough it happens to me that somebody else spotlights something that makes me facepalm on how I could have possibly overlooked that. But well, that is human. And one simply needs the readiness to accept these things". Intercoder agreement on the relevance of accepting one's mistakes was high. On average, this factor was coded seven times by each coder ($SD = 1$). Additionally, it arose in every group interview.

> *You have to, let's say, accept that there will be mistakes. And there will be people that point out those mistakes. But you don't have to be angry with them. [. . .] [It should] motivate you to take it: well, there's a chance to make something better.* [G1-P3]

The beneficial behavior of *Acceptance of Mistakes* is similar to the concept of *error competence*, describing "the knowledge of and capability to deal with errors *immediately*, when they occur" (Rybowiak et al., 1999, p. 532). Error competence is related to learning from errors and reflection at work, which in turn fosters employees' competence (Hetzner et al., 2011). Still, error competence and its relation to reflective work behavior may not be considered in a vacuum. Hetzner et al. (2011) found psychological safety regarding colleagues and supervisors (both are social and organizational factors) to mediate the relationship between error competence and reflection. *Take-away: Aim for a climate of psychological safety and constructive feedback and error culture, so employees feel safe to admit mistakes. Engineers should perceive their mistakes as a chance to improve the system and a gain in experience.*

4.1.15. Awareness towards fallibility

Awareness towards fallibility is the knowledge about and awareness towards the fact that humans make mistakes. Engineers who are aware of their and others' fallibility work more carefully, ask others to review their own work, or review others' work meticulously, because they are aware that mistakes happen. It includes being aware of one's knowledge and skills and especially about the limitations of these. Engineers with this property are conscious that they do *not* know everything and that they are not experts in every field they have *some* knowledge about. The importance of being aware of human fallibility was substantiated by G2-P3, outlining that "nobody is almighty anyway". This topic emerged in FGs 1 and 2 but did not come up in the third group.

> *I think [. . .] people need to be aware that we're all humans and we do make mistakes and we need to be aware that mistakes will happen [. . .]. And if I'm writing code, I need to be perfectly aware that there will be errors in there which I simply cannot spot. No matter how much I'm testing. And that's a hard fact to accept maybe for some people.* [G1-P3]

For information security awareness training, one of the key strategies is to teach employees "to avoid overestimating their capabilities to mitigate security risks" (Aldawood and Skinner, 2019, p. 115), which Howard and Lipner (2006) also outline concerning the SDLC: "Acknowledging that there's a lot that they don't know about software security makes people willing to have their ideas, designs, code, and test plans reviewed by others who do understand the security issues in depth" (Howard and Lipner, 2006, p. 58). Research on human errors in software requirements specification revealed a strong negative relationship between the understanding of possible human errors and the likelihood of making such an error (Hu et al., 2018). This indicates that deep understanding of possible mistakes goes along with committing fewer mistakes. Also, *Error Anticipation* was found to correlate with the tendency to learn from errors ($r = 0.54$, $p < 0.01$) (Rybowiak et al., 1999, p. 542). *Take-away: As an engineer, do not overestimate your capabilities and acknowledge your fallibility — you may make a mistake. Therefore, re-examine your work carefully and let coworkers check your work. Organizational approaches to be implemented are the four-eyes principle, conveying information about common human errors to increase engineers' awareness towards fallibility, and comprehensive testing of developed systems.*

4.1.16. Ability to work in a team

The ability to work in teams is the ability to cooperate, collaborate, and communicate with others, as well as to give and receive feedback in a respectful manner. Especially in testing matters, the ability to work in a team was emphasized: "I am mainly thinking about testing matters, for example. Simply being content with somebody else testing the whole thing and reviewing it and being open to take criticism. [. . .] I think these things are always easier to solve in a team than completely alone" [G3-P2]. Each coder identified the ability to work in a team in each FG as being referred to as fostering S&S in robotic systems.

> *Is he cooperative? Is he able to communicate? And is he able to fit in with a team? [. . .] Because the more people are in a team, the more positions are represented.* [G2-P3]

Research on required capabilities in agile teams suggests that cooperation skills are of highest relevance (Vishnubhotla et al., 2021; Wood et al., 2013; Groeneveld et al., 2020). Taking a FFM perspective suggests agreeableness to be related to teamwork abilities. This is supported by correlations between *Knowledge, skill, and ability requirements for teamwork* and agreeableness (Stevens and Campion, 1994; Kichuk and Wiesner, 1996) as well as the moderating effect of agreeableness on job performance (Witt et al., 2002). Evidence suggests teamwork interventions (workshops, simulation-based teamwork training, and team reviews) to positively affect teamwork itself as well as team performance (McEwan et al., 2017), mostly by improving affective states and team processes. *Take-away: It is important that teams collaborate well. If they do not, do not hesitate to provide teamwork interventions and team building training.*

4.1.17. Compliance

Compliant engineers follow rules, processes, and guidelines given by the organization (including transorganizational standardized norms). Engineers fulfill their tasks the way they are supposed to and do not circumvent guidelines or neglect certain aspects, e. g., for the sake of comfort. As a negative example G2-P4 mentioned that "If someone does not abide to any structures at all [. . .] – I would argue that that is obstructive to S&S". The effect of compliance on S&S was identified three times on average across coders. This topic only emerged in the first two FGs, while participants in the third group did not discuss this factor.

> *Follow the guidelines, which are drawn by the organization. And follow them as a religion.* [G1-P4]

Frameworks on compliance to norms and standards exist (Linnenberg and Fay, 2018; Kempe and Massey, 2021; Castellanos-Ardila et al., 2021; Moyón et al., 2020). Evidence suggests that compliance in the sense of adherence to organizational rules is influenced by people's values and self-regulatory processes (Tyler et al., 2007). To increase employees' compliance with organizational rules, activate individuals' morality and values by emphasizing work cultures of individual responsibility (Tyler et al., 2007; Tyler, 2005). *Take-away: To increase compliance and commitment, align individuals' values with organizational values by transmitting organizational culture from the time of onboarding. Provide explanations, convey key values and their importance by rituals and symbols (e. g., S&S-meetings and -awards), and make sure leaders act as role models.*

4.2. Organizational factors

RQ2 focuses on factors regarding organizational structures that are advantageous for S&S features of robots during development. Leaders, HR personnel, and managers can provide beneficial working environments by fostering specific organizational structures, processes, corporate values, team composition, and aspects of organizational climate.

4.2.1. Implementation of structured processes

The organization enforces guidelines (e. g., transorganizational norms/standards or internal rules) and implements structured S&S-processes (such as DevSecOps (Mayoral-Vilches et al., 2020b), four-eyes principles, documentation of previous mistakes, or utilization of formal testing methods). One participant emphasized that "there is no 'single moment' to do S&S, but instead it ought to be part of the process continuously" [G2-P1], and he mentioned DevSecOps as an example for a process that incorporates security. While this factor was coded more often by the second coder than by the other two coders, all of them identified it to be discussed as an important factor in all of the three FGs. With a mean frequency of $\bar{x} = 21.33$, *Implementation of Structured Processes* is the most prominent organizational factor discussed in the interviews.

> *A company can pose a standard on the employees. For example [. . .], "we want you to follow the IEC 27 standard series", which basically means that there is a standardized process deployed for risk management and S&S evaluation. And this also includes documentation tools, feedback cycles, organizational awareness, and risk management.* [G1-P5]

According to Assal and Chiasson (2019), not having formal processes for S&S is one of the greatest software security deterrents. Rather than "along the way", these matters should be addressed by implementing structured processes – e. g., compliance to S&S and robotics standards (International Electrotechnical Commission, 2010, 2003; Institute of Electrical and Electronics Engineers [IEEE], 2015, 2021; International Organization for Standardization, 2021; International Electrotechnical Commission, 2019) or internal guidelines on usage of best practices (Assal and Chiasson, 2018b; Migues et al., 2020). Examples of such processes from the software engineering domain include Microsoft's SDLC (Howard and Lipner, 2006) or DevOps (Virmani, 2015; Ebert et al., 2016). Mayoral-Vilches et al. (2020b) proposed a security-wise improved version of DevOps, called DevSecOps (Mohan and Othmane, 2016), also to be applicable and desirable for robotics engineering. Implementing structured processes also includes guidelines on the usage of, e. g., four-eyes-principles, which shape S&S culture (Glesner et al., 2020). *Take-away: Create structures and explicit processes for S&S – implement developmental frameworks, best-practices, document previous issues, and make comprehensive testing mandatory.*

4.2.2. Resources

S&S need resources. A S&S-aware organization is willing to invest time and money in order to achieve safer and more secure robots. Engineers do not have to fit in S&S-tasks within the same time frame they have for "common" tasks. Time is being allocated for S&S-purposes, or even one or more persons are employed exclusively for S&S-matters. A conclusion drawn by G3-P1 encapsulated the essence of this factor: "So, simply more time and money have to be allocated in order to implement S&S". The prerequisite of allocation of resources was mentioned and coded several times in each of the groups and by each coder. Therefore, it can be regarded as a reliably identified factor.

> *As long as the matter is present in management and management is willing to support things like S&S – financially and with resources, so with persons and so on – this will impact quality of products.* [G3-P3]

Regarding security costs, estimates on additional efforts to develop software with a high level of security range from 20%–87% (Venson et al., 2019). Weir et al. (2021) identified a variety of security activities adopted by practitioners using data from the large security study "Building Security In Maturity Model"

(BSIMM; Migues et al., 2020). Most of the activities they identified pose additional effort in the process and thereby require additional time, personnel, or money (Howard and Lipner, 2006). *Take-away: S&S require additional effort and therefore resources. Allocate time or additional workforce in order to make a difference.*

4.2.3. Safety & security as corporate values

The organization states S&S as corporate values of their organization. These values are not only written in some code of conduct, but are actually implemented within the corporate culture and are lived at any stage of development. The organization asserts safe and secure products because this is perceived as valuable, meaningful, and/or there is an overall feeling of social responsibility. The values of S&S are not only promoted for advertising/economic (i. e., cosmetic) reasons. Within the organization, there is a structured perception of S&S. There are documents containing definitions and guidelines (e. g., conducting risk analyses), and S&S are continuously mentioned as worth pursuing. The management is informed about (inter)national standards and norms and deals with possibilities to deploy processes and methodologies to implement S&S. Although frequencies of coded passages vary between coders ($SD = 8.14$), all of them coded this factor repeatedly. The relevance of a company's perception of S&S as precious values and acting these out as such emerged in every FG.

> *But that has to be an organizational value, has to be lived within the organization. You can't just ascribe this to yourself and then disregard it in your daily routine. Instead, it really should be embedded in onboarding, in culture and so on: that it is important, that it has significance and that we as an organization commit to deliver it.* [G2-P1]

Shared organizational values and a common mission and vision significantly predict quality and productivity in software engineering companies (Jossy, 2007) and general performance in companies in other domains (Donker et al., 2008). Additionally, corporate culture, behavior, and philosophy positively impact the sustainability of corporate performance as well as product, process, and market innovation (Staub et al., 2016). Therefore, perceiving S&S as corporate values and living them as such throughout the organization (Tahaei and Vaniea, 2019) pays off economically. *Take-away: Emphasize S&S as organizational values and create a pertinent corporate culture, e. g., by highlighting, rewarding, and honoring corresponding efforts and achievements, having leaders and established employees act as role models, periodic meetings, and actively conveying these values to new employees.*

4.2.4. Explicit roles

Explicit S&S-roles are assigned. This might be implemented internally via establishment of specific teams, single dedicated safety experts or security architects, or persons being responsible for these matters for a fixed share of their employment. Also, external experts or companies could be hired for a limited time span in order to create a road map for S&S. Respondent G2-P3 argued that "most of the time security is treated as an orphan, which somebody dabbles in with 30%, who actually has no depth in that or just has read up on it. And it should not be that way". For the factor *Explicit Roles* there was high congruence of frequencies of coded passages per coder ($\bar{x} = 7.33$, $SD = 0.58$) and the need for assigning explicit S&S-roles was mentioned in every group discussion.

> *If somebody in the company engages specifically with the matter of security or is allowed to do so as part of his working hours, that already is a great first step.* [G3-P3]

Although there seems to be a need for every single developer to be aware of S&S to a certain extent, explicit roles for these matters should be assigned (e. g., single security engineers, also called "champions" Jaatun and Soares Cruzes, 2021; Boström et al., 2006, or a group of them, also referred to as a "satellite" Migues et al., 2020; Tahaei and Vaniea, 2019). For software security in larger companies, a security advisor, a security team, and a contact person in the development team should be assigned (Howard and Lipner, 2006). Smaller companies with fewer employees could hire external security engineers/architects/testers (Tahaei and Vaniea, 2019). *Take-away: Create roles (part-time, full-time, or entire teams) who are explicitly in charge of S&S. Small companies may consider consulting external experts.*

4.2.5. Customer consideration

Some customers implicitly assume that the product they are interested in is safe and secure; thus, they might not explicitly mention these non-functional requirements. S&S, however, are not inherent to all robotic systems. Therefore, the actual S&S levels of the robot should be communicated in the course of requirements analysis in order to prevent misunderstandings or improper usage. If the company manages to promote S&S as features, it will become a formal requirement of the customer. Additionally, the product's usability should be encouraged during development as this might help to prevent issues later on. *Customer Consideration* was coded only by one of the coders, who identified this factor several times in each FG. As two coders did not code this factor at all, reliability cannot be claimed for it.

> *Robot manufacturers always argue "No customer has ever asked about security — why should we do that?". And the fun thing is, if you're talking to end-users, operators, they say: "We assume that it is secure. We do not even ask ourselves that question, we take that for granted". Yes, and that is a certain discrepancy between suppliers and consumers.* [G2-P1]

Customer Consideration relates to accurate identification of customers' (S&S) demands, identification of security objectives, and therefore related requirements (Tondel et al., 2008). As repeatedly mentioned in interviews with engineers (Assal and Chiasson, 2018b), security sometimes is neglected if respective properties are not formally stated as requirements. While Bartsch (2011) identified a lack of literature on customer involvement in regards to security, his interviewees emphasized scarce customer awareness as a challenge and discussed close customer integration as a way to mitigate it (Bartsch, 2011). *Take-away: Impart the importance of S&S to customers comprehensibly, so that they realize these matters' necessity and therefore are more likely to be willing to invest more in higher quality.*

4.2.6. Team diversity

Teams are diverse regarding technical expertise, soft skills (communication/presentation skills, conscientiousness, risk awareness, experience, etc.), professional background, and/or demographic variables. G2-P2 explained that as "you will never find all of these [required] traits within one single person", a company "will not get around a team" with a variety of personality traits (e. g., assertive, analytic, visionary, creative). *Team Diversity* occurred in every FG and was identified repeatedly by every coder.

> *So, in safety critical areas, it can be made sure that the people involved have knowledge in various backgrounds. And that the team is diverse [. . .] not only in the sense of education, but also in the sense of risk awareness and risk seeking behavior such that a balanced view is obtained.* [G1-P5]

A team can be diverse on a variety of levels (skills, personality, demographics, attitudes, etc.). While research on the relationship between team diversity in a broad sense and team performance/effectiveness shows controversial, complex, and inconsistent results (Meyer, 2017; Webber and Donahue, 2001; Campion et al., 1993; Liang et al., 2007; Horwitz, 2005), positive effects of increased cross-functionality in teams are found (Horwitz, 2005; Blindenbach-Driessen, 2015). Furthermore, functional diversity in teams seems to increase technical quality via increased external communication (Keller, 2001). *Take-away: Make sure to have a variety of perspectives within a team. As single persons are likely not to be high on each of the individual-level factors of the* Prevent*-Model, assemble multiple persons within a team who complement each other regarding these characteristics.*

4.2.7. *Employee training & development*

Organizations continuously offer training and employee development, inform employees about S&S by offering definitions of the terms (e. g., via intranet), and keep them up to date with current issues and new solutions. Engineers are being familiarized with the subjects of S&S on a recurrent basis or at least are given sufficient time to keep themselves up to date. Passages describing the need for training and development were coded 8.33 times on average per coder and every coder identified this factor to be named in each FG.

> *[. . .] this is a really fast-moving and constantly changing area, in which new issues arise incessantly. That is, further training of employees in this area is important and you have to be willing to invest a bit.* [G3-P2]

This factor is consistent with software engineering research findings, suggesting organizational learning to be a significant predictor of product quality (Jossy, 2007). In general, the positive impact of personnel development on individual and organizational performance is well established (Aguinis and Kraiger, 2009; Salas et al., 2012). Training might be particularly necessary with regards to cybersecurity, where new vulnerabilities are identified continuously (Mayoral-Vilches et al., 2019; Mitre Corporation, 1999). *Take-away: Directly inform engineers about new S&S developments on a recurrent basis, provide them time within their working hours to keep themselves up to date, and deliver training to improve their skills as well as to raise awareness.*

4.2.8. *Task allocation: Matching tasks & strengths*

Specific tasks are assigned to specific people who have the necessary skills for that task. Engineers are not given assignments at random, but personal and technical strengths, engineers' currently available resources, and maybe even their preferences are taken into account when distributing new tasks. Recruiters do not only assess technical knowledge and work experience, but they consider a potential employee's personality and intentions and assess how they fit into the existing team. A participant in group 2 expressed his opinion on the matching of tasks and strengths as follows: "It's no use to train the software developer for everything, then he will not do well anywhere. Instead, he should do what he has the ability to" [G2-P3]. Similarly, this factor was mentioned in the first and third FG, and respective passages were coded by each coder.

> *Evaluation of strengths and weaknesses of their employees, not for the sake of getting rid of the weaknesses, but also knowing whose strengths are where and then assigning duties according to the strengths of people [. . .]. So putting everyone on everything more or less takes everything to the average and not everything to the best possible.* [G1-P3]

A variety of research deals with models to automatically identify the best person-task fits (e. g., Aslam and Ijaz (2018), Duggan et al. (2004), Coelho et al. (2019) and Otero et al. (2009)), also showing benefits regarding costs due to higher efficiency (Chiang and Lin, 2020). Specifically targeting agile software development teams, Lin et al. (2014) identified competency as positively related to the number of assigned tasks ("workload"). Higher competency also leads to a higher probability of finishing a task on time (Lin et al., 2014); this suggest a higher efficiency in more competent individuals, highlighting the benefit of sophisticated task allocation. *Take-away: Make sure to know your employees' strengths, weaknesses, interests, and available resources in order to enable the best possible, most successful, and most efficient fit between persons and tasks.*

4.2.9. *Degrees of freedom*

Engineers have autonomy in *how* they solve a problem as long as they *solve* it appropriately. To a certain extent, they are allowed to develop individual, novel, creative solutions and to arrange their schedules individually. The advantages of high degrees of freedom in how engineers complete their tasks were discussed only in one FG but were coded by all three coders.

> *You can give them a kind of a framework of what you would need. But within that kind of frame, that goal description, leave some space and degree of freedom for people to come up with their own creative solutions.* [G1-P3]

The importance of employees' degrees of freedom might be illustrated best by inspecting the positive effects of job autonomy. Hackman and Oldham (1976, p. 258) define autonomy as "the degree to which the job provides substantial freedom, independence, and discretion to the individual in scheduling the work and in determining the procedures to be used in carrying it out". According to the JCM (Hackman and Oldham, 1976), jobs characterized by high autonomy lead to higher degrees of experienced responsibility for the outcomes of the conducted work, which in turn positively affects personal as well as work outcomes (high intrinsic motivation, high work performance, high job satisfaction, low absenteeism). Additional evidence suggests intrinsic motivation to mediate the positive relationship between job autonomy and job performance (Joo et al., 2010). *Take-away: Provide engineers with as much freedom and autonomy as possible for their respective position.*

4.2.10. *Feedback & error culture*

Mistakes are treated in a positive way and constructive feedback is provided. Routinely, produced code is reviewed and double-checked, but detected errors are not being punished. The overall climate regarding reporting errors is positive; hence, engineers feel free to report errors they found or mistakes they made. The positive impact of *Feedback & Error Culture* was the most substantial factor in the cluster of corporate *Collaboration* ($\bar{x} = 8.67$ coded passages, $SD = 2.52$).

> *Let's look for establishment of feedback culture, something like that. So is the company altogether feedback friendly or not? Are there any feedback processes that are established?* [G1-P2]

The job characteristic of feedback ("The degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance." Hackman and Oldham, 1976, p. 258) showed to be related to higher-rated work effectiveness (Hackman and Oldham, 1976). Especially when feedback is combined with instructions to reflect on it, future task performance is increased (Anseel et al., 2009). Steelman et al. (2004, p. 166) conceptualize feedback environment as "the contextual aspects of day-to-day supervisor-subordinate and coworker-coworker feedback processes" and this environment showed to be related to job satisfaction (Anseel and Lievens, 2007). Error culture comprises "organizational practices related to communicating about errors, to sharing error knowledge, to helping in error situations, and to quickly detecting and handling errors" (van Dyck et al., 2005, p. 1229). Research shows this constructive environment to relate to better firm performance (van Dyck et al., 2005), especially when accompanied by a climate of psychological safety (Javed et al., 2020). *Take-away: Leaders and colleagues should provide (constructive) feedback to employees regularly. Foster a climate of psychological safety, an error culture in which mistakes are openly dealt with, and a solution-focused attitude towards mistakes.*

4.2.11. Intraorganizational exchange

Within the organization, exchange between different departments and teams is facilitated (e. g., by team meetings, open offices, presentations, or newsletters). To a certain extent, there is transparency of each group's progress, goals, workflows, and problems. Each person has a vague understanding of what all the other teams are working on and is informed about development steps and procedures along the entire developmental cycle. The work culture makes it effortless to get in touch with other employees and cooperation between teams is facilitated. Teams and departments are not fully separated from each other. *Intraorganizational Exchange* was not mentioned in FG 3, but for groups 1 and 2 on average $\bar{x} = 6.67$ ($SD = 4.04$) passages were identified by each coder.

> *People should not sit within their silos and only witness their own world, but one should simply dovetail the areas a little more. Everybody should kind of get some insights into the worlds of the others.* [G2-P3]

Intraorganizational Exchange, also referred to as internal communication, is the "strategic management of interactions and relationships between stakeholders within organizations across a number of interrelated dimensions [. . .]." (Welch and Jackson, 2007, p. 184). Welch and Jackson (2007) summarize a variety of literature and outline the objectives of internal communication to increase engagement and commitment within an organization. Managing and sharing knowledge within an organization via meetings, newsgroups, and forms of collaboration between employees is demanded, especially in non-routine and interdependent tasks (Bhatt, 2002) as in robotic systems development. The related positive effects of sharing and managing knowledge within an organization are shown to manifest themselves in increased innovation and organizational performance (Alaarj et al., 2016; Koohang et al., 2017; Yusr et al., 2017). Also "boundary spanning" (interpersonal relationships between persons from different formal working groups) within the organization is shown to positively impact performance and goal achievement on a team and organizational level (Marrone, 2010). *Take-away: Coordinate different teams' and departments' tasks, facilitate information exchange between these, and make sure teams have an overview of other teams' goals and responsibilities.*

4.2.12. Fostering & facilitating collaboration

Encourage employees to exchange and collaborate. The organization promotes team meetings, team-building activities, and formal as well as informal (e. g., coffee breaks) exchange of ideas, thoughts, concerns, or current topics. Engineers are encouraged to review others' work (e. g., check code) and follow four-eyes-principles. The management coordinates and/or synchronizes the work of different teams. Young employees learning from more experienced engineers is welcomed and facilitated (e. g., mentoring programs). This factor was only discussed in groups 1 and 2, but not in the third group. Still, there was agreement of coders on the mention of *Fostering & Facilitating Collaboration* as positively influencing S&S.

> *What might also prove successful [. . .] would be a mentoring program. Meaning, really forming a dyadic team of a young and an old developer, of whom can be learned. And perhaps within a rotation principle, [. . .] so he can become acquainted with new areas again and again.* [G2-P3]

Stronger social ties between employees foster (knowledge) resource flows, which in turn increase organizational performance (Maurer et al., 2011). Also, trust partially mediates the relationship between knowledge processes and performance positively (Alaarj et al., 2016) and social cohesion was found to increase perceived team effectiveness, team satisfaction, and team viability (Tekleab et al., 2009). One possible and effective form of collaboration are mentoring programs. Perceived instrumental and psychosocial support, as well as relationship quality between mentors and mentees were investigated in a meta-analysis and were found to correlate with self-efficacy, situational satisfaction, strain reduction, and many more desirable aspects (Eby et al., 2013). *Take-away: Encourage collaboration between individual members of the organization e. g., by mentoring programs, team-building interventions, or appreciation of informal exchange regarding current matters.*

4.2.13. Participation

Make employees' voices heard. The organization perceives experienced employees as experts, who give valuable advice, and enables participation in decision-making. For example, by using innovation boards everybody's ideas can be heard and everybody can participate in the process of improvement. The organization values employees' S&S efforts and rewards engineers' efforts in some way (e. g., money or praise). The importance of participation of employees and of acknowledging their expertise was expressed in terms of "flat hierarchies, because [. . .] we need many experts from different spheres. And you need to have all of these voices heard" [G1-P5]. This factor was mentioned only in FGs 1 and 3.

> *Perhaps the point is, that if you are in a leading position, you have to be capable of acknowledging the expertise of your subordinates and then also take the things they say seriously — because that's the reason why you've hired experts.* [G3-P1]

Participation in decision-making correlates moderately but significantly with productivity, employee satisfaction, and managers' judgments of effectiveness (Campion et al., 1993). Also, if leadership is shared within a team and all team members bear leadership responsibilities, the team's effectiveness tends to increase (Daspit et al., 2013). Wohlgemuth et al. (2019) identify employee participation as a significant predictor of dynamic capability ("The ability to integrate, build, and reconfigure internal and external competencies to address rapidly changing environments [. . . and to] achieve new and innovative forms of competitive advantage [. . .]" Teece et al., 1997, p. 516). Employee participation enhances this type of organizational flexibility (Wohlgemuth et al., 2019), which is particularly important in the robotics domain (e. g., due to continuously changing security demands and disruptive innovations). *Take-away: Give employees an audience and listen to these experts' valuable advice. Ask employees about their opinions, suggestions, and needs and include them in decision-making processes e. g., by participative management or shared leadership.*

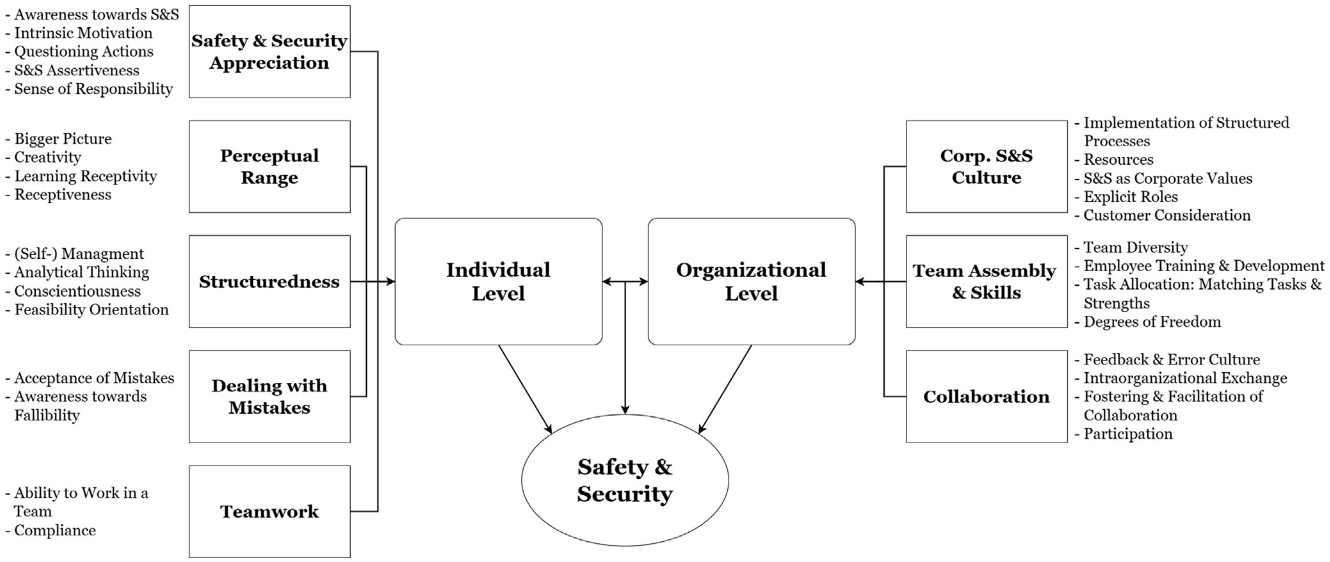

Fig. 1. The PREVENT-Model of human and organizational factors fostering safe and secure robotic systems engineering.

5. Discussion

The proposed PREVENT-Model (see Fig. 1) provides a framework of individual attributes and organizational structures that enable S&S by design in robotic systems engineering. Although the model was originally developed to identify relevant factors in robotics engineering, PREVENT might be applicable to the broader domain of software engineering for three reasons. Firstly, although robots operate in the physical world using hardware (e. g., robotic arms performing operations on industrial workpieces), they rely on highly sophisticated software to fulfill their purposes. Therefore, software development largely shapes the process of robotic systems engineering. These substantive overlaps can be reasonably assumed to make the proposed framework applicable to the software engineering domain. Secondly, although safety aspects are less crucial for (non-robotic) software engineering, the cybersecurity aspects facilitated by the presented model are central for responsible software. Thirdly, many of the identified factors are congruent with existing evidence from software engineering research, strongly enhancing the possibility of the PREVENT-Model to be expanded to this broader domain.

5.1. Practical implications

With the PREVENT-Model, we identified a variety of factors facilitating responsible engineering. We furthermore present practical implications and possible courses of action for engineers, leaders, management, and HR personnel for each factor in the form of *Take-aways*. The PREVENT framework enables practitioners to pay special attention to these factors crucial for developing safe and secure systems. The *Take-aways* provide practical suggestions to take action. Roboticists should consider the model's factors when allocating tasks, communicating within and outside the organization, during recruitment, in employee training and development, and many other daily practices to prevent hazardous systems from being developed. The model complements adherence to technical S&S standards, norms, and best practices and serves to elicit precautionary measures on a work and organizational behavioral as well as on a psychological level. With that, the PREVENT-Model helps practitioners to take evidence-based action towards fruitful preconditions for developing safe and secure robotic systems.

5.2. Limitations & future directions

The current study highlights work and organizational behaviors and psychological aspects in the engineering process and does not take into consideration technical skills or requirements imposed by laws or norms. The PREVENT-Model is deliberately limited to soft, managerial, and non-technical aspects, perceiving technical expertise as a prerequisite and transorganizational regulation as the overall frame for S&S. We argue that such "hard factors" are the foundation of safe and secure systems; the PREVENT-Model valuably complements them with human factors adding to development of responsible technology.

Qualitative studies are typically limited with respect to the generalizability of results. Therefore, the priority and importance of the identified factors cannot be derived from their frequencies reported here. Still, the resulting category system seems saturated as no new factors emerged in the third FG and research shows that even two to three FGs can yield great amounts of saturation (Guest et al., 2017). Moreover, we assume the model to be valid to a great extent, as the conducted study comprised three validation steps. Firstly, three distinct and independent FGs were conducted. Secondly, we examined the resulting category system using communicative validation and by calculating indices of agreement among coders. Thirdly, the model is in alignment with existing research findings. Therefore, we are confident about the validity of the model and we assume the PREVENT factors to positively impact S&S.

Future research might focus on empirical validation of the model. A first step to do so is the development of an assessment tool to evaluate the status quo within a company in order to identify malfunctioning structures. Thereupon, researchers can evaluate the prevalence of each factor as well as their distinct magnitude of impact on systems' S&S.

6. Conclusion

With the rise of robotic systems, safety and security are becoming increasingly vital to ensure in order to prevent economic damage or even human harm. The present study collected systematically and empirically individual human and organizational factors influencing safety and security in the development of robotic systems. The resulting PREVENT-Model integrates many aspects that are discussed and shown to be influential in related realms into a coherent, comprehensive model. Human resource managers, individual engineers, leaders, and other organizational members may utilize the presented framework to focus on influential human aspects fostering the responsible development of robotic technologies.

Research data

To ensure the anonymity of the focus group participants, the interview transcripts cannot be disclosed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author thanks the respondents of the focus group interviews for participating in the study. Special thanks are owed to the research team Linda Maurer and Anna Schultz, who conducted qualitative content analytical coding of the interview transcripts. This work was supported by the Karl-Popper-Kolleg SEEROSE (Responsible, Safe, and Secure Robotic Systems Engineering) at the University of Klagenfurt. The author especially thanks Larissa Krainer, Martin Pinzger, and Rainer W. Alexandrowicz for their advisory support.

Appendix A. Supplementary data

The interview guidelines used in the focus group interviews can be accessed at https://doi.org/10.1016/j.jss.2022.111548.

References

Aeon, B., Faber, A., Panaccio, A., 2021. Does time management work? A meta-analysis. PLoS One 16 (1), e0245066. http://dx.doi.org/10.1371/journal.pone.0245066.

Aguinis, H., Kraiger, K., 2009. Benefits of training and development for individuals and teams, organizations, and society. Annu. Rev. Psychol. 60, 451–474. http://dx.doi.org/10.1146/annurev.psych.60.110707.163505.

Alaarj, S., Abidin-Mohamed, Z., Bustamam, U.S.B.A., 2016. Mediating role of trust on the effects of knowledge management capabilities on organizational performance. Procedia - Soc. Behav. Sci. 235, 729–738. http://dx.doi.org/10.1016/j.sbspro.2016.11.074.

Aldawood, H., Skinner, G., 2019. Challenges of implementing training and awareness programs targeting cyber security social engineering. In: 2019 Cybersecurity and Cyberforensics Conference (CCC). IEEE, pp. 111–117. http://dx.doi.org/10.1109/CCC.2019.00004.

Anseel, F., Lievens, F., 2007. The long-term impact of the feedback environment on job satisfaction: A field study in a belgian context. Appl. Psychol. 56 (2), 254–266. http://dx.doi.org/10.1111/j.1464-0597.2006.00253.x.

Anseel, F., Lievens, F., Schollaert, E., 2009. Reflection as a strategy to enhance task performance after feedback. Organ. Behav. Human Decis. Process. 110 (1), 23–35. http://dx.doi.org/10.1016/j.obhdp.2009.05.003.

Anu, V., Hu, W., Carver, J.C., Walia, G.S., Bradshaw, G., 2018. Development of a human error taxonomy for software requirements: A systematic literature review. Inf. Softw. Technol. 103, 112–124. http://dx.doi.org/10.1016/j.infsof.2018.06.011.

Anu, V., Sultana, K.Z., Samanthula, B.K., 2020. A human error based approach to understanding programmer-induced software vulnerabilities. In: 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, Coimbra, Portugal, pp. 49–54. http://dx.doi.org/10.1109/ISSREW51248.2020.00036.

Aslam, W., Ijaz, F., 2018. A quantitative framework for task allocation in distributed agile software development. IEEE Access 6, 15380–15390. http://dx.doi.org/10.1109/ACCESS.2018.2803685.

Assal, H., Chiasson, S., 2018a. Motivations and amotivations for software security. In: USENIX Symposium on Usable Privacy and Security (SOUPS).

Assal, H., Chiasson, S., 2018b. Security in the software development lifecycle. In: Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018). USENIX Association, Baltimore, MD, pp. 281–296.

Assal, H., Chiasson, S., 2019. '*Think secure from the beginning*': A survey with software developers. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, Glasgow Scotland Uk, pp. 1–13. http://dx.doi.org/10.1145/3290605.3300519.

Baghaei Lakeh, A., Ghaffarzadegan, N., 2015. Does analytical thinking improve understanding of accumulation? Syst. Dyn. Rev. 31 (1–2), 46–65. http://dx.doi.org/10.1002/sdr.1528.

Bartsch, S., 2011. Practitioners' perspectives on security in agile development. In: 2011 Sixth International Conference on Availability, Reliability and Security. IEEE, pp. 479–484. http://dx.doi.org/10.1109/ARES.2011.82.

Bass, B.M., 1985. Leadership: Good, better, best. Organ. Dyn. 13 (3), 26–40. http://dx.doi.org/10.1016/0090-2616(85)90028-2.

Bermudez, A., 2015. Four tools for critical inquiry in history, social studies, and civic education. Rev. Estud. Soc. (52), 102–118. http://dx.doi.org/10.7440/res52.2015.07.

Bhatt, G.D., 2002. Management strategies for individual knowledge and organizational knowledge. J. Knowl. Manage. 6 (1), 31–39. http://dx.doi.org/10.1108/13673270210417673.

Bielefeldt, A., Canney, N., 2014. Social responsibility attitudes of first-year engineering students and the impact of courses. In: 2014 ASEE Annual Conference & Exposition Proceedings. ASEE Conferences, pp. 24.1089.1–24.1089.16. http://dx.doi.org/10.18260/1-2--23022.

Bielefeldt, A.R., Canney, N.E., 2016. Changes in the social responsibility attitudes of engineering students over time. Sci. Eng. Ethics 22 (5), 1535–1551. http://dx.doi.org/10.1007/s11948-015-9706-5.

Blindenbach-Driessen, F., 2015. The (in)effectiveness of cross-functional innovation teams: The moderating role of organizational context. IEEE Trans. Eng. Manage. 62 (1), 29–38. http://dx.doi.org/10.1109/TEM.2014.2361623.

Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., Kruchten, P., 2006. Extending XP practices to support security requirements engineering. In: Bruschi, D., de Win, B., Monga, M. (Eds.), Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems - SESS '06. ACM Press, New York, New York, USA, p. 11. http://dx.doi.org/10.1145/1137627.1137631.

Buchanan Hill, J., 2016. Questioning techniques: A study of instructional practice. Peabody J. Educ. 91 (5), 660–671. http://dx.doi.org/10.1080/0161956X.2016.1227190.

Campion, M.A., Medsker, G.J., Higgs, A.C., 1993. Relations between work group characteristics and effectiveness: Implications for designing effective work groups. Pers. Psychol. 46 (4), 823–847. http://dx.doi.org/10.1111/j.1744-6570.1993.tb01571.x.

Canney, N., Bielefeldt, A., 2015. A framework for the development of social responsibility in engineers. Int. J. Eng. Educ. 31, 414–424.

Canney, N.E., Bielefeldt, A.R., 2016. Validity and reliability evidence of the engineering professional responsibility assessment tool. J. Eng. Educ. 105 (3), 452–477. http://dx.doi.org/10.1002/jee.20124.

Castellanos-Ardila, J.P., Gallina, B., Governatori, G., 2021. Compliance-aware engineering process plans: the case of space software engineering processes. Artif. Intell. Law 29 (4), 587–627. http://dx.doi.org/10.1007/s10506-021-09285-5.

Chiang, H.Y., Lin, B.M.T., 2020. A decision model for human resource allocation in project management of software development. IEEE Access 8, 38073–38081. http://dx.doi.org/10.1109/ACCESS.2020.2975829.

Cilliers, F., Greyvenstein, H., 2012. The impact of silo mentality on team identity: An organisational case study. SA J. Ind. Psychol. 38 (2), 75–84. http://dx.doi.org/10.4102/sajip.v38i2.993.

Coelho, F.D., Reis, R.Q., de Souza, C.R.B., 2019. A genetic algorithm for human resource allocation in software projects. In: 2019 XLV Latin American Computing Conference (CLEI). IEEE, pp. 01–08. http://dx.doi.org/10.1109/CLEI47609.2019.235055.

Conger, A.J., 1980. Integration and generalization of kappas for multiple raters. Psychol. Bull. 88 (2), 322–328. http://dx.doi.org/10.1037/0033-2909.88.2.322.

Cruz, S., Da Silva, F.Q., Capretz, L.F., 2015. Forty years of research on personality in software engineering: A mapping study. Comput. Human Behav. 46, 94–113. http://dx.doi.org/10.1016/j.chb.2014.12.008.

Daspit, J., Justice Tillman, C., Boyd, N.G., Mckee, V., 2013. Cross–functional team effectiveness: An examination of internal team environment, shared leadership, and cohesion influences. Team Perform. Manage. Int. J. 19 (1/2), 34–56. http://dx.doi.org/10.1108/13527591311312088.

De O. Melo, C., Cruzes, S., Kon, F., Conradi, R., 2013. Interpretative case studies on agile team productivity and management. Inf. Softw. Technol. 55 (2), 412–427. http://dx.doi.org/10.1016/j.infsof.2012.09.004.

Denning, T., Matuszek, C., Koscher, K., Smith, J.R., Kohno, T., 2009. A spotlight on security and privacy risks with future household robots: Attacks and lessons. In: Proceedings of the 11th International Conference on Ubiquitous Computing. UbiComp '09, Association for Computing Machinery, New York, NY, USA, pp. 105–114. http://dx.doi.org/10.1145/1620545.1620564.

Donker, H., Poff, D., Zahir, S., 2008. Corporate values, codes of ethics, and firm performance: A look at the Canadian context. J. Bus. Ethics 82 (3), 527–537. http://dx.doi.org/10.1007/s10551-007-9579-x.

Duggan, J., Byrne, J., Lyons, G.J., 2004. A task allocation optimizer for software construction. IEEE Softw. 21 (3), 76–82. http://dx.doi.org/10.1109/MS.2004.1293077.

Dysvik, A., Kuvaas, B., 2011. Intrinsic motivation as a moderator on the relationship between perceived job autonomy and work performance. Eur. J. Work Organ. Psychol. 20 (3), 367–387. http://dx.doi.org/10.1080/13594321003590630.

Ebert, C., Gallardo, G., Hernantes, J., Serrano, N., 2016. DevOps. IEEE Softw. 33 (3), 94–100. http://dx.doi.org/10.1109/MS.2016.68.

Eby, L.T.d.T., Allen, T.D., Hoffman, B.J., Baranik, L.E., Sauer, J.B., Baldwin, S., Morrison, M.A., Kinkade, K.M., Maher, C.P., Curtis, S., Evans, S.C., 2013. An interdisciplinary meta-analysis of the potential antecedents, correlates, and consequences of protégé perceptions of mentoring. Psychol. Bull. 139 (2), 441–476. http://dx.doi.org/10.1037/a0029279.

Englehardt, E., Werhane, P.H., Newton, L.H., 2021. Leadership, engineering and ethical clashes at boeing. Sci. Eng. Ethics 27 (1), 12. http://dx.doi.org/10.1007/s11948-021-00285-x.

Fiske, D.W., 1949. Consistency of the factorial structures of personality ratings from different sources. J. Abnorm. Psychol. 44 (3), 329–344. http://dx.doi.org/10.1037/h0057198.

Frayne, C.A., Geringer, J.M., 2000. Self-management training for improving job performance: a field experiment involving salespeople. J. Appl. Psychol. 85 (3), 361–372. http://dx.doi.org/10.1037/0021-9010.85.3.361.

Gall, H., 2008. Functional safety IEC 61508/IEC 61511 the impact to certification and the user. In: 2008 IEEE/ACS International Conference on Computer Systems and Applications. IEEE, pp. 1027–1031.

Gardner, B., 2014. Building an Information Security Awareness Program: Defending Against Social Engineering and Technical Threats. Elsevier Science, Burlington.

Gawlik, K., 2018. Focus group interviews. In: Ciesielska, M., Jemielniak, D. (Eds.), Qualitative Methodologies in Organization Studies: Volume II: Methods and Possibilities. Springer International Publishing, Cham, pp. 97–126. http://dx.doi.org/10.1007/978-3-319-65442-3_5.

Gerhardt, M., Ashenbaum, B., Newman, W.R., 2009. Understanding the impact of proactive personality on job performance. J. Leadersh. Organ. Stud. 16 (1), 61–72. http://dx.doi.org/10.1177/1548051809334192.

Glesner, C., van Oudheusden, M., Turcanu, C., Fallon, C., 2020. Bringing symmetry between and within safety and security cultures in high-risk organizations. Safety Sci. 132, 104950. http://dx.doi.org/10.1016/j.ssci.2020.104950.

Goldberg, L.R., 1993. The structure of phenotypic personality traits. Am. Psychol. 48 (1), 26–34. http://dx.doi.org/10.1037/0003-066X.48.1.26.

Grant, A.M., 2007. Relational job design and the motivation to make a prosocial difference. Acad. Manage. Rev. 32 (2), 393–417. http://dx.doi.org/10.5465/AMR.2007.24351328.

Groeneveld, W., Jacobs, H., Vennekens, J., Aerts, K., 2020. Non-cognitive abilities of exceptional software engineers. In: Zhang, J., Sherriff, M., Heckman, S., Cutter, P., Monge, A. (Eds.), Proceedings of the 51st ACM Technical Symposium on Computer Science Education. ACM, New York, NY, USA, pp. 1096–1102. http://dx.doi.org/10.1145/3328778.3366811.

Groeneveld, W., Luyten, L., Vennekens, J., Aerts, K., 2021. Exploring the role of creativity in software engineering. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). IEEE, pp. 1–9. http://dx.doi.org/10.1109/ICSE-SEIS52602.2021.00009.

Guest, G., Namey, E., McKenna, K., 2017. How many focus groups are enough? Building an evidence base for nonprobability sample sizes. Field Methods 29 (1), 3–22. http://dx.doi.org/10.1177/1525822X16639015.

Gwet, K.L., 2008. Computing inter-rater reliability and its variance in the presence of high agreement. Br. J. Math. Statist. Psychol. 61 (Pt 1), 29–48. http://dx.doi.org/10.1348/000711006X126600.

Gwet, K.L., 2019. irrCAC: Computing chance-corrected agreement coefficients (CAC). URL: https://CRAN.R-project.org/package=irrCAC, R package version 1.0.

Hackman, J., Oldham, G.R., 1976. Motivation through the design of work: test of a theory. Organ. Behav. Human Perform. 16 (2), 250–279. http://dx.doi.org/10.1016/0030-5073(76)90016-7.

Halpern, D.F., 1998. Teaching critical thinking for transfer across domains: Disposition, skills, structure training, and metacognitive monitoring. Am. Psychol. 53 (4), 449–455. http://dx.doi.org/10.1037/0003-066X.53.4.449.

Halpern, D.F., 2014. Thought and Knowledge: An Introduction to Critical Thinking, fifth ed. Psychology Press Taylor & Francis Group, New York and London.

Hartnell, C.A., Ou, A.Y., Kinicki, A., 2011. Organizational culture and organizational effectiveness: A meta-analytic investigation of the competing values framework's theoretical suppositions. J. Appl. Psychol. 96 (4), http://dx.doi.org/10.1037/a0021987.

Helsdingen, A.S., van den Bosch, K., van Gog, T., van Merriënboer, J.J.G., 2010. The effects of critical thinking instruction on training complex decision making. Human Factors 52 (4), 537–545. http://dx.doi.org/10.1177/0018720810377069.

Herlambang, M.B., Cnossen, F., Taatgen, N.A., 2021. The effects of intrinsic motivation on mental fatigue. PLoS One 16 (1), e0243754. http://dx.doi.org/10.1371/journal.pone.0243754.

Herzberg, F., 1968. One more time: How do you motivate employees? Harv. Bus. Rev..

Hetzner, S., Gartmeier, M., Heid, H., Gruber, H., 2011. Error orientation and reflection at work. Vocat. Learn. 4 (1), 25–39. http://dx.doi.org/10.1007/s12186-010-9047-0.

Horwitz, S.K., 2005. The compositional impact of team diversity on performance: Theoretical considerations. Hum. Resour. Dev. Rev. 4 (2), 219–245. http://dx.doi.org/10.1177/1534484305275847.

Howard, M., Lipner, S., 2006. The Security Development Lifecycle, Vol. 34. Microsoft Press, http://dx.doi.org/10.1007/s11623-010-0021-7.

Hu, W., Carver, J.C., Anu, V., Walia, G.S., Bradshaw, G.L., 2018. Using human error information for error prevention. Empir. Softw. Eng. 23 (6), 3768–3800. http://dx.doi.org/10.1007/s10664-018-9623-8.

Hüttermann, M., 2012. DevOps for Developers. In: The Expert's Voice in Web Development, A Press, Berkeley, CA, http://dx.doi.org/10.1007/978-1-4302-4570-4, URL: http://proquest.tech.safaribooksonline.de/9781430245698.

Institute of Electrical and Electronics Engineers [IEEE], 2015. IEEE Standard Ontologies for Robotics and Automation. IEEE Std 1872-2015, IEEE, pp. 1–60. http://dx.doi.org/10.1109/IEEESTD.2015.7084073.

Institute of Electrical and Electronics Engineers [IEEE], 2021. IEEE Ontological Standard for Ethically Driven Robotics and Automation Systems. IEEE Std 7007-2021, IEEE, pp. 1–119. http://dx.doi.org/10.1109/IEEESTD.2021.9611206.

International Electrotechnical Commission, 2003. Industrial communication networks-network and system security (IEC 62443).

International Electrotechnical Commission, 2010. Functional safety of electrical/electronic/programmable electronic safety-related systems (IEC 61508).

International Electrotechnical Commission, 2019. Risk management – risk assessment techniques.

International Organization for Standardization, 2021. URL: https://www.iso.org/committee/5915511/x/catalogue/.

Jaatun, M.G., Soares Cruzes, D., 2021. Care and feeding of your security champion. In: 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). IEEE, pp. 1–7. http://dx.doi.org/10.1109/CyberSA52016.2021.9478254.

Jamont, J.-P., Raievsky, C., Occello, M., 2014. Handling safety-related non-functional requirements in embedded multi-agent system design. In: Demazeau, Y., Zambonelli, F., Corchado, J.M., Bajo, J. (Eds.), Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection. Springer International Publishing, Cham, pp. 159–170.

Javed, B., Jalees, T., Herani, G.M., Rolle, J.-A., 2020. Error management culture and its impact on organizational performance: A moderated mediation model. J. Bus. Retail Manage. Res. 15 (01), http://dx.doi.org/10.24052/JBRMR/V15IS01/ART-03.

Joo, B.-K.B., Jeung, C.-W., Yoon, H.J., 2010. Investigating the influences of core self-evaluations, job autonomy, and intrinsic motivation on in-role job performance. Human Resour. Dev. Quart. 21 (4), 353–371. http://dx.doi.org/10.1002/hrdq.20053.

Jossy, M., 2007. The relationship of organisational culture with productivity and quality. Employee Relat. 29 (6), 677–695. http://dx.doi.org/10.1108/01425450710826140.

Judge, T.A., Rodell, J.B., Klinger, R.L., Simon, L.S., Crawford, E.R., 2013. Hierarchical representations of the five-factor model of personality in predicting job performance: integrating three organizing frameworks with two theoretical perspectives. J. Appl. Psychol. 98 (6), 875–925. http://dx.doi.org/10.1037/a0033901.

Kamuto, M.B., Langerman, J.J., 2017. Factors inhibiting the adoption of DevOps in large organisations: South African context. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, pp. 48–51. http://dx.doi.org/10.1109/RTEICT.2017.8256556.

Keller, R.T., 2001. Cross-functional project groups in research and new product development: Diversity, communications, job stress, and outcomes. Acad. Manage. J. 44 (3), 547–555. http://dx.doi.org/10.2307/3069369.

Kempe, E., Massey, A., 2021. Perspectives on regulatory compliance in software engineering. In: 2021 IEEE 29th International Requirements Engineering Conference (RE). IEEE, pp. 46–57. http://dx.doi.org/10.1109/RE51729.2021.00012.

Kichuk, S.L., Wiesner, W.H., 1996. Selection measures for a team environment: the relationships among the wonderlic personnel test, the neo-FFI, and the teamwork KSA test.

Kirschgens, L.A., Ugarte, I.Z., Gil-Uriarte, E., Rosas, A.M., Mayoral-Vilches, V., 2019. Robot hazards: from safety to security. CoRR abs/1806.06681.

Koohang, A., Paliszkiewicz, J., Goluchowski, J., 2017. The impact of leadership on trust, knowledge management, and organizational performance. Ind. Manage. Data Syst. 117 (3), 521–537. http://dx.doi.org/10.1108/IMDS-02-2016-0072.

Krippendorff, K., 2004. Content Analysis: An Introduction to Its Methodology, second ed. Sage, Thousand Oaks, Calif., URL: http://www.loc.gov/catdir/enhancements/fy0658/2003014200-d.html.

Kvale, S., 2007. Doing Interviews. In: The Sage Qualitative Research Kit, Sage Publications Ltd, Thousand Oaks, CA, http://dx.doi.org/10.4135/9781849208963.

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1), 159–174. http://dx.doi.org/10.2307/2529310.

Leonhardt, J.M., Keller, L.R., Pechmann, C., 2011. Avoiding the risk of responsibility by seeking uncertainty: Responsibility aversion and preference for indirect agency when choosing for others. J. Consum. Psychol. 21 (4), 405–413. http://dx.doi.org/10.1016/j.jcps.2011.01.001.

Leveson, N.G., 2016. Engineering a Safer World. MIT Press, Cambridge, Massachusetts.

Li, P.L., Ko, A.J., Zhu, J., 2015. What makes a great software engineer? In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, pp. 700–710. http://dx.doi.org/10.1109/ICSE.2015.335.

Liang, T.-P., Liu, C.-C., Lin, T.-M., Lin, B., 2007. Effect of team diversity on software project performance. Ind. Manage. Data Syst. 107 (5), 636–653. http://dx.doi.org/10.1108/02635570710750408.

Lin, J., Yu, H., Shen, Z., Miao, C., 2014. Studying task allocation decisions of novice agile teams with data from agile project management tools. In: Crnkovic, I., Chechik, M., Grünbacher, P. (Eds.), Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering. ACM, New York, NY, USA, pp. 689–694. http://dx.doi.org/10.1145/2642937.2642959.

Linnenberg, T., Fay, A., 2018. Software engineering for agent based energy systems. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). IEEE, pp. 174–180. http://dx.doi.org/10.1109/COASE.2018.8560487.

Lombard, M., Snyder-Duch, J., Bracken, C.C., 2002. Content analysis in mass communication: Assessment and reporting of intercoder reliability. Human Commun. Res. 28 (4), 587–604. http://dx.doi.org/10.1111/j.1468-2958.2002.tb00826.x.

Lutz, R.R., 2000. Software engineering for safety: a roadmap. In: Finkelstein, A. (Ed.), Proceedings of the Conference on the Future of Software Engineering. In: ACM Conferences, ACM, New York, NY, pp. 213–226.

Manz, C.C., Sims, H.P., 1980. Self-management as a substitute for leadership: A social learning theory perspective. Acad. Manage. Rev. 5 (3), 361–367. http://dx.doi.org/10.5465/AMR.1980.4288845.

Marrone, J.A., 2010. Team boundary spanning: A multilevel review of past research and proposals for the future. J. Manage. 36 (4), 911–940. http://dx.doi.org/10.1177/0149206309353945.

Maurer, I., Bartsch, V., Ebers, M., 2011. The value of intra-organizational social capital: How it fosters knowledge transfer, innovation performance, and growth. Organ. Stud. 32 (2), 157–185. http://dx.doi.org/10.1177/0170840610394301.

Mayoral-Vilches, V., 2020. Quality, safety and security in robotics. Cybersecurity and robotics. Thoughts and news on cybersecurity and robotics. URL: https://cybersecurityrobotics.net/quality-safety-security-robotics/.

Mayoral-Vilches, V., Carbajo, U.A., Gil-Uriarte, E., 2020a. Industrial robot ransomware: Akerbeltz. In: 2020 Fourth IEEE International Conference on Robotic Computing (IRC). pp. 432–435. http://dx.doi.org/10.1109/IRC.2020.00080.

Mayoral-Vilches, V., García-Maestro, N., Towers, M., Gil-Uriarte, E., 2020b. DevSecOps in robotics. arXiv:2003.10402 [cs]. URL: http://arxiv.org/abs/2003.10402.

Mayoral-Vilches, V., Juan, L.U.S., Dieber, B., Carbajo, U.A., Gil-Uriarte, E., 2019. Introducing the robot vulnerability database RVD. arXiv:1912.11299 [cs.CR]. URL: https://arxiv.org/abs/1912.11299.

Mayring, P., 2000. Qualitative content analysis. In: Forum: Qualitative Social Research. http://dx.doi.org/10.17169/fqs-1.2.1089.

Mayring, P., 2014. Qualitative Content Analysis: Theoretical Foundation, Basic Procedures and Software Solution. n.p., Klagenfurt, p. 143, URL: https://nbn-resolving.org/urn:nbn:de:0168-ssoar-395173.

Mayring, P., Fenzl, T., 2013. QCAmap. A software for qualitative content analysis. URL: https://www.qcamap.org/.

McCrae, R.R., John, O.P., 1992. An introduction to the five-factor model and its applications. J. Pers. 60 (2), 175–215. http://dx.doi.org/10.1111/j.1467-6494.1992.tb00970.x.

McEwan, D., Ruissen, G.R., Eys, M.A., Zumbo, B.D., Beauchamp, M.R., 2017. The effectiveness of teamwork training on teamwork behaviors and team performance: A systematic review and meta-analysis of controlled interventions. PLoS One 12 (1), e0169604. http://dx.doi.org/10.1371/journal.pone.0169604.

Meyer, B., 2017. Team diversity. In: Salas, E., Rico, R., Passmore, J. (Eds.), The Wiley Blackwell Handbook of the Psychology of Team Working and Collaborative Processes. Wiley, pp. 151–175. http://dx.doi.org/10.1002/9781118909997.ch7.

Migues, S., Steven, J., Ware, M., 2020. BSIMM11 report: Technical report. URL: https://www.bsimm.com/download.html.

Minbashian, A., Earl, J., Bright, J.E., 2013. Openness to experience as a predictor of job performance trajectories. Appl. Psychol. 62 (1), 1–12. http://dx.doi.org/10.1111/j.1464-0597.2012.00490.x.

Mitre Corporation, 1999. CVE – common vulnerabilities and exposures. URL: www.cve.org.

Mitre Corporation, 2021. CVE – common vulnerabilities and exposures. URL: www.cve.org/About/Metrics.

Mohan, V., Othmane, L.B., 2016. SecDevOps: Is it a marketing buzzword? - Mapping research on security in DevOps. In: 2016 11th International Conference on Availability, Reliability and Security (ARES). IEEE, pp. 542–547. http://dx.doi.org/10.1109/ARES.2016.92.

Mohanani, R., Ram, P., Lasisi, A., Ralph, P., Turhan, B., 2017. Perceptions of creativity in software engineering research and practice. In: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp. 210–217. http://dx.doi.org/10.1109/SEAA.2017.21.

Mohapeloa, T., 2017. Effects of silo mentality on corporate ITC's business model. pp. 1009–1019. http://dx.doi.org/10.1515/picbe-2017-0105.

Moyón, F., Méndez, D., Beckers, K., Klepper, S., 2020. How to integrate security compliance requirements with agile software engineering at scale? In: Morisio, M., Torchiano, M., Jedlitschka, A. (Eds.), Product-Focused Software Process Improvement. Springer International Publishing, Cham, pp. 69–87.

Mussel, P., 2013. Introducing the construct curiosity for predicting job performance. J. Organ. Behav. 34 (4), 453–472. http://dx.doi.org/10.1002/job.1809.

Myrbakken, H., Colomo-Palacios, R., 2017. DevSecOps: a multivocal literature review. In: International Conference on Software Process Improvement and Capability Determination. pp. 17–29.

Oliveira, D., Rosenthal, M., Morin, N., Yeh, K.-C., Cappos, J., Zhuang, Y., 2014. It's the psychology stupid: how heuristics explain software vulnerabilities and how priming can illuminate developer's blind spots. In: Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC '14. ACM Press, New Orleans, Louisiana, pp. 296–305. http://dx.doi.org/10.1145/2664243.2664254.

Otero, L.D., Centeno, G., Ruiz-Torres, A.J., Otero, C.E., 2009. A systematic approach for resource allocation in software projects. Comput. Ind. Eng. 56 (4), 1333–1339. http://dx.doi.org/10.1016/j.cie.2008.08.002.

Parker, S., 2000. From passive to proactive motivation: The importance of flexible role orientations and role breadth self–efficacy. Appl. Psychol. 49 (3), 447–469. http://dx.doi.org/10.1111/1464-0597.00025.

Pearsall, M., Ellis, A., 2006. The effects of critical team member assertiveness on team performance and satisfaction. J. Manage. 32, 575–594. http://dx.doi.org/10.1177/0149206306289099.

Purna Sudhakar, G., Farooq, A., Patnaik, S., 2011. Soft factors affecting the performance of software development teams. Team Perform. Manage. 17 (3/4), 187–205. http://dx.doi.org/10.1108/13527591111143718.

R Core Team, 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL: https://www.R-project.org/.

Rehman, M., Mahmood, A.K., Salleh, R., Amin, A., 2012. Mapping job requirements of software engineers to big five personality traits. In: 2012 International Conference on Computer & Information Science (ICCIS). IEEE, pp. 1115–1122. http://dx.doi.org/10.1109/ICCISci.2012.6297193.

Rulifson, G., Bielefeldt, A., 2015. Engineering students' varied and changing views of social responsibility. In: 2015 ASEE Annual Conference and Exposition Proceedings. ASEE Conferences, pp. 26.643.1–26.643.16. http://dx.doi.org/10.18260/p.23981.

Ryan, R.M., Deci, E.L., 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychol. 55 (1), 68–78. http://dx.doi.org/10.1037//0003-066X.55.1.68.

Rybowiak, V., Garst, H., Frese, M., Batinic, B., 1999. Error orientation questionnaire (EOQ): reliability, validity, and different language equivalence. J. Organ. Behav. 20 (4), 527–547. http://dx.doi.org/10.1002/(SICI)1099-1379(199907)20:4{%}3C527::AID-JOB886{%}3E3.0.CO;2-G.

Salas, E., Tannenbaum, S.I., Kraiger, K., Smith-Jentsch, K.A., 2012. The science of training and development in organizations: What matters in practice. Psychol. Sci. Publ. Interest 13 (2), 74–101. http://dx.doi.org/10.1177/1529100612436661.

Salleh, N., Mendes, E., Grundy, J., 2014. Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. Empir. Softw. Eng. 19 (3), 714–752. http://dx.doi.org/10.1007/s10664-012-9238-4.

Saunders, B., Sim, J., Kingstone, T., Baker, S., Waterfield, J., Bartlam, B., Burroughs, H., Jinks, C., 2018. Saturation in qualitative research: exploring its conceptualization and operationalization. Qual. Quant. 52 (4), 1893–1907. http://dx.doi.org/10.1007/s11135-017-0574-8.

Sedelmaier, Y., Landes, D., 2014. Software engineering body of skills (SWEBOS). In: 2014 IEEE Global Engineering Education Conference (EDUCON). IEEE, pp. 395–401. http://dx.doi.org/10.1109/EDUCON.2014.6826125.

Serna, M.E., Serna, A., 2015. Knowledge in engineering: A view from the logical reasoning. Int. J. Comput. Theory Eng. 7 (4), 325–331. http://dx.doi.org/10.7763/IJCTE.2015.V7.980.

Snyder, C.A., Manz, C.C., LaForge, R.W., 1983. Self-management: A key to entrepreneurial survival? Am. J. Small Bus. 8 (1), 20–26. http://dx.doi.org/10.1177/104225878300800107.

Snyder, L.G., Snyder, M.J., 2008. Teaching critical thinking and problem solving skills. J. Res. Bus. Educ. 50 (2), 90–99.

Song, M., Wang, P., Yang, P., 2018. Promotion of secure software development assimilation: stimulating individual motivation. Chin. Manage. Stud. 12 (1), 164–183. http://dx.doi.org/10.1108/CMS-01-2017-0005.

Staub, S., Kaynak, R., Gok, T., 2016. What affects sustainability and innovation — Hard or soft corporate identity? Technol. Forecast. Soc. Change 102, 72–79. http://dx.doi.org/10.1016/j.techfore.2015.06.033.

Steelman, L.A., Levy, P.E., Snell, A.F., 2004. The feedback environment scale: Construct definition, measurement, and validation. Educ. Psychol. Meas. 64 (1), 165–184. http://dx.doi.org/10.1177/0013164403258440.

Stevens, M.J., Campion, M.A., 1994. The knowledge, skill, and ability requirements for teamwork: Implications for human resource management. J. Manage. 20 (2), 503–530.

Tahaei, M., Vaniea, K., 2019. A survey on developer-centred security. In: IEEE European Symposium on Security and Privacy Workshops. pp. 129–138. http://dx.doi.org/10.1109/EuroSPW.2019.00021.

Tang, K.N., 2019. Leadership and Change Management. Springer Singapore, Singapore, http://dx.doi.org/10.1007/978-981-13-8902-3.

Tantawi, K.H., Sokolov, A., Tantawi, O., 2019. Advances in industrial robotics: From industry 3.0 automation to industry 4.0 collaboration. In: 2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-ICON). IEEE, pp. 1–4. http://dx.doi.org/10.1109/TIMES-iCON47539.2019.9024658.

Taurer, S., Breiling, B., Svrta, S., Dieber, B., 2019. Case study: Remote attack to disable MiR100 safety. In: Proceedings of the First Cybersecurity for Robotics 2019 Conference (CSfR2019). pp. 11–18.

Teece, D.J., Pisano, G., Shuen, A., 1997. Dynamic capabilities and strategic management. Strategic Manage. J. 18 (7), 509–533. http://dx.doi.org/10.1002/(SICI)1097-0266(199708)18:7<509::AID-SMJ882>3.0.CO;2-Z.

Tekleab, A.G., Quigley, N.R., Tesluk, P.E., 2009. A longitudinal study of team conflict, conflict management, cohesion, and team effectiveness. Group Organ. Manage. 34 (2), 170–205. http://dx.doi.org/10.1177/1059601108331218.

Tondel, I.A., Jaatun, M.G., Meland, P.H., 2008. Security requirements for the rest of us: A survey. IEEE Softw. 25 (1), 20–27. http://dx.doi.org/10.1109/MS.2008.19.

Trbusic, H., 2014. Engineering in the community: Critical consciousness and engineering education. Interdiscip. Descr. Complex Syst. 12 (2), 108–118. http://dx.doi.org/10.7906/indecs.12.2.1.

Tyler, T.R., 2005. Promoting employee policy adherence and rule following in work settings: The value of self-regulatory approaches. Brooklyn Law Rev. 70, 1287–1312.

Tyler, T.R., Callahan, P.E., Frost, J., 2007. Armed, and dangerous (?): Motivating rule adherence among agents of social control. Law Soc. Rev. 41 (2), 457–492. http://dx.doi.org/10.1111/j.1540-5893.2007.00304.x.

Valori, M., Scibilia, A., Fassi, I., Saenz, J., Behrens, R., Herbster, S., Bidard, C., Lucet, E., Magisson, A., Schaake, L., Bessler, J., Prange-Lasonder, G.B., Kühnrich, M., Lassen, A.B., Nielsen, K., 2021. Validating safety in human–robot collaboration: Standards and new perspectives. Robotics 10 (2), 65. http://dx.doi.org/10.3390/robotics10020065.

van Dyck, C., Frese, M., Baer, M., Sonnentag, S., 2005. Organizational error management culture and its impact on performance: a two-study replication. J. Appl. Psychol. 90 (6), 1228–1240. http://dx.doi.org/10.1037/0021-9010.90.6.1228.

Venson, E., Guo, X., Yan, Z., Boehm, B., 2019. Costing secure software development. In: Proceedings of the 14th International Conference on Availability, Reliability and Security. ACM, New York, NY, USA, pp. 1–11. http://dx.doi.org/10.1145/3339252.3339263.

Vieira, E.R., Alves, C., Duboc, L., 2012. Creativity patterns guide: Support for the application of creativity techniques in requirements engineering. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Winckler, M., Forbrig, P., Bernhaupt, R. (Eds.), Human-Centered Software Engineering. In: Lecture Notes in Computer Science, vol. 7623, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 283–290. http://dx.doi.org/10.1007/978-3-642-34347-6_19.

Virmani, M., 2015. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In: Fifth International Conference on the Innovative Computing Technology (INTECH 2015). IEEE, pp. 78–82. http://dx.doi.org/10.1109/INTECH.2015.7173368.

Vishnubhotla, S.D., Mendes, E., Lundberg, L., 2021. Understanding the perceived relevance of capability measures: A survey of agile software development practitioners. J. Syst. Softw. 180, 111013. http://dx.doi.org/10.1016/j.jss.2021.111013.

Webber, S.S., Donahue, L.M., 2001. Impact of highly and less job-related diversity on work group cohesion and performance: a meta-analysis. J. Manage. 27 (2), 141–162. http://dx.doi.org/10.1177/014920630102700202.

Weir, C., Migues, S., Ware, M., Williams, L., 2021. Infiltrating security into development: exploring the world's largest software security study. In: Spinellis, D., Gousios, G., Chechik, M., Di Penta, M. (Eds.), Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, New York, NY, USA, pp. 1326–1336. http://dx.doi.org/10.1145/3468264.3473926.

Welch, M., Jackson, P.R., 2007. Rethinking internal communication: a stakeholder approach. Corp. Commun. Int. J. 12 (2), 177–198. http://dx.doi.org/10.1108/13563280710744847.

Wilmot, M.P., Ones, D.S., 2019. A century of research on conscientiousness at work. Proc. Natl. Acad. Sci. USA 116 (46), 23004–23010. http://dx.doi.org/10.1073/pnas.1908430116.

Witt, L.A., Burke, L.A., Barrick, M.R., Mount, M.K., 2002. The interactive effects of conscientiousness and agreeableness on job performance. J. Appl. Psychol. 87 (1), 164–169. http://dx.doi.org/10.1037/0021-9010.87.1.164.

Wohlgemuth, V., Wenzel, M., Berger, E.S., Eisend, M., 2019. Dynamic capabilities and employee participation: The role of trust and informal control. Eur. Manage. J. 37 (6), 760–771. http://dx.doi.org/10.1016/j.emj.2019.02.005.

Wood, S., Michaelides, G., Thomson, C., 2013. Successful extreme programming: Fidelity to the methodology or good teamworking? Inf. Softw. Technol. 55 (4), 660–672. http://dx.doi.org/10.1016/j.infsof.2012.10.002.

Yaacoub, J.-P.A., Noura, H.N., Salman, O., Chehab, A., 2021. Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. Int. J. Inf. Secur. 1–44, Publisher: Springer.

Yilmaz, M., O'Connor, R.V., Colomo-Palacios, R., Clarke, P., 2017. An examination of personality traits and how they impact on software development teams. Inf. Softw. Technol. 86, 101–122. http://dx.doi.org/10.1016/j.infsof.2017.01.005.

Yusr, M.M., Mokhtar, S.S.M., Othman, A.R., Sulaiman, Y., 2017. Does interaction between TQM practices and knowledge management processes enhance the innovation performance? Int. J. Qual. Reliab. Manage. 34 (7), 955–974. http://dx.doi.org/10.1108/IJQRM-09-2014-0138.

Zell, E., Lesick, T.L., 2021. Big five personality traits and performance: A quantitative synthesis of 50+ meta-analyses. J. Pers. http://dx.doi.org/10.1111/jopy.12683.

Glasauer Christina, BSc. MSc. is a Psychologist, currently researching as a Senior Scientist at the University of Klagenfurt, Austria. She is part of the Karl-Popper-College SEEROSE (Responsible, Safe, and Secure Robotic Systems Engineering). Her primary research focus is on scale development and psychometric assessment of psychological factors enhancing safety and security of robotic systems by design.