

# Topic 3 Als werkende software belangrijker is dan uitgebreide documentatie van die software, wordt deze dan minder veilig en betrouwbaar?.

## Wat is de rol van documentatie in een softwareproduct?

Albert Jan Geertjes  
834bee1ffdb

**Abstract**—Binnen software ontwikkelingsprocessen waarbij gebruik wordt gemaakt van meerdere iteratiestappen ("agile software processes") acht men werkende software belangrijker dan uitgebreide documentatie. Het lijkt alsof de documentatie van de software onbelangrijk gevonden wordt. Dit wordt nog versterkt door moderne hulpmiddelen die het mogelijk maken om steeds kortere en frequentere iteratiestappen uit te voeren. Agile software-ontwikkeling overheerst, waarbij Scrum de boventoon voert. Deze ontwikkeling zou de betrouwbaarheid en veiligheid van de software negatief beïnvloeden. Dit heeft vooral te maken dat men binnen deze ontwikkelmethoden minder aandacht schenkt aan uitgebreide documentatie. Onderzocht nu moet worden in welke mate software documentatie de veiligheid (privacy) en betrouwbaarheid positief, negatief of nauwelijks van invloed is. Daarbij dienen vragenlijsten over kwaliteit van software voorgelegd aan zowel software-bouwers als -gebruikers gehanteerd te worden. Om de veiligheid/ privacy te toetsen zal ook gebruik gemaakt moeten worden van ondervraging van in beveiliging gespecialiseerde bedrijven. Daarnaast is het ook van belang te toetsen of het aantal iteratiestappen van invloed is op de kwaliteit van de software-documentatie.

### I. INTRODUCTIE

Bij software ontwikkelprojecten waarbij een waterval methodologie wordt gehanteerd, wordt uitgebreide documentatie uiterst belangrijk gevonden. Bij een methodologie waarbij de software ontwikkeld wordt met behulp van iteratiestappen gaat men uit van de principes verwoord in het Agile Manifesto[1]. Betrouwbaarheid is altijd een probleem geweest bij het ontwikkelen van software. Het feit dat er tegenwoordig minder aandacht besteed wordt aan documentatie heeft dat niet wezenlijk veranderd. Een legitieme vraag is of het probleem van betrouwbaarheid hiermee niet is toegenomen. Sinds de enorme opkomst van het gebruik van internet dienen veiligheid en bescherming van privacy steeds grotere punten van aandacht te worden als het om ontwikkeling van computersoftware gaat. Overheden zullen daar in de toekomst met wetgeving steeds meer een belangrijke rol in gaan spelen. In hun artikel **Software process. In Proceedings of the on Future of Software Engineering** uit 2014 spreken Fuggetta en Di Nitto hun zorg uit over deze ontwikkelingen en roepen de academische wereld op hier gedegen onderzoek naar te doen[2]. Het is dan echter wel van belang van er goede criteria komen aangaande zowel beveiliging, privacybescherming als kwaliteit van software. Deze definities moeten uit goed, gedegen onderzoek komen.

### II. IS AGILE ONTWIKKELDE SOFTWARE ONBETROUWBARE SOFTWARE?

Bij software, ontwikkeld met behulp van een waterval-methode, wordt er bijzonder veel aandacht besteed aan uitgebreide begeleidende documentatie. Dat moet ook wel, want door de afwezigheid van iteratiestappen moet de communicatie tussen gebruiker en bouwer van de software goed vastgelegd worden. Naast beschrijving van wat de software doet, wordt deze documentatie veelal ook beschouwd als een contract tussen gebruiker en bouwer. Bij software dat ontwikkeld is door een agile (cyclische) methode zijn deze stappen er wel. Bij toepassing van een dergelijke methode meent men dat er (veel) minder aandacht geschonken hoeft te worden aan uitgebreide documentatie. Immers, niet documentatie maar werkzame software is het primaire doel. Maar is dit laatste soort software minder betrouwbaar dan de eerste? Paul E. McMahon schrijft in zijn artikel **Bridging Agile and Traditional Development Methods: A Project Management Perspective**, dat het een misverstand is te denken dat software ontwikkelaars die agile methodes volgen, weinig of geen waarde hechten aan documentatie [3]. Sutherland en Schwaber, beschrijven in **The scrum guide** welke belangrijk rol adequate documentatie in softwareontwikkeling speelt[4]. Scrum en hybride vormen ervan, waren in 2012 al veruit de belangrijkste ontwikkelingsmethoden. Volgens een onderzoek, uitgevoerd door VersionOne [5] onder ruim 4000 participanten was dat 72% Onderzoek moet uitwijzen welk verband er bestaat tussen betrouwbare software en documentatie (kwalitatief en kwantitatief). Het onderzoek dat beschreven wordt door Verner en Evance in het artikel **In-house software development: what project management practices lead to success?** kan als onderzoeksmethode prima als uitgangspunt dienen. Zij hebben onderzocht welke aspecten van invloed zijn op een falend of succesvol software ontwikkelingsproject [6]. Zij maakten gebruik van een vragenlijst die ze software-professionals voorlegden. De onderzoeksresultaten zijn echter in dit kader weinig bruikbaar en wel om de volgende redenen:

- 1) Het onderzoek is 15 jaar oud en daardoor hanteerde geen van de projecten uit dit onderzoek een agile methodiek. Om bovenstaande onderzoeksvraag te kunnen beantwoorden moeten ook dit soort projecten onderzocht worden.
- 2) De software-professionals gaven hun mening over de

kwaliteit van hun project; afgezien dat kwaliteit een subjectief begrip is, moeten ook de gebruikers van die software in het onderzoek betrokken worden.

- 3) Wil men aan zo 'n onderzoek enige statistische relevantie toekennen, moeten de onderzochte projecten zoveel mogelijk onderling onafhankelijk zijn. Deze onafhankelijkheid komt onder druk als een betrokken respondent meerdere projecten 'inbrengt'. Bij het onderzoek, beschreven door Verner en Evance is dat lang niet altijd het geval.
- 4) De projecten in het onderzoek zijn statisch, ze veranderen niet mee met een veranderende omgeving. In het nieuwe onderzoek dient ook onderzocht te worden of bij elke iteratiestap de software ook daadwerkelijk verbeterd.

### III. LEVEREN AGILE METHODOLOGIEËN ONVEILIGERE SOFTWARE OP? EN HOE IS HET MET DE PRIVACY GESTELD?

Het is het beste als het onderzoeken van de veiligheids- en privacy-aspecten getoetst worden door professionals/organisaties, gespecialiseerd in software-beveiliging. Hoewel waarschijnlijk lastig uit te voeren, zou het het mooiste zijn (een deel van) de projecten die onderzocht worden in de vorige paragraaf door deze experts onderzoeken. Ook lastig is objectieve beveiligingscriteria vast te stellen. Idealiter zou een beoordelingssysteem zoals die gehanteerd wordt in de automobiellindustrie. Daar hanteert men een sterrensysteem die heet veiligheidsniveau van een model auto aangeeft. Eén ster is onveilig; vijf sterren de meest veilige. Introductie van een zodanig systeem zou de kwaliteitsbeoordeling van computersoftware enorm vooruit helpen. Ook overheden behept met het maken van privacywetten zouden hieraan meer houvast krijgen.

### IV. AANTAL ITERATIESTAPPEN

Borjesson en Mathiassen schrijven in hun artikel **Successful process implementation (SPI)** over softwareontwikkeling bij Ericsson, dat was in 2004 een grote softwareproducent [7]. Eén van de conclusies uit dat artikel was, dat de kwaliteit van de software recht evenredig was met het aantal iteratiestappen (althans meestal). Aangezien in hun onderzoek het aantal stappen beperkt was, kwamen ze niet verder dan 4 stappen. Boven dit aantal treedt geen noemenswaardige verbetering op. Opgemerkt dient echter te worden dat auteurs slechts drie kwaliteitsniveaus hanteren (Low, Medium en High). Omdat er maar van een beperkt aantal iteratiestappen (cycli) sprake was (in het gunstigste geval 4), er niet gemeten werd of de opgeleverde software ook kon 'meegroeien' met de toekomstige veranderingen, waren meerdere kwaliteitsniveaus waarschijnlijk ook weinig relevant. Daarnaast wordt in het artikel gesuggereerd dat het iteratieproces alleen verbetering van de functionaliteit oplevert. Van iteraties naar verbeterde documentatie en performance wordt niet gesproken. Met de huidige technologische stand van zaken, is softwareontwikkeling mogelijk met veel meer iteratiestappen. Levert het toepassen van meer

stappen ook betere software op, ook in de zin van documentatie en performance? Onderzoek daarnaar geeft in ieder geval het antwoord op de vraag of agile methodes werkelijk slecht gedocumenteerde software opleveren en derhalve ook slechtere software dan die ontwikkeld met behulp van de watervalmethode.

### REFERENCES

- [1] A. Cockburn, "Agile software development," pp. 215–218, 2002.
- [2] A. Fuggetta and E. Di Nitto, "Software process," in *Proceedings of the on Future of Software Engineering*. ACM, 2014, pp. 1–12.
- [3] P. McMahon, "Bridging agile and traditional development methods: A project management perspective," May 2004.
- [4] J. Sutherland and K. Schwaber, <https://www.scrumguides.org/scrum-guide.html>, Scrum.org, 2017.
- [5] VersionOne, <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>.
- [6] J. Verner and W. Evancho, "In-house software development: what project management practices lead to success?" pp. 86–93, 2005.
- [7] A. Borjesson and L. Mathiassen, pp. 36–44, 2004.