Poster: Towards A Reliable Privacy-Enhanced V-Model For Software Development

Ala'a Al-Momani*, Frank Kargl*, Robert Schmidt†, Antonio Kung‡, Christoph Bösch*
*Institute of Distributed Systems, Ulm University, Ulm, Germany
{alaa.al-momani, frank.kargl, christoph.boesch}@uni-ulm.de
†Denso Automotive Deutschland GmbH, Munich, Germany
r.schmidt@denso-auto.de
‡Trialog, Paris, France
antonio.kung@trialog.com

***Abstract*—This poster is associated with the paper entitled: *a privacy-aware V-model for software development*[1]. In this poster, we propose solutions to the issue of incorporating privacy by design in the commonly used V-model for system development. In particular, we propose the *W-model* as an extension of the V-model, and further build on the W-model by proposing the novel σ-*model* which solves some limitations of the W-model.**

I. Introduction

Privacy-by-Design (PbD) has gained tremendous attention in the last decade, especially after the adoption of new data protection regulations such like *GDPR*. In order to bring this relatively new notion into an easier deployment, a systematic methodology has to be present. PRIPARE [1] proposed a privacy engineering methodology to address PbD and prepare the industry for it. To ease the integration with the current system development models, PRIPARE disucssed how to integrate PbD into the widespread Agile and Waterfall models.

One of the widely used system development models is the *V-model*. The V-model is a relatively old model but still preferred to be used in scenarios where explicit development phases with a clear documentation of each phase is required for, e.g., safety purposes. The steps of system development according to the V-model state that the system is designed based on pre-defined business and system requirements. Then, the system is implemented and, thereafter, verified at several levels. Moreover, a final acceptance testing takes place before releasing the system.

The V-model is a generic model which does not specifically address privacy nor privacy-related activities within its development phases. The V-model has been — so far — disregarded when integrating privacy processes into system development models. In order to avoid the potential ambiguity in organizations that follow the V-model when integrating PbD activities into their already-established processes, we propose extensions to the V-model that integrates PbD into the V-model. In particular, we introduce the *W-model* as a privacy-aware extension of the classical V-model. We, further, point out some potential limitations of the proposed W-model. Therefore, we build on the W-model to remedy those potential limitations and propose the σ-*model* as an enhanced version of the W-model.

II. The W-Model

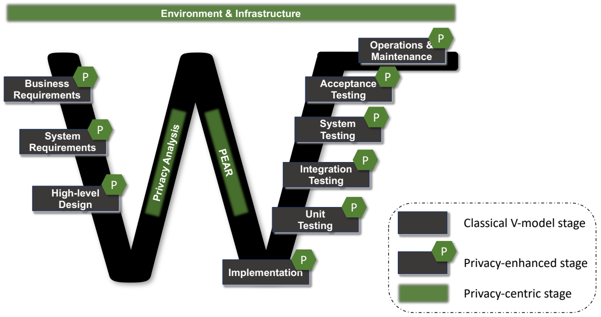

Fig. 1: The W-Model

One of the main differences between the W- and the V-model is that the W-model adds two stages that specifically address privacy; i.e., the *privacy analysis* and privacy-enhanced architecture (*PEAR*) stages as shown in Figure 1. Those additional stages are required to deeply analyze privacy in the system after its initial design. Furthermore, the W-model fundamentally changes the scope and the activities of each stage of the V-model while highlighting the requirement of promoting privacy awareness among the organization through the stage of *environment & infrastructure*. As seen in Figure 1, the W-model suggests to consider individuals' privacy at all stages of development. At the *business requirements* stage, the W-model suggests to change the (obvious) privacy-violating requirements to more privacy-considered requirements. In addition, business requirements might potentially include explicit privacy requirements, e.g., to comply with a specific regulation. In the *system requirements* stage, system engineers keep individuals' privacy in mind while mirroring business requirements into system ones. If a potential privacy violation is found during such a process, then system engineers have to either: 1.) come up with a system requirement that yields the desired functionality but requires less-sensitive data items, or 2.) document their findings to the subsequent teams to be further analyzed. Thereafter, system engineers create a *high-*

[1] A. Al-Momani, F. Kargl, R. Schmidt, A. Kung, C. Bösch, "A Privacy-Aware V-Model for Software Development," in *2019 IEEE International Workshop on Privacy Engineering — IWPE'19.*

level design of the system which is considered a *somewhat privacy-preserving design*.

Privacy analysis is the first introduced privacy-centric stage in the W-model. Privacy engineers perform a PIA to elicit privacy threats in the initial design, and to find suitable countermeasures accordingly. Some of the elicited threats might be unsolvable with the current design of the system, thus, some of the business or system requirements have to be changed to remedy the privacy threats. The second privacy-centric stage is ***PEAR*** which is divided into two substages; privacy-preserving high- and low-level design of the system. This stage yields an architecture that is privacy-enhanced and meets all of business, system, and privacy requirements. Thus, it is at a level that allows system developers to start implementing it in the stage of *implementation*. Thereafter, system verifiers test the implemented units as well as the interaction among them in the stages of *unit testing* and *integration testing* while ensuring that neither the units nor the interactions pose privacy threats to individuals. Then, the verifiers make sure that the implemented system meets the system requirements while ensuring no privacy risks in the stage of *system testing*. Before releasing the system, the *acceptance testing* takes place to ensure that the business requirements are met and, thus, the system qualifies to be released. Upon releasing the system, the developing organization establishes comprehensive practices as a response in case of incidents such as, e.g., data breaches. After releasing the system, periodic legal and technical assessments take place to capture emerging requirements including privacy ones, e.g., in case of new regulations.

The proposed W-model extends the V-model to include all PbD phases that PRIPARE suggested. We foresee the W-model to be used in organizations that follow the V-model for software development and that wish to design and introduce privacy-preserving systems. Despite that the W-model addresses PbD phases, it does not properly reflect the recurrent and repetitive nature of PbD phases before releasing the system. That is, several rounds of the pre-release PbD phases are required to satisfactorily address PbD. This limitation of the W-model is because of the static and strict nature inherited from the classical V-model. To address this feature of PbD properly in the system development lifecycle, we build on the W-model and propose the σ-*model* in the following section.

III. The σ-Model

We present the σ-*model* in Figure 2. Akin to the W-model, the σ-model addresses all of the PbD phases and pours them into development lifecycle stages. However, the key difference between the two models is that the σ-model properly addresses the recurrent nature of the PbD phases especially before releasing the system. The σ-model consists of two parts; a loop of 10 stages and an appendage. The loop includes the stages prior to releasing the system while the appendage represents the post-release stage. According to the σ-model, system development starts with the stage of *business requirements* and cycles clockwise. We foresee several loop cycles before releasing the system. Within these cycles, some stages might be skipped in order to, intuitively, address the counterclockwise direction, i.e., when system development status needs to cycle backward to a previous stage for, e.g., modification.

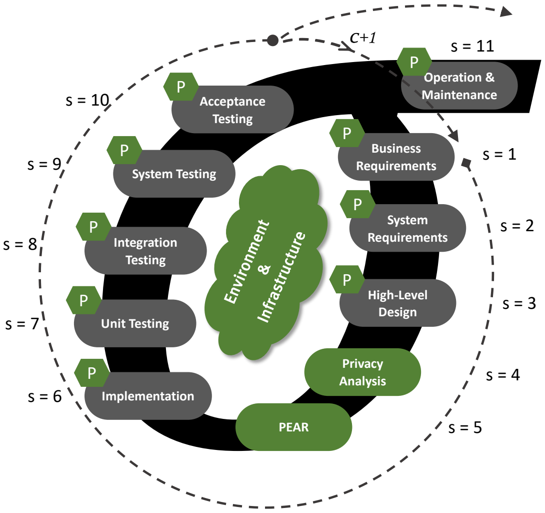

Fig. 2: The σ-Model

However, in order to release the system, i.e., move the development status to the appendage, all of the stages in the loop have to be addressed sequentially in one *clockwise* cycle. Going through the stages of the loop without skipping any stage conveys that the system has been developed in a way that meets functional as well as privacy requirements, and, therefore, is ready to be released. Formally, we denote each processed stage as $X_{c,s}$, where c is the cycle number, and s is the stage number. For example, $X_{2,4}$ refers to the *privacy analysis* stage in the *second cycle*. Thus, in order to release the system, the following condition has to be met:

$$\exists X_{c,s} \exists c \, \forall \, s \in \{1, 2, ..., 10\}$$

IV. Conclusion & Future Work

In this poster, we raised the question of how to address PbD processes and include them in the V-model that is frequently used for software development. We proposed the W-model as a privacy-aware extension of the classical V-model. The W-model addresses privacy and adds two privacy-centric stages to the V-model. Furthermore, we proposed the σ-model that addresses the recurrent nature of PbD through the development lifecycle in a better way than the W-model does.

Future work should focus on evaluating the applicability of such models on a practice level. In other words, examining their ability to assist system developers, who follow the V-model, in introducing privacy-enhanced systems.

References

[1] N. Notario, A. Crespo, Y.-S. Martin, J. M. Del Alamo, D. Le Métayer, T. Antignac, A. Kung, I. Kroener, and D. Wright, "PRIPARE: integrating privacy best practices into a privacy engineering methodology," in *Security and Privacy Workshops (SPW), 2015 IEEE*. pp. 151–158.