PRIVACY UND SECURITY BY DESIGN IM AGILEN SOFTWAREENTWICKLUNGSPROZESS

Oliver Terbu / Walter Hötzendorfer / Maria Leitner / Arndt Bonitz / Stefan Vogl / Sebastian Zehetbauer

Projektmitarbeiter, Österreichische Staatsdruckerei GmbH, e-government innovations
Tenschertstraße 7, 1239 Wien, AT
{terbu; vogl; zehetbauer}@staatsdruckerei.at; https://www.staatsdruckerei.at

Projektassistent, Universität Wien, Arbeitsgruppe Rechtsinformatik
Schottenbastei 10-16/2/5, 1010 Wien, AT
walter.hoetzendorfer@univie.ac.at; http://rechtsinformatik.univie.ac.at

Scientist; Engineer, AIT Austrian Institute of Technology GmbH, Digital Safety Security, Information Management
Donau-City-Straße 1, 1220 Wien, AT
{maria.leitner; arndt.bonitz}@ait.ac.at; http://www.ait.ac.at

Schlagworte: ***Privacy by Design, Security by Design, Softwareentwicklung, Scrum, Datenschutz, E-Democracy, E-Partizipation***

Abstract: *Der Beitrag präsentiert ein Prozessmodell zur agilen Softwareentwicklung, welches die Umsetzung der Anforderungen des Privatsphäre- und Datenschutzes sowie der Informationssicherheit zu einem zentralen Systembestandteil macht. Dieser Softwareentwicklungsprozess involviert sowohl Informationssicherheits- als auch Datenschutz-Experten, welche die Umsetzung der genannten Anforderungen verantworten. Das Prozessmodell wird im Rahmen des KIRAS-Projektes «ePartizipation – Authentifizierung bei demokratischer Online-Beteiligung» eingesetzt; erste daraus gewonnene Erfahrungen werden diskutiert.*

1. Einleitung

Privacy by Design (PbD) gewinnt zunehmend an Bedeutung und fordert im Wesentlichen die Berücksichtigung von Privatsphäreschutz und Datenschutz bereits während der Entwicklung eines Systems und nicht erst dann, wenn dieses schließlich eingesetzt wird. Dasselbe gilt für die Berücksichtigung der Informationssicherheit (Security) des Systems. Dies klingt ziemlich einleuchtend und einfach; es stellt sich allerdings die Frage, was ein solches Vorgehen in der Praxis bedeutet. Mit dieser Frage beschäftigt sich der vorliegende Beitrag.

Es wird ein auf Scrum basierender Softwareentwicklungsprozess vorgestellt, der systematische Mechanismen zur Umsetzung von Security- und Privacy-Anforderungen beinhaltet. Scrum ist ein leichtgewichtiges Framework zur Lösung komplexer Probleme, um produktiv und kreativ ein bestmögliches Produkt zu liefern. Das Framework beruht auf der empirischen Prozesskontrolle, d.h. produzierte Artefakte werden nicht als gegeben angesehen, sondern stets einem iterativen Verbesserungsprozess unterzogen.[1] Weitere Grundlagen zu Scrum können in der an der jeweiligen Stelle angeführten Literatur und in zahlreichen weiteren Quellen nachgelesen werden und werden als Basis der nachfolgenden Ausführungen vorausgesetzt.

In der Konzeption des Prozesses wurde auf verschiedenen Ansätzen aufgebaut, die den Gebieten Security by

[1] Vgl. Schwaber/Sutherland, The Scrum Guide, http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf (aufgerufen am 7. Januar 2016), 2013.

Design bzw. Secure Coding zuzuordnen sind und jeweils referenziert sind. Die daraus abgeleiteten Konzepte wurden darüber hinaus um die Berücksichtigung von Privacy by Design erweitert. Der so konzipierte Prozess dient im Projekt «ePartizipation – Authentifizierung bei demokratischer Online-Beteiligung» als Grundlage für die Entwicklung des Demonstrators. Erste Erfahrungen mit dem Prozess werden in Kapitel 4 diskutiert. Im vorliegenden Beitrag soll überwiegend auf die Privacy-Aspekte des Prozesses eingegangen werden.

2. Privacy und Security by Design

PbD «bedeutet, dass der Schutz der Privatsphäre und der Datenschutz in den gesamten Technologie-Lebenszyklus integriert werden, vom frühen Entwurfsstadium bis zu deren Einführung, Nutzung und letztendlichen Außerbetriebnahme.»[2] Wie in dieser knappen und treffenden Definition der Europäischen Kommission deutlich wird, ist es das Kernelement des Prinzips PbD, den Schutz der Privatsphäre und den Datenschutz nicht auf ein bereits entwickeltes System aufzusetzen, sondern bereits im Entwicklungsprozess des Systems in das System «einzubauen». In der künftigen Datenschutzgrundverordnung (DSGVO) wird «data protection by design» zu einem Grundprinzip des Datenschutzes erhoben, sodass «both at the time of the determination of the means for processing and at the time of the processing itself» angemessene technische und organisatorische Maßnahmen getroffen werden müssen, um die Anforderungen der Verordnung zu erfüllen und die Rechte der Betroffenen zu wahren.[3] PbD kann mit «eingebauter Datenschutz»[4] oder «Datenschutz durch Technik»[5] übersetzt werden, hier wird aber dem englischen Terminus der Vorzug gegeben.

Das Konzept PbD geht bereits auf die 1990er-Jahre zurück und ist eng mit der Person Ann Cavoukian verbunden, der ehemaligen Datenschutzbeauftragten der kanadischen Provinz Ontario, die sieben Grundprinzipien von PbD postulierte.[6] Bei diesen handelt es sich allerdings nur um abstrakte Vorgaben, die nichts über ihre Umsetzung aussagen und denen Mechanismen zur Integration von Datenschutz in den Entwicklungsprozess eines Systems fehlen.[7]

Zahlreiche Publikationen haben sich in den letzten Jahren mit der datenschutzgerechten Gestaltung von Systemen beschäftigt, manche auch, ohne dies als PbD zu bezeichnen.[8] All diese Publikationen machen zweierlei deutlich: Erstens ist es die Einhaltung einiger weniger Grundprinzipien, die PbD ausmacht, allen voran das Prinzip der Datenminimierung. Die folgende Liste von Privacy-Strategien stammt von Hoepman[9] und findet

[2] Europäische Kommission, Eine Digitale Agenda für Europa, KOM(2010)245 endg., 19. Mai 2010, S. 20.

[3] Art. 23 des finalen Kompromisstexts der DSGVO, http://www.statewatch.org/news/2015/dec/eu-council-dp-reg-draft-final-compromise-15039-15.pdf (abgerufen am 10. Januar 2016).

[4] So Art. 12 Abs. 3 lit c eIDAS-VO (Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates vom 23. Juli 2014 über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG, ABl. L 247, 73 vom 28. August 2014), Vergleich der deutschen mit der englischen Fassung.

[5] So der deutsche Titel von Art 23 des Entwurfs der DSGVO (Vorschlag für VERORDNUNG DES EUROPÄISCHEN PARLAMENTS UND DES RATES zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr (Datenschutz-Grundverordnung), KOM (2012) 11 endg., 25. Januar 2012).

[6] Cavoukian, Privacy by Design: The 7 Foundational Principles, http://www.privacybydesign.ca/content/uploads /2009/08/7foundationalprinciples.pdf (aufgerufen am 3. Juni 2015), 2009, Hervorhebungen im Original.

[7] Vgl. ENISA, Privacy and Data Protection by Design – from policy to engineering, https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/privacy-and-data-protection-by-design /at_download/fullReport (abgerufen am 6. August 2015), 2014, S. 2.

[8] Z.B. die soeben zitierte sowie Gürses/Troncoso/Diaz, Engineering Privacy by Design, Computers, Privacy & Data Protection (CPDP 2011), 2011; van Rest/Boonstra/Everts/van Rijn/van Paassen, Designing Privacy-by-Design. In: *Preneel/Ikonomou* (Hrsg.), Proceedings of the First Annual Privacy Forum, APF 2012, Lecture Notes in Computer Science, Springer, Berlin und Heidelberg 2014, S. 55–72; Deng/Wuyts/Scandariato/Preneel/Joosen, A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements, Requirements Engineering 2011, S. 3–32; Kung, PEARs: Privacy Enhancing ARchitectures, Proceedings of the Second Annual Privacy Forum APF 2014, Lecture Notes in Computer Science, Springer, Berlin und Heidelberg 2014, S.18–29; Spiekermann/Cranor, Engineering Privacy, IEEE Transactions on Software Engineering 2009, S. 67–82 u.a.

[9] Hoepman, Privacy Design Strategies, ICT Systems Security and Privacy Protection, IFIP Advances in Information and Communica-

sich auch im bereits zitierten Papier der ENISA[10]:

- *MINIMISE: The amount of personal data that is processed should be restricted to the minimal amount possible.*
- *HIDE: Any personal data, and their interrelationships, should be hidden from plain view.*
- *SEPARATE: Personal data should be processed in a distributed fashion, in separate compartments whenever possible.*
- *AGGREGATE: Personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.*
- *INFORM: Data subjects should be adequately informed whenever personal data is processed.*
- *CONTROL: Data subjects should be provided agency over the processing of their personal data.*
- *ENFORCE: A privacy policy compatible with legal requirements should be in place and should be enforced.*
- *DEMONSTRATE: Be able to demonstrate compliance with the privacy policy and any applicable legal requirements.*

Diese Strategien können in zwei Gruppen eingeteilt werden: Die vier erstgenannten sind datenorientiert, die anderen prozessorientiert.

Zweitens kann wohl kein einfacher, allgemein anwendbarer Standardprozess definiert werden, um diese Grundsätze im Entwicklungsprozess eines Systems zuverlässig umzusetzen. Dies muss jeweils individuell erfolgen, indem unter Berücksichtigung des Zwecks und der gewünschten Funktionalität des jeweiligen Systems geeignete Strategien, Design Patterns und Technologien (Privacy-enhancing Technologies, PETs) im System implementiert werden. Dies kann als «Privacy Engineering» bezeichnet werden und erfordert u.E. Experten – «Privacy Engineers» –, die technische Kenntnisse und Kenntnisse des Datenschutzes haben, einschließlich «weicher» Faktoren, wie das Wissen über Good Practice, Design Patterns, häufige Fehler, aktuelle Bedrohungen und Angriffsmethoden etc. Dies schlägt eine Brücke zu den Security-Experten, eine Profession, die bereits wesentlich länger etabliert ist und sich auch bereits länger mit der praktischen Umsetzung von Security by Design befasst, für das die in diesem Kapitel dargestellte Zielsetzung analog zu sehen ist.[11] Es zeigt sich, dass Security by Design und Privacy by Design einander überschneiden und nicht strikt getrennt betrachtet werden können.

Der nachfolgend präsentierte Entwicklungsprozess ist ein Prozess, in dem inkrementell ein Softwareprodukt entwickelt wird. Die Darstellung konzentriert sich dabei auf den Prozess und die Prozessbeteiligten. Die Behandlung der Frage, wie und woraus Security-Anforderungen abgeleitet werden und wie aus den genannten Privacy-Strategien Privacy-Anforderungen abgeleitet werden, würde den Rahmen sprengen. Zu dieser Frage sei auf andere Publikationen verwiesen.[12] Zu betonen ist auch, dass Privacy und Security by Design nur dann

tion Technology 428, 2014, S. 446.

10 ENISA, Privacy and Data Protection by Design – from policy to engineering, S. 18 ff.

11 Vgl. Waidner/Backes/Müller-Quade, Entwicklung sicherer Software durch Security by Design, Fraunhofer SIT, https://www.kastel.kit.edu/downloads/Entwicklung_sicherer_Software_durch_Security_by_Design.pdf (aufgerufen am 10. Januar 2016), 2013.

12 Z.B. PRIPARE, Deliverable D1.2 Privacy and Security-by-design Methodology, v1.04, Draft, 2014 http://pripareproject.eu/wp-content/uploads/2013/11/PRIPARE_Deliverable_D1.2_draft.pdf (aufgerufen am 10. Januar 2016), liegt noch nicht in einer finalen Version vor; Hörbe/Hötzendorfer, Privacy by Design in Federated Identity Management, Proceedings of the 2015 IEEE Security

effektiv umgesetzt werden kann, wenn es bereits beim Entwurf der Softwarearchitektur berücksichtigt wird.

3. Konzept eines iterativen Privacy und Security by Design-Entwicklungsprozesses

Wie gezeigt wurde, existiert PbD als Konzept bereits sehr lange, hat aber in die Praxis der Softwareentwicklung noch wenig Eingang gefunden. Dies wird, wie erwähnt, in Form eines Standardprozesses, dessen Einhaltung automatisch ein datenschutzfreundliches Endprodukt garantiert, auch gar nicht möglich sein. Vielmehr muss PbD unter Berücksichtigung der Spezifika der jeweiligen Software individuell umgesetzt werden. Ein Softwareentwicklungsprozess, der dies sowohl in Bezug auf Privacy als auch in Bezug auf Security gewährleisten soll, wird im Folgenden vorgeschlagen. Dabei handelt es sich um einen iterativen, agilen Entwicklungsprozess, der auf dem Scrum-Framework basiert.

Scrum sieht keine standardisierte Methode vor, um Privacy- und Security-Anforderungen umzusetzen. Abbildung 1 zeigt unseren Prozess, der mittels Scrum definiert wurde, und in welchen Phasen welche Maßnahmen gesetzt werden, um diese Anforderungen dennoch zu berücksichtigen. Die Entwicklung erfolgt in einer Anzahl (je nach Bedarf) an Iterationen – auch als *Sprints* bezeichnet – mit der Dauer von 2-3 Wochen. Das Ergebnis eines jeden *Sprints* ist jeweils ein um Funktionalität erweitertes, lauffähiges und getestetes Produkt (oder Inkrement). Innerhalb eines *Sprints* werden die Scrum-typischen Phasen durchlaufen. Zusätzlich gibt es ein *Backlog Refinement* Meeting (oder auch als *Backlog Grooming* bezeichnet)[13] vor dem ersten *Sprint Planning* und später jeweils optional während der *Sprints*. Zusätzlich gibt es dedizierte Privacy- und Security-Sprints, die sich besonders intensiv mit der Umsetzung von Privacy und Security beschäftigen. Diese können auch unter Zuhilfenahme spezieller Privacy- und Security-Teams erfolgen. Ein zentrales Security-Team sieht auch Kumar[14] vor.

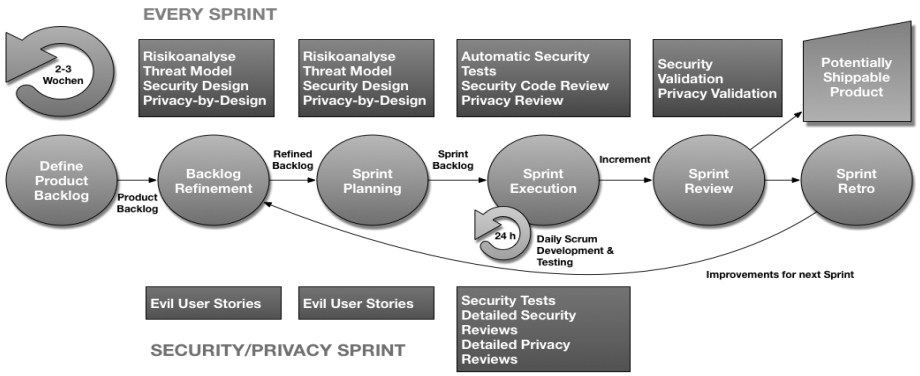

Abbildung 1: Iterativer Privacy und Security by Design-Entwicklungsprozesses mit Scrum

and Privacy Workshops, 2015, S. 167–174.

13 Vgl. Cohn, Product Backlog Refinement (Grooming), https://www.mountaingoatsoftware.com/blog/product-backlog-refinement-grooming (aufgerufen am 8. Januar 2016), 2015.

14 Kumar, How to address Application Security in agile Scrum Teams?, http://vitalflux.com/how-to-address-application-security-in-agile-scrum-teams/ (aufgerufen am 8. Januar 2016), 2013.

3.1. Zielsetzung

Mit dem in diesem Kapitel beschriebenen Prozess soll die Qualität der Software hinsichtlich Privacy und Security erhöht werden. Dies umfasst die Reduktion der Wahrscheinlichkeit einer Kompromittierung des Systems. Außerdem soll die ständige Begleitung durch Datenschutzexperten die Akzeptanz der Plattform erhöhen. Privacy- und Security-Anforderungen werden einfacher vergleich- und priorisierbar und dadurch immanent adressiert. Nicht zuletzt sollen einfach zu verstehende Maßnahmen Entwickler hinsichtlich Privacy und Security sensibilisieren und somit das Verständnis für diese Schwerpunkte nachhaltig erhöhen.

3.2. Maßnahmen zur Forcierung und Überprüfung von Privacy und Security

3.2.1. Abbilden von Privacy- und Security-Anforderungen im Product Backlog

Sind Privacy- und Security-Anforderung funktional, so werden sie als *Product Backlog Items* (PBI) dem *Product Backlog* hinzugefügt. PBI werden von den *Product Ownern* (PO) gepflegt und während der *Backlog Refinement*- und *Planning* Meetings mit dem Team überarbeitet. PBI werden zum besseren Verständnis in natürlicher Sprache als *User Stories*[15] mittels Satzschablone formuliert und lassen erkennen, welche Rolle, welches Ziel und welchen Nutzen der Benutzer am System hat, z.B. *Als Bürger möchte ich mich mit Bürgerkarte anmelden, um zu verhindern, dass jemand anders an meiner Stelle abstimmt*. Zu jeder *User Story* werden Akzeptanzkriterien definiert. Sofern möglich, werden diese gleich als *Story Tests*[16] mittels Satzschablone strukturiert erfasst – z.B. *Verifiziere, dass der Nutzer sich auf dem Portal mittels Bürgerkarte anmelden kann*.

Privacy- und Security Anforderungen fallen oftmals in die Kategorie nicht-funktionale Anforderungen. In Fachkreisen werden drei Methoden diskutiert, wie diesen begegnet wird. Schließlich soll eine Kombination dieser Methoden angestrebt werden:

Story Tests sind der primäre Weg, um nicht-funktionale Anforderungen abzubilden und eignen sich vor allem, wenn diese nicht wiederkehrend sind – z.B. *Verifiziere, dass der Benutzer nach fünf aufeinanderfolgenden fehlgeschlagenen Anmeldeversuchen gesperrt ist*.

User Stories sind in erster Linie dazu gedacht, Anforderungen mit Geschäftswert abzubilden, können jedoch auch eingesetzt werden, um nicht-funktionale Anforderungen zu modellieren. Betrifft eine Anforderung nicht alle PBI, oder ist sie zu groß als dass sie als *Story Test* erfasst werden kann, wird eine *technische User Story*[17] dafür geschrieben – z.B. *Als IT-Security-Mitarbeiter möchte ich die Anzahl an nicht erfolgreichen Authentifizierungsversuchen beschränken, um Brute-Force-Attacken zu verhindern*. Ein Vorteil dieser Methode ist, dass Security und Privacy für die PO explizit sichtbar, priorisierbar und vergleichbar gemacht wird. SAFECode[18] beschreibt einen Katalog an generischen *technischen User Stories*, die helfen die Sicherheit des Systems zu erhöhen.

Die *Definition of Done* (DoD)[19] ist eine Checkliste, die festlegt, welche Eigenschaften eine *User Story* er-

[15] Vgl. Cohn, User Stories Applied: For Agile Software Development, Addison-Wesley Professional, Boston und andere 2004.

[16] Vgl. Bradley, Story Testing Patterns, http://www.scrumcrazy.com/Story+Testing+Patterns (aufgerufen am 6. September 2015), 2013.

[17] Vgl. Ambler, Beyond Functional Requirements On Agile Projects, http://www.drdobbs.com/architecture-and-design/beyond-functional-requirements-on-agile/210601918 (aufgerufen am 6. September 2015), 2008.

[18] SAFECode, Practical Security Stories and Security Tasks for Agile Development Environments, http://www.safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf (aufgerufen am 6. September 2015), 2012.

[19] Vgl. Agile Alliance, Definition of Done, http://guide.agilealliance.org/guide/definition-of-done.html (aufgerufen am 6. September 2015), 2013.

füllen muss, um sie als abgeschlossen anzusehen. Sie wird vom Entwicklungsteam gemeinsam mit den PO erarbeitet und ist einem stetigen Änderungsprozess unterworfen. Hat die nicht-funktionale Anforderung globale Relevanz, kann sie in die DoD mitaufgenommen werden – z.B. *Persönliche Daten eines Nutzers dürfen von anderen Benutzern nicht eingesehen werden*. So wird gewährleistet, dass die Anforderung später nicht gebrochen wird.

3.2.2. User Stories aus Sicht eines Angreifers

OWASP beschreibt vier generelle *User Stories,* sogenannte *Evil User Stories*. Nutznießer ist nicht der Benutzer, sondern ein potentieller Angreifer. *Evil User Stories* richten den Fokus auf die Kompromittierung des Systems.[20] Analog dazu antizipieren *Misuse* oder *Abuse Stories* unerwünschtes Verhalten, das die Systemfunktionalität zu missbrauchen sucht. Der Grundgedanke ist, dass das Schreiben sicherer Software auch das Wissen um den Missbrauch der Systemfunktionalität voraussetzt.[21] Beide Varianten werden in Zusammenarbeit mit Privacy- und Security-Experten definiert und unter dem Begriff *Evil User Stories* zusammengefasst. Sie bieten eine weitere Möglichkeit, Privacy und Security im *Product Backlog* explizit sichtbar zu machen.

3.2.3. Statische Code-Analyse

Bei jedem Check-In prüft der Continuous-Integration-Server automatisch mittels statischer Code-Analyse-Tools Änderungen des Quellcodes auf Einhaltung der (Secure) Coding Guidelines und gibt den Entwicklern Rückmeldung über etwaige Verstöße. Dies kann beispielsweise auch beinhalten, dass vor dem Zugriff auf bestimmte Daten (z.B. personenbezogen Daten) diese stets entschlüsselt und nach Verwendung wieder verschlüsselt werden.

3.2.4. Automatische Privacy- und Security-Tests

So weit als möglich werden auch Privacy- und Security-Tests durch den Einsatz entsprechender Scanner und Fuzzer automatisiert – z.B. Prüfen auf Anfälligkeit gegenüber Cross Site Scripting (XSS).

3.2.5. Inkrementelle Privacy- und Security-Reviews

Vor Abschluss einer *User Story* werden alle damit verbundenen Änderungen am Quellcode einem kurzen Review unterzogen. Ziel ist es, den Code auf Verstöße gegen Security Best Practices zu prüfen (Buffer Overflows etc.). Andererseits liegt das Augenmerk auch auf dem Schutz der Privatsphäre und personenbezogener Daten. Die Prüfung muss sich nicht zwingend auf den Quellcode, sondern kann sich auch auf beliebige Artefakte beziehen. Ziel ist unter anderem, die Einhaltung der oben aufgezählten Privacy-Strategien zu gewährleisten. Reviews können mittels Review-Tools semi-automatisch erfolgen. Am Ende eines Sprints müssen alle bestehenden Änderungen am Code geprüft worden sein.

3.3. Zusätzliche Rollen

Scrum sieht vor, dass die Fähigkeiten innerhalb des Entwicklungsteams konvergieren. In der Praxis ist dies gerade auf den Gebieten Privacy und Security nicht möglich. Analyse, Planung und Umsetzung entsprechender Maßnahmen erfordern i.d.R. Spezialwissen, weshalb je ein Security- und ein Privacy-Team außerhalb des Entwicklungsteams dafür vorgesehen sind:

[20] Vgl. OWASP, Agile Software Development: Don't forget EVIL User Stories, https://www.owasp.org/index.php/Agile_Software_Development:_Don%27t_Forget_EVIL_User_Stories (aufgerufen am 6. September 2015), 2011.

[21] Vgl. McGraw, Misuse and Abuse Cases: Getting Past the Positive, IEEE Security & Privacy, Volume 2, Issue 3, 2004, S. 90–92.

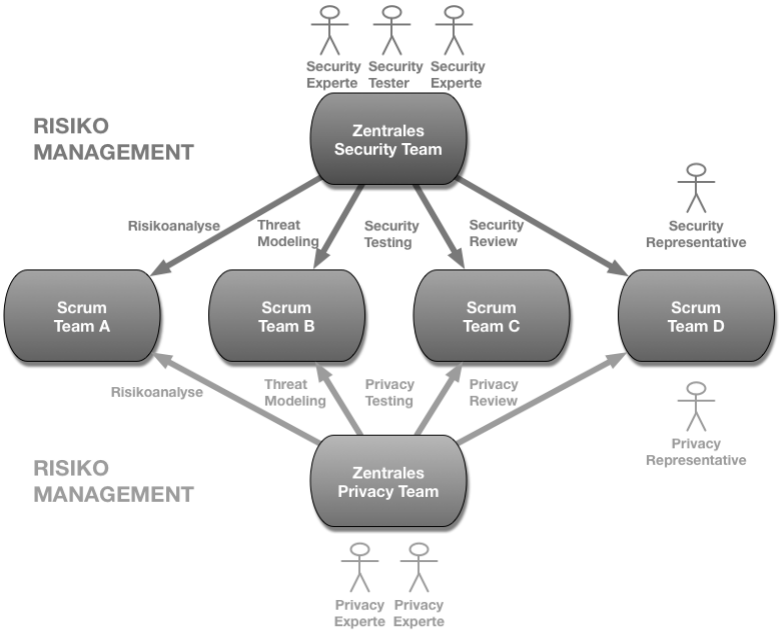

Abbildung 2: Zusätzliche Rollen

Beide Teams sind als zentrale Instanzen zu verstehen, die auch mehr als ein Scrum Team unterstützen können. Dies eignet sich vor allem bei Projekten mit verteilten Teams. In Security- und Privacy-Sprints werden *Evil User Stories* von diesen Teams übernommen. Wird die *Velocity* des Entwicklungsteams bestimmt, so muss der Anteil der extern vergebenen Tätigkeiten klar ersichtlich sein und herausgerechnet werden. Dies ist wichtig, um die Kapazität des Entwicklungsteams für künftige *Sprint Plannings* zu bestimmen.

Folgende neue Rollen ergeben sich dadurch:

- Security-Experten und Tester: Als Mitglieder des Security Teams leisten sie Hilfestellung oder übernehmen Aufgaben im Bereich Security (z.B. Risikoanalyse, Threat Modeling, Security Testing, Code Reviews mit Fokus auf Security und Secure Coding).
- Privacy-Experten: Als Mitglieder des Privacy Teams kümmern sie sich um die Themen Datenschutz, Datensicherheit und Privatsphäreschutz. Sie unterstützen das Entwicklungsteam im Bereich Privacy (z.B. Risikoanalyse, Threat Modeling, Privacy Testing, Privacy Reviews).

Die Teams werden durch einen *Privacy Representative* (PR) bzw. einen *Security Representative* (SR) vertreten. Sie haben eine ganzheitliche Sicht auf Entwicklung und Infrastruktur bezüglich Security, Risikomanagement bzw. Datenschutz, Datensicherheit und Privatsphäreschutz. Das qualifiziert sie als zusätzliche PO für das Entwicklungsteam, d.h. sie dürfen unter Absprache mit den anderen PO, für ihre Bereiche relevante PBI und/oder Akzeptanzkriterien definieren, erweitern und priorisieren. Es gibt jedoch weiterhin nur ein *Product Backlog* mit eindeutiger Priorisierung.

Darüber hinaus hat ein Representative folgende Aufgaben: Er vertritt sein Team in den Scrum Meetings, koordiniert die Mitglieder des jeweiligen Teams (während des Sprints werden Tasks zuerst dem SR zugewiesen, der die Tasks weiter an seine Teammitglieder verteilt), führt die sofortige Risikoanalyse und das Threat Modeling während des *Backlog Refinement* und *Planning Meetings* für kleine *User Stories* durch, fordert und plant periodische Sprints in seinem Bereich, berücksichtigt die Ergebnisse des Sprints des jeweiligen Bereichs im *Product Backlog*, priorisiert die *User Stories* in Absprache mit den anderen PO, verpflichtet sich zur Teilnahme an den *Backlog Refinement*-, *Planning*-, *Review*- und *Retrospective Meetings* und nimmt optional bei den *Daily Scrums* teil.

Der SR fordert die Implementierung von Security Best Practices und der Maßnahmen zur Gewährleistung der Sicherheit und fordert kritische Software/Framework-Updates. Ein PR hat zusätzlich die Gewährleistung der Compliance (Datenschutzrecht, ggf. Normen etc.) und die Umsetzung von PbD-Prinzipien und der Maßnahmen zu deren Gewährleistung zu verantworten.

4. Diskussion und erste Erfahrungen aus dem Projekt ePartizipation

Im Projekt ePartizipation wird der soeben beschriebene Prozess angewandt, allerdings in einer vereinfachten Form, da sowohl das Team als auch die zu entwickelnde Software verhältnismäßig klein sind. Die Entwicklung läuft erst seit wenigen Monaten, doch bisher erweist sich der Prozess als praxistauglich und gut umsetzbar. Privacy- und Security-relevante *(technische) User Stories* finden ihren Weg in das *Product Backlog*. Deren Aufwand wird gemeinsam im *Sprint Planning* mit klassischen *User Stories* in *Story Points* geschätzt. Dies erhöht objektiv automatisch die Vergleich- und Priorisierbarkeit bei gleichzeitiger zunehmender Sensibilisierung des Entwicklungsteams für Privacy und Security. Letzteres wird vor allem deutlich durch viele einschlägige Rückfragen des Entwicklungsteams an die Representatives während der *Sprint Plannings* und *Backlog Refinement Meetings*. Als Konsequenz zeigt sich, dass während der *Daily Scrums* die Anwesenheit der Representatives nicht immer erforderlich ist. Dies wird vor allem in regulären Sprints so gelebt. In Privacy- und Security Sprints ist eine ständige Koordination zwischen den Representatives bzw. den externen Teams und dem Entwicklungsteam durch die *Daily Scrums* weiterhin notwendig.

Obwohl die Existenz mehrerer PO als potenzielles Problem identifiziert wurde, zeigt sich kein Interessenkonflikt zwischen den Representatives, die formal die Rolle eines PO bekleiden, und dem traditionellen PO. Um Konflikte zu vermeiden, wäre es allerdings denkbar, die Representatives lediglich als Stakeholder mit besonderen Pflichten und nicht als PO einzustufen.

Die bisherigen Praxiserfahrungen mit dem beschriebenen Prozess sind somit äußerst positiv. Aufgrund der erst kurzzeitigen Anwendung des Prozesses in einem einzelnen Projekt, in dem überdies Entwickler tätig sind, die großes Interesse an den Themen Privacy und Security haben, können allerdings keine allgemeinen Aussagen über die generelle Umsetzbarkeit und Eignung des präsentierten Prozesses abgeleitet werden. Mit dieser Publikation sollen daher auch andere animiert werden, den Prozess oder Elemente davon in der Praxis einzusetzen, sodass einerseits den Konzepten Privacy und Security by Design Vorschub geleistet wird und andererseits weitere wertvolle Erfahrungen mit dem Einsatz des vorgestellten Prozessmodells gesammelt werden.

5. Danksagung

Das Projekt ePartizipation wird finanziert im Sicherheitsforschungs-Förderprogramm KIRAS vom Bundesministerium für Verkehr, Innovation und Technologie.