Sistematični pregled literature agilnih in vitkih pristopov k razvoju varne programske opreme

Anže Mihelič[1,2,3], Simon Vrhovec[1], Tomaž Hovelja[3]
[1]Univerza v Mariboru, Fakulteta za varnostne vede, Kotnikova 8, 1000 Ljubljana
[2]FernUniversität in Hagen, Fakultät für Mathematik und Informatik, Universitätsstraße 47, 58097 Hagen
[3]Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana anze.mihelic@um.si, simon.vrhovec@um.si, tomaz.hovelja@fri.uni-lj.si

Izvleček

Izvedli smo sistematičen pregled literature v štirih bibliografskih zbirkah, pri čemer smo se osredotočali na pomanjkljivosti trenutnih preglednih del. Identificirali smo 23 predlaganih pristopov, ki so bili večinoma teoretični. Le 21,7 odstotkov pristopov je bilo empirično preverjenih v industrijskih okoljih. Vsi identificirani pristopi temeljijo na predpostavki, da varnost v razvojnem procesu ni zadostno upoštevana, ker varnostni elementi niso sestavni in stalni del agilnih metod. Najpogosteje dodani varnostni elementi so procesi (48 odstotkov), sledita kombinacija procesov in artefaktov (26 odstotkov) in kombinacija procesov, artefaktov in vlog (13 odstotkov).
Ključne besede: metodologija, informacijska varnost, računalniška varnost, agilne metode, razvoj varne programske opreme

Abstract

We conducted a systematic literature survey in four bibliographic databases. We focused on secure software development with special attention to the shortcomings of existing surveys. We identified 23 approaches. Most identified approaches were theoretical and only 21.7 percent were empirically tested in an industrial setting. All identified approaches are based on the assumption that security is not considered in the development process since security elements are not an integral and permanent part of agile methods. The most frequently proposed security elements are processes (48 percent), followed by combination of processes and artefacts (26 percent) and combination of processes, artefacts and roles (13 percent).
Keywords: Methodology, information security, computer security, agile methods, secure software development.

1 UVOD

Agilne metode razvoja programske opreme (npr. Scrum in Extreme Programming) so se pojavile kot odgovor na pomanjkljivosti tradicionalnih metod [Oueslati et al., 2015]. Glavne značilnosti agilnih metod so iterativni in postopni pristop, samoorganizirajoče se skupine, vsakodnevno komuniciranje med člani skupine in hitre povratne informacije [Adelyar and Norta, 2016, Pohl and Hof, 2015]. Zaradi teh značilno- sti agilne metode predstavljajo zelo prilagodljiv, učinkovit in hiter pristop k razvoju programske opreme [Gwanhoo and Weidong, 2010, Jyothi and Rao, 2011, Othmane et al., 2014]. Ker je osrednji poudarek agil- nih metod na funkcionalnih zahtevah razvite programske opreme, varnost pa ima močno nefunkcionalno kakovostno komponento, agilne metode niso povsem primerne za varen razvoj programske opreme [Tøndel and Jaatun, 2020, Bishop and Rowland, 2019]. Poleg omenjenega je manj kot 50 odstotkov tradici- onalnih dejavnosti zagotavljanja varnosti združljivih z agilnimi metodami, povsem primernih pa je manj kot 10 odstotkov [Beznosov and Kruchten, 2005].

S selitvijo trga s prodaje programske opreme kot izdelka na prodajo programske opreme kot storitve, je ideja o združevanju razvojnih in operativnih sku-

pin rezultirala v paradigmi DEVOPS [Lwakatare et al., 2016, Myrbakken and Colomo-Palacios, 2017]. Tako kot agilni, tudi DEVOPS, sam po sebi ni prilagojen za razvojne varne programske opreme [Lee, 2018]. Kot rešitev te pomanjkljivosti se je pojavil DEVSECOPS, ki DEVOPS združuje z varnostjo [Allison et al., 2020]. Zavzema se za implementacijo varnostnih elementov v vsaki fazi razvojnega procesa. Bolj kot nabor posebnih orodij in ukrepov, DEVSECOPS predstavlja ogrodje za razvoj varne programske opreme [Allison et al., 2020, Bezdedeanu, 2019, Kiuwan, 2019].

Č etudi je v literaturi mogoče zaslediti nekaj sistematičnih pregledov literature [Inayat et al., 2015, Mellado et al., 2010] in primerjalnih študij [Curcio et al., 2018, Khan and Ikram, 2017] na področju agilnih pristropov k razvoju programske upreme, so se takšni pregledi, ki se osredotočajo na razvoj *varne* pro- gramske opreme [Rindell et al., 2017, Kasauli et al., 2018, Barbosa and Sampaio, 2017] in ogrodja DEVSE- COPS [Myrbakken and Colomo-Palacios, 2017] pojavili šele pred kratkim. Pregledi literature na področju agilnega varnega razvoja programske opreme so k problemu pristopali z različnih zornih kotov. Tako po- nujajo celovite vodiče za izboljševanje varnostih ukrepov v agilnih projektih [Barbosa and Sampaio, 2017], pregled agilnih pristopov k razvoju varnostno kritičnih sistemov [Kasauli et al., 2018], pregled agilnih pri- stopov inženirstva zahtev [Villamizar et al., 2018], pregled agilin metod razvoja varne programske opreme [Rindell et al., 2017], in pregled izzivov in rešitev na tem področju [Oueslati et al., 2015, Riisom et al., 2018]. Neodvisno od agilnih pristopov, se je le en pregledni prispevek osredotočal na ogrodje DEVSECOPS [Myrbakken and Colomo-Palacios, 2017], ki pa se osredotoča na poskus iskanja ustrezne opredelitve de- finicije tega ogrodja in ne na varnostne elemente in rešitve, ki bi bile lahko vključene v proces razvoja in vzdrževanja.

Naš prispevek gradi na omenjenih pregledih literature, vendar se osredotoča na širši časovni okvir, poleg agilnih pa vključuje tudi vitke pristope, ki se pogosto omenjajo skupaj z ogrodjem DEVSECOPS. V prispevku bomo odgovorili na naslednja raziskovalna vprašanja.

- **RV1:** Katera kategorija elementov je najpogosteje predlagana kot rešitev za razvoj varne programske opreme?
- **RV2:** Kako so bili predlagani pristopi empirično preverjeni?
- **RV3:** Na kakšen način je varnost vključena v predlagane pristope?

Da bi raziskali kakšne rešitve ponuja obstoječa literatura, smo opravili pregled agilnih in vitkih pristopov k razvoju varne programske opreme. Sistematično smo pregledali relevantno literaturo v znanstvenih bibliografskih zbirkah. Identificirane elemente smo nato razvrstili v tri kategorije (artefakti, vloge, procesi) in analizirali njihovo kompatibilnost z agilnimi in vitkimi pristopi glede na njihovo osnovno predpostavko. Prav tako je cilj prispevka identificirati vse pristope k razvoju varne programske opreme, ki so bili zabeleženi v znanstveni literaturi v zadnjih enajstih letih.

2 METODA

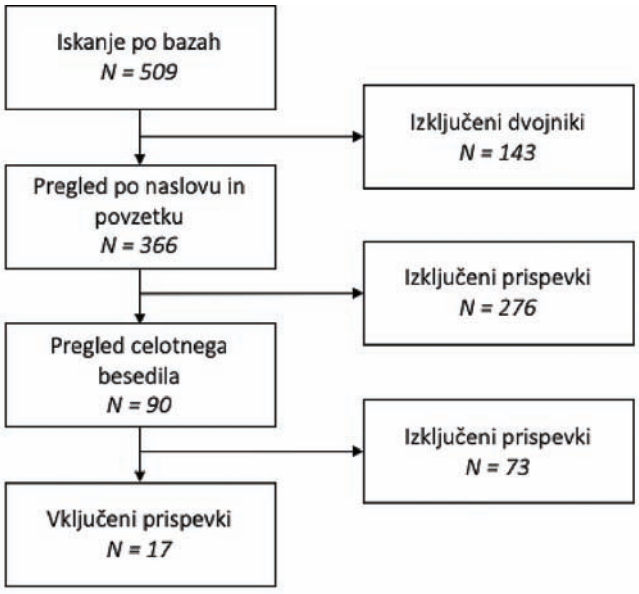
Opravili smo sistematičen pregled literature, ki je naslavljala probleme, usmerjene v razvoj varne programske opreme. Pregled je obsegal prispevke na konferencah in članke v znanstvenih revijah, objavljenih od leta 2009. Slika 1 prikazuje proces pregleda s številom vključenih del v posameznem koraku.

Pregled je bil opravljen po naslednjem postopku. V znanstvenih bibliografskih zbirkah (ACM DL, IEEE Xplore, Scopus in Web of Science) smo 8. januarja 2020 izvedli poizvedbo po naslovu in povzetku s kombinacjo ključnih besed *agilno, vitko, varno, programska oprema, razvoj, inženiring, metoda* in *upravljanje*. Za poizvedbo je bil uporabljen iskalni niz:

(agile **OR** lean) **AND** (secur* **AND** software **AND** (development **OR** engineering) **AND** (method* **OR** manag*)),

ki je bil prilagojen posamezni bibliografski zbirki. Iskanje je vrnilo skupno $N = 509$ zadetkov od leta 2009. Vse rezultate smo izvozili in jih shranili v lastno zbirko podatkov.

V naslednjem koraku smo odstranili vse podvojene zapise, kar je rezultiralo v $N = 366$ unikatnih bibliograf- skih zapisih, ki smo jih nato na podlagi pregleda naslova in povzetka izločali skladno z vključitvenimi in izključitvenimi kriteriji predstavljenimi v tabeli 1. V pregled celotnega besedila so bila vključena tudi dela, pri katerih iz naslova ali povzetka ni bilo



Slika 1: **Proces sistematičnega pregleda literature.**

jasno ali zadoščajo omenjenim kriterijem. Končno je bilo pregledano celotno besedilo $N = 66$ del in vključenih $N = 17$ del, v katerih so bili predlagani novi pristopi k naslavljanju problema razvoja varne programske opreme. Rezultati po bibliografskih bazah so predstavljeni v tabeli 2. Dodaten pregled izveden po metodi snežne kepe, je rezultiral v dodatnih $N = 6$ delih objavljenih leta 2009 ali kasneje, katerih izvor je bilo mogoče najti med deli, vključenimi v sistematični pregled literature.

3 REZULTATI IN RAZPRAVA

V nadaljevanju predstavljamo obstoječe pristope, ki predlagajo posebne rešitve za vključitev varnostnih praks v postopek razvoja programske opreme. Dela predstavljamo v tabelah 3 in 4. Posamezni pristopi so razvrščeni v skupine glede na to, na katero kategorijo elementov ali njihovo kombinacijo je osredotočen: procese (P), artefakte (A) ali vloge (V).

Odgovor na RV1: Rezultati nakazujejo, da so najpogosteje (48%) predlagani elementi, ki spadajo v kateogirijo procesov [Mougouei et al., 2013, Daud, 2010, Pohl and Hof, 2015, Rygge and Jøsang, 2018, Williams and Meneely, 2010, Tøndel et al., 2019, Yu and Le, 2012, Singh, 2018, Nguyen and Dupuis, 2019, Koc et al., 2019, Giacalone et al., 2014]. Sledi mu kombinacija procesov in artefaktov (26%) [Stålhane and Johnsen, 2017, Singhal and Singhal, 2011, Othmane et al., 2014, Maier et al., 2017,

Tabela 1: **Vključitveni in izključitveni kriteriji**

Vključitveni kriteriji	Izključitveni kriteriji
Članek objavljen v znanstveni reviji ali prispevek na konferenci	Teoretični pregledni članek
Povezan z razvojem varne programske opreme	Članek ni v angleškem jeziku ne
Povezan z nefunkcionalnimi zahtevami	Ni povezan z agilnimi ali vitkimi metodami
Povezan z razvojem varnostno-kritičnih sistemov	Objavljen leta 2009 ali kasneje
Ima izvirni znanstveni prispevek	Celotno besedilno ni dostopno

Tabela 2: **Zadetki po bibliografskih bazah.**

Biografska zbirka	Skupaj zadetkov	Pregled celotnega besedila	Vključwni
Web of Science	106	27	7
Scopus	171	30	8
IEEE Xplore	166	5	0
ACM DL	66	4	2

Tabela 3: **Predlagani pristopi, ki izvajajo iz sistematičnega pregleda literature. Črke v oglatih oklepajih nakazujejo kateri tip predlaganih elementov je dominanten (P - procesi, A - artefakti, V - vloge)**

Vir	Metoda	Kratek opis
[Ghani et al., 2014]	Študija primera v industrijskem okolju	Predlagani pristop v Scrum metodo uvaja varnostno specifičen seznam zahtev – security backlog. **[A]**
[Baca et al., 2015]	Kvazieksperiment v industrijskem okolju	Pristop Security-Enhanced Agile Software Develop- ment Process (SEAP) predlaga vključitev skupine za varnost, ki vključuje več varnostnih vlog. **[P,V]**
[Mohino et al., 2019]	Teoretičen z anketo	Secure Software Development Life Cycle (S-SDLC) predlaga številne dodatne procese, artefakte in vloge skozi na vsakem koraku razvojnega procesa. **[P,A,V]**
[Firdaus et al., 2014]	Študija primera med študenti	Secure Feature Driven Development (SFDD) gradi na znanem pristopu Feature Driven Development (FDD) z osredotočanjem na varnost vseh elementov v posameznih fazah. Metoda dodaja dve dodatni varnostni fazi in predlaga uvedbo skrbnika za varnost (security master). **[P,A,V]**
[Giacalone et al., 2014]	Študija primera v industrij- skem okolju	Predlagana metoda vključuje dva glavna procesa: (1) varnostno raziskovanje (security survey) zagotavlja celovito karakterizacijo informacijsko-komunikacijskih tehnologij in poslovnih storitev lastnika (naročnika); (2) triaža varnosti (security triage) za identifikacijo ravni ustreznosti zahtev za oceno varnosti. **[P]**
[Ionita et al., 2019]	Študija primera v industrij- skem okolju	Predlagani pristop uvaja identifikacijo, določanje prednosti in izvajanje varnostnih zahtev v treh fazah: (i) ocena tveganja, (ii) določitev prednostnih nalog varnostnih zahtev in (iii) dopolnjevanje seznama zahtev (product backlog). **[P,A]**
[Koc et al., 2019]	Anketa	Trustworthy Scrum v klasičen Scrum vnaša z varnostjo povezane aktivnosti v vsak sprint (npr. statična analiza, pregled kode). Poleg tega predlaga tudi dodaten, v varnost usmerjen sprint.**[P]**
[Maier et al., 2017]	Anketa	Secure Scrum v celoten proces vpeljuje številne prakse v varnost usmerjenega razvoja, ki jih predlagajo ISO standardi. **[P,A]**
[Maria et al., 2015]	Študija primera v akademskem okolju	ScrumS dodaja tehniko Secure Project za Scrum. Tehnika predvideva dodajanje številnih dodatnih elementov v obstoječo metodologijo (npr. varnostne uporabniške zgodbe, analiza tveganj itd.). **[P,A]**
[Nguyen and Dupuis, 2019]	Teoretičen	Technology Development Lifecycle je osnovan na ogrodju DevSecOps in skuša vzdrževati hitre povratne informacije med vsemi deležniki v razvojnem procesu. **[P]**
[Othmane et al., 2014]	Študija primera v industrijskem okolju	Pristop gradi na razširanju posameznih faz v razvojnem procesu z dodajanjem različnih z varnostjo povezanih aktivnosti. **[P,A]**
[Rindell et al., 2015]	Teoretičen	Predlagani model dodaja varnostno-specifične prilagoditve in dodatke vlogam, procesom in artefaktom v klasični metodi Scrum. **[P,A,V]**
[Singhal and Singhal, 2011]	Teoretičen	Pristop Agile Security Framework v razvojni proces vpeljuje varnost v vsaki fazi življenjskega cikla skozi uvajanje varnostnih elementov, kot so zgodbe zlorab (abuser stories), izobraževanje o varnosti in kategori- ziranje varnostnih izahtev.**[P,A]**
[Singh, 2018]	Teoretičen	Predlagani model se osredotoča na naslavljanje avtentikacije in preverbe v procesu razvoja programske opreme. **[P]**
[Yu and Le, 2012]	Teoretičen	Metoda SQUARE-R gradi na metodi SQUARE pri čemer vnaša neprekinjeno obvladovanje tveganj. Tako zagotavlja uresničevanje varnostnih poslovnih ciljev hkrati.**[P]**
[Stålhane and Johnsen, 2017]	Teoretičen	SafeScrum prilagaja Scrum tako, da seznam zahtev razdeli na dva dela (standardni in varnostni) in spremeni analizo vpliva za uspešno upravljanje z zahtevami. **[P,A]**
[Tøndel et al., 2019]	Teoretičen	Sestanki Security Intention Recap Meetings so oblikovani za evalviranje trenutnih praks povezanih z varnostnimi namerami v določenem projektu in sprejemanje odločitev kako naslavljati probleme. **[P]**

Tabela 4: **Predlagani pristopi, ki izvirajo iz pregleda z metodo snežne kepe. Črke v oglatih oklepajih nakazujejo kateri tip predlaganih elementov je dominanten (P – procesi, A – artefakti, V – vloge).**

Vir	Metoda	Kratek opis
[Williams and Meneely, 2010]	Teoretičen	Protection poker je pristop z igrifikacijskimi elementi in temelji na Wideband Delphi metodi. **[P]**
[Rygge and Jøsang, 2018]	Teoretičen	Pristop Threat poker je predlagan po vzoru prej omenjenega Protection poker z dodano kompleksnostjo. **[P]**
[Pohl and Hof, 2015]	Kvazieksperiment v akademskem okolju	Secure Scrum je razširitev tradicionalne Scrum metode. Predlaga štiri dodatne aktivnosti: (i) identifikacija varnostnih vprašanj, (ii) uvajanje varnostnih komponent, (iii) preverjanje / pregled in (iv) definicija končane zadeve (definition of done). **[P]**
[Siiskonen et al., 2014]	Teoretičen	Generic security user stories so vnaprej pripravljene zgodbe, ki jih lahko razvojne ekipe uporabijo tudi v pri- meru, ko naročnik nima zadostnega znanja, da bi var- nostne komponente eksplicitno zahteval. Predlagane varnostne zgodbe so nato skupinjene v večje enote imenovane varnostne teme. **[A]**
[Daud, 2010]	Teoretičen	Secure Software Lifecycle (S-SL) predvideva uvajanje varnostnih elementov na vsaki stopnji življenjskega cikla razvoja programske opreme z (i) varnostno ana lizo, (ii) varnostnim načrtovanjem in izvajanjem ter (iii) varnostnim testiranjem. **[P]**
[Mougouei et al., 2013]	Teoretičen	S-Scrum gradi na dveh novih aktivnostih, in sicer Security Spikes: (i) konica za varnostno analizo (ii) konica za varnostno modeliranje. Konkretne tehnike niso predlagane. [P]

Ionita et al., 2019] in nato kombinacija vseh treh kategorij elementov (13%) v obliki celostnih rešitev [Mohino et al., 2019, Firdaus et al., 2014, Rindell et al., 2015]. Najmanjši skupini sta skupina, ki se osredotoča samo na artefakte (9%) in kombinacija procesov in vlog (4%). Pristopov, ki bi bili usmerjeni le v dodajanje novih vlog nismo zasledili.

Odgovor na RV2: Med identificiranimi pristopi je le pet pristopov (21,7 %) empirično preverjenih s študijo primera v realnem industrijskem okolju [Ghani et al., 2014, Baca et al., 2015, Giacalone et al., 2014, Ionita et al., 2019], trije pristopi (13,1 %) pa so bili preverjetni v akademskem okolju s pomočjo študentov [Firdaus et al., 2014, Maria et al., 2015, Pohl and Hof, 2015]. Nadaljnji trije pristopi (13,1 %) so bili predmet empirične raziskave s pomočjo ankete [Mohino et al., 2019, Koc et al., 2019, Maier et al., 2017]. Več kot polovica predlaganih pristopov (52,1 %) ni bila empirično preverjena.

Odgovor na RV3: Vsi identificirani pristopi temeljijo na predpostavki, da varnost ne velja za nepogrešljivo kakovost programske opreme, ker ni sestavni del posamezne metode. Zato avtorji teh pristopov predlagajo, da se varnostni elementi stalno vključijo v postopek razvoja programske opreme. Identificirane pristope je mogoče razvrstiti v tri skupine. Pristope, ki se osredotočajo na: (a) dvig motivacije za razvoj varne programske opreme, (b) povečevanje znanja s področja varnosti in (c) prilagajanje metode razvoja programske opreme z varnostno-specifičnimi elementi.

Glede na zorni kot s katerega rešujejo problem razvoja varne programske opreme, lahko pristope razde- limo v tri skupine: dodajanje varnostnih elementov z namenom (a) dvigovanja *motivacije*, (b) vključevanje *varnostno-specifičnega znanja* in (c) *ostali varnostno-specifični elementi*. *Motivacija* je bila naslovljena z igri- fikacijo, kot sta *Protection poker* [Williams and Meneely, 2010] in *Threat poker* [Rygge and Jøsang, 2018], oba navdihnjena po pristopu *Planning poker*, na konseznu temelječi tehniki ocenjevanja težavnosti nalog v agilnih projektih [Grenning, 2002]. Oba pristopa temeljita na ideji vključevanja v obstoječe agilne metode in vključujeta oceno tveganj za varnost in zasebnost, ki ju določata enostavnost izvedbe napada in njegova resnost. Vključitev *varnostno-specifičnega znanja* je predlagalo več avtorjev, najpogosteje z dodaja- njem novih vlog, povezanih z varnostjo [Baca et al., 2015]. Te vloge so običajno skrbnik za varnost (angl. *Security master*) [Azham et al., 2011], varnostni strokovnjak (angl. *Security expert*) [Musa et al., 2011] ali penetracijski tester (angl. *Penetratioin tester*) [Tomanek and Klima, 2015]. Naloga varnostnih strokovnjakov je prepoznavanje in zmanjševanje varnostnih

tveganj, upravljanje varnostnih pomanjkljivosti, in izvajanje varnostnih in penetracijskih testiranj. Najbolj celovit pristop so predlagali [Baca et al., 2015] in vključuje niz varnostnih vlog, ki tvorijo varnostno skupino, sestavljeno iz štirih varnostnih vlog z različ- nimi pristojnostmi (tj. tehničnimi, netehničnimi in pravnimi). Najpogostejši predlagani *varnostno--specifični elementi* so varnostne uporabniške zgodbe (kot prilagoditev primerov zlorabe (angl. *Misuse and abuse cases*) iz tradicionalne metode razvoja varne programske opreme [Lee and Park, 2016]) ali njihove razli- čice, ki zagotavljajo pregled nad varnostnimi zahtevami in so lahko vključene v v obstoječi seznam zahtev (angl. *Product backlog*) ali pa v posebni in namenski seznam varnostnih zahtev (anlg. *Security backlog*) [Azham et al., 2011, Mougouei et al., 2013, Siiskonen et al., 2014].

Trenutna literatura ne ponuja nedvoumne in celovite rešitve za razvoj varne programske opreme z agil- nimi metodami, primernimi za manjša podjetja z omejenimi proračuni [Siiskonen et al., 2014]. Obstoječi pristopi imajo več problemov. Prvič, predlagajo niz varnostnih elementov (vloge, procesi, artefakti), ki jih je treba trajno vključiti v obstoječo agilno metodo [Baca et al., 2015]. Pogosta težava takšnih pristopov je, da ogrožajo produktivnost, agilnost in znatno povečajo stroške projektov razvoja programske opreme [Boström et al., 2006]. Drugič, ne upoštevajo situacijskih dejavnikov, kot so obstoječa agilna metoda, medo- sebni odnosi znotraj razvojne skupine in podjetja ter drugih dejavnikov, ki skladno s kontingenčno teorijo pomembno vplivajo na uspešnost projekta [Fiedler, 1964, Song et al., 2018]. Tretjič, varnostni elementi so zasnovani za točno določeno metodo (npr. Scrum ali XP), kar razvojno podjetje omeji na metodo, za katero so bili zasnovani [Ghani et al., 2014, Türpe and Poller, 2017, Azham et al., 2011]. Č etrtič, igrifikacija kot pristop k dvigu motivacije znotraj razvojne skupine lahko po daljših obdobjih uporabe povzroči stres in napetost v delovnem procesu [Platonova and Be¯ rziša, 2017]. Petič, da bi odpravili pomanjkanje znanja v razvojni skupini, opredeljeni pristopi predlagajo vključitev novih varnostnih vlog, običajno z zaposlitvijo dragih strokovnjakov za varnost [Baca et al., 2015].

Glede na ugotovitve, ki izhajajo iz sistematičnega pregleda literature in predvsem identificirane prednosti in pomanjkljivosti omenjenih pristopov, bi bilo smiselno iskati trajno rešitev, ki bi bila dosežena z le začasno prilagoditvijo agilne metode. Idealno bi morala biti predlagana rešitev prilagodljiva različnim agilnim metodam, ki jih podjetja uporabljajo. Takšno rešitev deloma ponujajo pristopi, ki uvajajo varnostno znanje skozi različne varnostne vloge (npr. [Azham et al., 2011, Musa et al., 2011, Tomanek and Klima, 2015]). Vendar temeljno pomanjkljivost tovrstnih pristopove predstavlja tudi do 500-odstotno povišanje stroškov projekta [Baca et al., 2015]. Postopno pridobivanje znanja in s tem tudi ozaveščenosti o varnosti bi bilo tako smiselno delegirati na razvojno ekipo. Začasni značaj te rešitve bi bil dosežen z implementacijo pristopa do trenutka, ko ravni varnostne ozaveščenosti in znanja razvojne ekipe ne dosežejo zadovoljive ravni. Ponovitve bi bile potrebne le za ohranitev želene ravni varnostnega znanja. Ta zasnova bi torej (a) omogočala izogibanje zaposlovanju dragih strokovnjakov za varnost, ki so običajno zadolženi za vnašanje varnostnega znanja, in (b) reševala problem togosti in višanja stroškov obstoječih pristopov, ker jih je v obstoječo agilno metod treba vključiti trajno. Poleg tega bi na podlagi predpostavke, da varnostno znanje in varnostna zavest višata motivacijo, (c) naslavljal vprašanje motivacije brez elementov igrifikacije, ki po daljši uporabi lahko v ekipo vnaša stres in napetost [Platonova and Be¯ rziša, 2017]. Takšen pristop bi predstavljal razmeroma trajno rešitev z začasnim dodajanjem elementov k obstoječi metode, pri čemer ne bi postal sestavni del obstoječe metode.

4 OMEJITVE IN NADALJNJE DELO

Kot vsaka druga ima tudi ta raziskava omejitve. Prvič, osredotočili smo se na štiri glavne bibliografske zbirke podatkov, pri čemer je nekaj manjših ostalo nepregledanih oz. so bile pregledane le posredno, skozi dela, vključena v sistematični pregled literature. Drugič, pregled je opravil en raziskovalec, kar izpostavlja možnost raziskovalčeve pristranskosti. Tretjič, pregled je bil opravljen med deli, objavljenimi od leta 2009, kar omeji širino pregleda na določeno časovno obdobje. Tako so dela, objavljena pred tem datumom, iz predgleda izpuščena.

V nadaljnjem delu bi se bilo smiselno osredotočiti na analizo prednosti in slabosti vsake od identificiranih metod, kar bi rezultiralo v bolj poglobljenem pregledu dejanskega stanja. Smiselno bi bilo tudi razširiti časovni okvir (npr. od leta 2001, ko so se agilni pristopi začeli pojavljati) in razširiti iskalni niz,

kar bi omogočalo bolj celovit pregled področja. Ob upoštevanju širšega pregleda bi bilo treba predlagane pristope razvrstiti v časovne okvirje (npr. petletna obdobja), kar bi omogočalo umestitev posameznega pristopa v določen socio-tehnološki kontekst.

5 ZAKLJUČEK

Izvedli smo sistematični pregled literature agilnih in vitkih pristopov k razvoju varne programske opreme. Pregled se je osredotočil na relativno široko časovno obdobje in temeljil na razmeroma širokem iskalnem nizu. S tem smo naslavljali odprta vprašanja preteklih del. Rezultati nakazujejo, da si obstoječi pristopi delijo skupne pomanjkljivosti. Predvsem je večina del teoretičnih, testiranih v akademskem okolju ali empiričnih z anketo. Popolnoma razumljivo je, da je testiranje pristopov v industrijskem okolju težavno, pa vendar le takšno testiranje lahko poda vpogled v izvedljivost in učinkovitost pristop ter tiste posledice, ki niso bile pričakovane pri teoretičnem modeliranju pristopa ali njegovem testiranju v kontroliranem okolju.

Identificirane pomanjkljivosti na področju agilnih in vitkih pristopov k razvoju varne programske opreme usmerjeno kliče k iskanju rešitve, ki ne bi bila trajno vključena v razvojni proces. Tak pristop bi lahko ponudil dolgotrajno rešitev, obenem pa obetal učinkovitost, brez znatnega povečanja stroškov ali ogrožanja agilnosti metode, ki jo razvojna skupina že uporablja.

LITERATURA

[1] [Adelyar and Norta, 2016] Adelyar, S. H. and Norta, A. (2016). Towards a Secure Agile Software Deve- lopment Process. In *10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, pages 101–106.

[2] [Allison et al., 2020] Allison, I., Tiplitsky, J., Kennedy, S., Kersten, N., Lietz, S., Lim, F., Nikulshin, M., Price, C., Dhungel, R., Rose, K., and Sherman, B. (2020). The DevSecOps Manifesto.

[3] [Azham et al., 2011] Azham, Z., Ghani, I., and Ithnin, N. (2011). Security backlog in scrum security practices. *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*, pages 414–417.

[4] [Baca et al., 2015] Baca, D., Boldt, M., Carlsson, B., and Jacobsson, A. (2015). A novel security-enhanced agile software development process applied in an industrial setting. *10th International Conference on Availability, Reliability and Security*, pages 11–19.

[5] [Barbosa and Sampaio, 2017] Barbosa, D. A. and Sampaio, S. (2017). Guide to the Support for the En- hancement of Security Measures in Agile Projects. In *Brazilian Workshop on Agile Methods*, pages 25–31.

[6] [Bezdedeanu, 2019] Bezdedeanu, A. (2019). DevSecOps is Not a Role or Technology: It's a Culture to Wholly Embrace.

[7] [Beznosov and Kruchten, 2005] Beznosov, K. and Kruchten, P. (2005). Towards agile security assurance. In *Proceedings New Security Paradigms Workshop*, pages 47–54.

[8] [Bishop and Rowland, 2019] Bishop, D. and Rowland, P. (2019). Agile and Secure Software Development: An Unfinished Story. *Issues in Information Systems*, 20(1):144–156.

[9] [Boström et al., 2006] Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., and Kruchten, P. (2006). Extending XP practices to support security requirements engineering. *2006 international workshop on Software engineering for secure systems,* page 11.

[10] [Curcio et al., 2018] Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 139(1):32– 50.

[11] [Daud, 2010] Daud, M. I. (2010). Secure software development model: A guide for secure software life cycle. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, pages 724–728, Hong Kong.

[12] [Fiedler, 1964] Fiedler, F. E. (1964). A theory of leadership effectiveness. In Berkowitz, L., editor, *Advances in experimental social psychology.* Academic Press, New York.

[13] [Firdaus et al., 2014] Firdaus, A., Ghani, I., and Jeong, S. R. (2014). Secure Feature Driven Development (SFDD) Model for Secure Software Development. In *Procedia - Social and Behavioral Sciences,* volume 129, pages 546–553. Elsevier B.V.

[14] [Ghani et al., 2014] Ghani, I., Azham, Z., and Jeong, S. R. (2014). Integrating software security into agile-Scrum method. *Transactions on Internet and Information Systems*, 8(2):646–663.

[15] [Giacalone et al., 2014] Giacalone, M., Paci, F., Mammoliti, R., Perugino, R., Massacci, F., and Selli, C. (2014). Security Triage: An Industrial Case Study on the Effectiveness of a Lean Methodology to Identify Security Requirements Matteo. In *Symposium on Empirical Software Engineering and Measurement,* pages 1–8.

[16] [Grenning, 2002] Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning.

[17] [Gwanhoo and Weidong, 2010] Gwanhoo, L. and Weidong, X. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data. *MIS Quarterly*, 34(1):87–114.

[18] [Inayat et al., 2015] Inayat, I., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51:915–929.

[19] [Ionita et al., 2019] Ionita, D., Van Der Velden, C., Ikkink, H.-j. K., and Eelko, N. (2019). Towards Risk- Driven Security Requirements Management in Agile Software Development. *Lecture Notes in Business Information Processing*, 350(628):133–144.

[20] [Jyothi and Rao, 2011] Jyothi, V. E. and Rao, K. N. (2011). Effective Implementation of Agile Practices Ingenious and Organized Theoretical Framework. *International Journal of Advanced Computer Science and Applications*, 2(3):41–48.

[21] [Kasauli et al., 2018] Kasauli, R., Knauss, E., Kanagwa, B., Nilsson, A., and Calikli, G. (2018). Safety-critical systems and agile development: A mapping study. In *44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2018*, pages 470–477. IEEE.

[22] [Khan and Ikram, 2017] Khan, N. F. and Ikram, N. (2017). Security requirements engineering: A systematic mapping (2010-2015). In *2016 International Conference on Software Security and Assurance*, pages 31–36.

[23] [Kiuwan, 2019] Kiuwan (2019). The Benefits of a DevSecOps Approach to the SDLC.

[24] [Koc et al., 2019] Koc, G., Aydos, M., and Tekerek, M. (2019). Evaluation of Trustworthy Scrum Emplo- yment for Agile Software Development based on the Views of Software Developers. In *4th International Conference on Computer Science and Engineering*, pages 63–67.

[25] [Lee, 2018] Lee, J. S. (2018). The DevSecOps and Agency Theory. In *29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018*, pages 243–244. IEEE.

[26] [Lee and Park, 2016] Lee, K. H. and Park, Y. B. (2016). Adaption of integrated secure guide for secure software development lifecycle. *International Journal of Security and its Applications*, 10(6):145–154.

[27] [Lwakatare et al., 2016] Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016). Relationship of DevOps to Agile, Lean and Continuous Deployment: A Multivocal Literature Review Study. In Abrahamsson, P., Jedlitschka, A., Nguyen, D. A., Felderer, M., Amasaki, S., and Mikkonen, T., editors, *Lecture Notes in Computer Science*, volume 10027, pages 198–214. Springer, Cham.

[28] [Maier et al., 2017] Maier, P., Ma, Z., and Bloem, R. (2017). Towards a Secure SCRUM Process for Agile Web Application Development. In *Proceedings of the 12th International Conference on Availability, Reliability and Security - ARES '17*, pages 1–8.

[29] [Maria et al., 2015] Maria, R. E., Rodrigues, L. A., and Pinto, N. A. (2015). ScrumS - A model for safe agile development. In *7th International ACM Conference on Management of Computational and CollEctive Intelligence in Digital EcoSystems, MEDES 2015*, pages 43–47.

[30] [Mellado et al., 2010] Mellado, D., Blanco, C., Sánchez, L. E., and Fernández-Medina, E. (2010). A syste- matic review of security requirements engineering. *Computer Standards and Interfaces*, 32(4):153–165.

[31] [Mohino et al., 2019] Mohino, J. d. V., Higuera, J. B., Higuera, J. R. B., and Montalvo, J. A. S. (2019). The application of a new secure software development life cycle (S-SDLC) with agile methodologies. *Electronics (Switzerland)*, 8(11):1–28.

[32] [Mougouei et al., 2013] Mougouei, D., Sani, N. F. M., and Almasi, M. M. (2013). S-Scrum : a Secure Methodology for Agile Development of Web Services. *World of Computer Science and Information Technology Journal (WSCIT)*, 3(1):15–19.

[33] [Musa et al., 2011] Musa, S. B., Norwawi, N. M., Selamat, M. H., and Sharif, K. Y. (2011). Improved extreme programming methodology with inbuilt security. *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*, pages 674–679.

[34] [Myrbakken and Colomo-Palacios, 2017] Myrbakken, H. and Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. In Mas, A., Mesquida, A., and O'Connor, R., editors, *Communications in Computer and Information Science*, volume 770, pages 30–42. Springer, Cham.

[35] [Nguyen and Dupuis, 2019] Nguyen, J. and Dupuis, M. (2019). Closing the feedback loop between UX design, software development, security engineering, and operations. In *Proceedings of the 20th Annual Conference on Information Technology Education*, pages 93–98.

[36] [Othmane et al., 2014] Othmane, L., Angin, L., Weffers, H., and Bhargava (2014). Extending the Agile Development Approach to Develop Acceptably Secure Software. *IEEE Transactions on Dependable and Secure Computing*, 11(6):497–509.

[37] [Oueslati et al., 2015] Oueslati, H., Rahman, M. M., and ben Othmane, L. (2015). Literature Review of the Challenges of Developing Secure Software Using the Agile Approach. In *10th International Conference on Availability, Reliability and Security*, pages 540–547.

[38] [Platonova and Be¯rziša, 2017] Platonova, V. and Be¯rziša, S. (2017). Gamification in Software Development Projects. *Information Technology and Management Science*, 20(1):58–63.

[39] [Pohl and Hof, 2015] Pohl, C. and Hof, H.-J. (2015). Secure Scrum: Development of Secure Software with Scrum. In *The Ninth International Conference on Emerging Security Information, Systems and Technologies Secure*, pages 15–20.

[40] [Riisom et al., 2018] Riisom, K. R., Hubel, M. S., Alradhi, H. M., Nielsen, N. B., Kuusinen, K., and Jabangwe, R. (2018). Software security in agile software development: A literature review of challenges and solutions. In *ACM International Conference Proceeding Series*, pages 1–5.

[41] [Rindell et al., 2015] Rindell, K., Hyrynsalmi, S., and Leppänen, V. (2015). Securing scrum for VAHTI. In *CEUR Workshop Proceedings*, pages 236–250.

[42] [Rindell et al., 2017] Rindell, K., Hyrynsalmi, S., and Leppänen, V. (2017). Busting a myth: Review of agile security engineering methods. In *ACM International Conference Proceeding Series,* pages 1–10.

[43] [Rygge and Jøsang, 2018] Rygge, H. and Jøsang, A. (2018). Threat Poker : Solving Security and Privacy Threats in Agile Software Development. (November):1–15.

[44] [Siiskonen et al., 2014] Siiskonen, T., Sars, C., Vah Sipila, A., and Pietikain, A. (2014). Generic Security User Stories. In Pietikinen, P. and Rning, J., editors, *Handbook of the Secure Agile Sotware Development Life Cycle*, chapter 9, pages 9–14. University of Oulu, Oulu.

[45] [Singh, 2018] Singh, A. (2018). Integrating the Extreme Programing Model with Secure Process for Re- quirement Selection. In *2nd International Conference on Electronics, Communication and Aerospace Technology*, pages 423–426.

[46] [Singhal and Singhal, 2011] Singhal, S. and Singhal, A. (2011). Development of Agile Security Framework Using a Hybrid Technique for Requirements Elicitation. In Unnikrishnan, S., Surve, S., and Bhoir, D., editors, *Advances in Computing, Communication and Control*, pages 178–188.

[47] [Song et al., 2018] Song, M., Wang, P., and Yang, P. (2018). Promotion of secure software development assimilation. *Chinese Management Studies,* 12(1):164–183.

[48] [Stålhane and Johnsen, 2017] Stålhane, T. and Johnsen, S. O. (2017). Resilience and safety in agile development. In *27th European Safety and Reliability Conference*, pages 945–954.

[49] [Tomanek and Klima, 2015] Tomanek, M. and Klima, T. (2015). Penetration Testing in Agile Software Development Projects. *International Journal on Cryptography and Information Security,* 5(1):01–07.

[50] [Tøndel et al., 2019] Tøndel, I. A., Cruzes, D. S., Jaatun, M. G., and Rindell, K. (2019). The Security Intention Meeting Series as a way to increase visibility of software security decisions in agile development projects. In *International Conference on Availability, Reliability and Security*, pages 1–8, Canterbury. ACM Press.

[51] [Tøndel and Jaatun, 2020] Tøndel, I. A. and Jaatun, M. G. (2020). Towards a Conceptual Framework for Security Requirements Work in Agile Software Development. *International Journal of Systems and Software Security and Protection*, 11(1):33–62.

[52] [Türpe and Poller, 2017] Türpe, S. and Poller, A. (2017). Managing security work in scrum: Tensions andchallenges. *CEUR Workshop Proceedings*, 1977(SecSE):34–49.

[53] [Villamizar et al., 2018] Villamizar, H., Kalinowski, M., Viana, M., and Fernández, D. M. (2018). A systematic mapping study on security in agile requirements engineering. In 44th *Euromicro Conference on Software Engineering and Advanced* Applications, pages 454–461.

[54] [Williams and Meneely, 2010] Williams, L. and Meneely, A. (2010). Protection poker: The new software security »game«. *IEEE Security & Privacy*, 8(3):pp. 14–20.

[55] [Yu and Le, 2012] Yu, W. D. and Le, K. (2012). Towards a secure software development lifecycle with SQUARE+R. In *International Computer Software and Applications Conference*, pages 565–570.

▪

Anže Mihelič je doktorski kandidat na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko ter Pravni fakulteti. Kot asistent je zaposlen na Univerzi v Mariboru, Fakulteti za varnostne vede, kot raziskovalec pa na Univerzi v Hagnu, Fakulteti za matematiko in računalništvo. Za svoje raziskovalno delo je prejel nagrado Fakultete za varnostne vede in rektorjevo nagrado Univerze v Mariboru. Bil je predsednik organizacijskega odbora mednarodne konference Central European Cybersecurity Conference 2019, ki se je odvijala v Münchnu. Prav tako je sodeloval in sodeluje pri nacionalnih in mednarodnih projektih s področja informacijske in kibernetske varnosti. Njegovi raziskovalni interesi obsegajo tehnične, zasebnostne, pravne ter psihološke vidike informacijske in kibernetske varnosti.

▪

Simon Vrhovec je docent na Fakulteti za varnostne vede Univerze v Mariboru. Leta 2015 je doktoriral na Fa- kulteti za računalništvo in informatiko Univerze v Ljubljani. V letih 2018 in 2019 je sopredsedoval mednarodni konferenci Central European Cybersecurity Conference (CECC),od leta 2019 pa je član usmerjevalnega od- bora European Interdisciplinary Cybersecurity Conference (EICC). Je član uredniškega odbora znanstvenih revij Journal of Cyber Security and Mobility, International Journal of Cyber Forensics and Advanced Threat Investigations in EUREKA: Social and Humanities. Bil je oz. je gostujoči urednik v znanstvenih revijah IEEE Security Privacy, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications in Journal of Universal Computer Science. Njegova glavna raziskovalna področja so človeški dejavniki v kibernetski varnosti, razvoj varne programske opreme, agilne metode, odpor do sprememb in zdravstvena informatika.

▪

Tomaž Hovelja je doktoriral iz organizacije in managementa na Ekonomski fakulteti Univerze v Ljubljani. Zaposlen je kot izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Njegova raziskovalna področja so družbeni, gospodarski in organizacijski dejavniki uvajanja IT v podjetja in uspešnost IT projektov. Objavlja v revijah, kot so Business information systems engineering, International journal of project management, International journal of engineering education, Assessment evaluation in higher education