

**WACHTEN OP  
ONTWIKKELAARS  
GROTERE ERGERNIS  
DAN TECHNISCHE  
FOUTEN**

# Social Software Engineering

# KOESTER DE ONTWIKKELAAR

**SOFTWARE ENGINEERING GAAT TEGENWOORDIG EVENVEEL OM MENSEN ALS OM TECHNIEK. ER ZIJN VEEL VERSCHILLENDE INVALSHOEKEN OM DE MENSELIJKE ASPECTEN VAN SOFTWARE ENGINEERING TE BESTUDEREN. NATHAN CASSEE EN ALEXANDER SEREBRENİK ZIEN DAARBIJ DAT DIVERSITEIT EN REKENING HOUDEN MET MENSELIJKE ASPECTEN BELANGRIJK ZIJN OM KWALITEIT VAN CODE HOOG TE HOUDEN.**

door Nathan Cassee en Alexander Serebrenik beeld Shutterstock



SOFTWARE WORDT (MEESTAL) ONTWIKKELD DOOR MENSEN en, logischerwijs worden zowel het ontwikkelingsproces als de software zelf beïnvloed door ontwikkelaars en hoe ze met elkaar omgaan. ‘Social software engineering’ bestudeert deze kwesties met het oog op de vereenvoudiging van ontwikkeling en onderhoud van softwaresystemen.

Hoewel het inzicht in het belang van menselijke aspecten in software engineering teruggaat tot minstens de jaren ‘70, toen ‘The Mythical Man-Month’ van Fred Brooks werd gepubliceerd, hebben verschillende recente belangrijke veranderingen dit onderwerp in de schijnwerpers gezet. Ten eerste is het proces van softwareontwikkeling veranderd, omdat meer en meer software in samenwerking met mensen van verschillende culturen wordt ontwikkeld. Ten tweede is ook de maatschappij zelf veranderd, met veel

meer aandacht voor onderwerpen zoals geestelijke gezondheid en welzijn, gelijkheid en discriminatie, diversiteit en inclusie.

## EEN GOEDE DAG

Recente studies benadrukken dat het welzijn van de ontwikkelaars en de onderlinge communicatie een wezenlijk onderdeel zijn van de productiviteit (Forsgren et al., 2021), dat gelukkige ontwikkelaars beter presteren (Graziotin et al., 2013) en dat meer genderdiverse teams productiever zijn (Vasilescu et al., 2015) dan minder genderdiverse. Bedrijven zoals Microsoft investeren veel in uitgebreide studies om te begrijpen wat ‘een goede dag’ voor een ontwikkelaar betekent en hoe men meer dagen goed kan maken (Meyer et al., 2021). Daarnaast is Microsoft bezig met meerdere programma’s rondom diversiteit en inclusie.

Samen met collega’s uit het binnen- en buitenland verrichten we onderzoek naar het raakvlak tussen sociale aspecten zoals communicatie en technische aspecten. Zo hebben we samen met Bogdan Vasilescu (Carnegie Mellon University, VS) bestudeerd hoe de introductie van Continuous Integration (CI) de samenwerking beïnvloedt tussen de ontwikkelaars. Door 685 open-source-projecten te bestuderen hebben we kunnen vaststellen dat na de introductie van CI softwareontwikkelaars even veel werk doen als daarvoor, maar dat ze hiervoor minder hoeven te communiceren. Voor de introductie van CI moest men gemiddeld 4.5 berichten plaatsen per voorgestelde codewijziging; een jaar na de introductie kost het gemiddeld één berichtje minder om dezelfde wijziging te reviewen. Op deze wijze kan men CI niet louter en alleen als een technische oplossing zien maar ook als een stille

## AUTEUR



ALEXANDER SEREBRENİK  
is hoogleraar Social Software Engineering aan de TU Eindhoven

helper die ontwikkelaars ondersteunt en ze werk uit handen neemt.

## ZWARTE WOLK

Daarnaast is het verminderen van nutteloze communicatie een manier om duidelijkheid en structuur in te brengen in de communicatie. Gebrek aan een dergelijke structuur in combinatie met overdaad aan communicatie is een kenmerk van de zogeheten 'zwarte wolk', een anti-patroon in communicatie waarbij de betrokken partijen geen onderscheid kunnen maken tussen nuttige en nutteloze communicatie. 'Zwarte wolk' is een voorbeeld van een ongewenste communicatie tussen softwareontwikkelaars die negatieve gevolgen heeft voor de kwaliteit van software. Door een grootschalige analyse van opensourceprojecten te combineren met een survey van projectmanagers hebben we aangetoond dat dergelijke ongewenste communicatiepatronen kunnen leiden tot het ontstaan van codegeuren. Zo kan de 'zwarte wolk' leiden tot te lange methodes in de code en 'eenzame wolf', waar-

bij teamleden die hun werk doen zonder rekening te houden met de rest van het team. Dit kan leiden tot spaghetti-code (Palomba et al., 2021).

## GENDERDIVERSITEIT

Gezien de invloed van dergelijke ongewenste communicatiepatronen op softwarekwaliteit hebben we ook onderzoek gedaan naar aanwezigheid van dergelijke patronen in verschillende teams van softwareontwikkelaars. Daaruit bleek dat de aanwezigheid van vrouwen in de teams over het algemeen de hoeveelheid ongewenste communicatiepatronen vermindert, en daardoor dus bijdraagt aan betere software. We hebben ook gezien dat bij de 'zwarte wolk' genderdiversiteit belangrijk was, terwijl voor andere patronen het eerder ging over de aanwezigheid van vrouwen (Catolino et al., 2019). Met andere woorden: zelfs wanneer vrouwen in de minderheid zijn, kunnen zij als bemiddelaars optreden tegen de verspreiding van specifieke communicatieproblemen. Dit suggereert dat bedrijven en opensourceprojecten

## AUTEUR



NATHAN CASSEE  
is promovendus TA, Mathematics and Computer Science, Social Software Engineering aan de TU Eindhoven



een belang hebben bij het waarborgen van genderdiversiteit binnen softwareontwikkelteams.

In andere studies hebben we ons samen met collega's uit Italië gericht op het fenomeen 'technical debt'. De term is bedacht door Ward Cunningham, een van de schrijvers van het Agile Manifesto. Technical debt is een metafoor waarbij problemen in broncode of het ontwerp gelijk worden gesteld aan financiële schulden. Door geld te lenen kan men meer doen of dingen eerder doen, maar totdat alles terugbetaald is, behoort men ook een rente te betalen. Op een vergelijkbare manier kan men sneller releasen of meer functionaliteit aanbieden door imperfecte technische beslissingen te nemen. Deze beslissingen horen echter opnieuw bekeken te worden en de bijbehorende broncode zal ooit gerefactord moeten worden.

## SELF ADMITTED TECHNICAL DEBT

In ons werk richten we ons op zogenaamde 'self-admitted technical debt' (SATD), situaties waarbij de ontwikkelaars door middel van commentaar aangeven (of toegeven) dat de broncode niet helemaal goed is, bijv. "EATM This might be better written as a single loop for the EObject case" (uit het project EMF). SATD komt voor zowel bij opensourceprojecten als bij closed-sourceprojecten uit de industrie. Uit een recente enquête blijkt dat 53% van de opensource ontwikkelaars en 29% van hun industriële collega's dergelijke commentaren plaatsen bij minstens een kwart van hun taken. Het verschil tussen de opensource ontwikkelaars en de industriële suggereert dat de laatste minder geneigd zijn TD toe te geven, hetzij wegens organisatorische richtlijnen, hetzij wegens een cultuur waar onderprestaties liefst verborgen worden -iets wat contraproductief kan zijn (Zampetti et al., 2021).

Deze enquête heeft ons alleen weinig inzichten gegeven in welke situaties de ontwikkelaars als SATD annoteren.

## Met vrouwen in een team vermindert overbodige communicatie

Daarom hebben we een steekproef genomen van 1038 stukken SATD commentaar afkomstig uit vier opensourceprojecten (Fucci et al., 2021). We categoriseerden de commentaren in negen categorieën, waarbij de slechte implementatiekeuzes de meest omvangrijke categorie blijkt te zijn, met meer dan 40% van SATD-commentaren. De op een na grootste en op twee na grootste categorieën zijn gedeeltelijk geïmplementeerde functionaliteit (22%) en functionele fouten (13%). Bij de overige categorieën zien we ook het wachten op de functionaliteit die nog geïmplementeerd moet worden al dan niet binnen het project zelf, problemen in documentatie of testen.

De meest voorkomende categorieën zijn evenwel niet noodzakelijk degene die softwareontwikkelaars zelf als het belangrijkste beschouwen, of waaraan ze zich het meest ergeren. Daarom hebben we de commentaren nogmaals geanalyseerd, nu vanuit een perspectief van menselijke emotie. Ieder commentaar is door de onderzoekers als "negatief" of als "niet-negatief" bestempeld. Bijvoorbeeld "this method is a nightmare" suggereert een negatieve houding van de ontwikkelaar ten opzichte van de broncode, terwijl "TODO: Why is this a special case?" neutraal is. Onze studie toont aan dat SATD-commentaren over functionele fouten vaak een negatief sentiment uitdrukken. Ook lijkt wachten op

ontwikkelaars meer te storen dan andere categorieën technical debt. Ontwikkelaars zijn daarentegen neutraler bij het rapporteren van slechte implementatiekeuzes of problemen met documentatie en testen. Dit suggereert dat naast de functionele fouten, je aandacht moet schenken aan de situaties waar de ontwikkelaars moeten wachten en zich dus niet productief voelen. 🧠

## Referenties

- Gemma Catolino, Fabio Palomba, Damian A. Tamburri, Alexander Serebrenik, Filomena Ferrucci: *Gender diversity and women in software teams: how do they affect community smells?* ICSE-SEIS 2019: 11-20
- Nicole Forsgren, Margaret-Anne D. Storey, Chandra Shekhar Maddila, Thomas Zimmermann, Brian Houck, Jenna L. Butler: *The SPACE of developer productivity.* Commun. ACM 64(6): 46-53 (2021)
- Gianmarco Fucci, Nathan Cassee, Fiorella Zampetti, Nicole Novielli, Alexander Serebrenik, Massimiliano Di Penta: *Waiting around or job half-done? Sentiment in self-admitted technical debt.* MSR 2021: 403-414
- Daniel Graziotin, Xiaofeng Wang, Pekka Abrahamsson: *Are Happy Developers More Productive? - The Correlation of Affective States of Software Developers and Their Self-assessed Productivity.* PROFES 2013: 50-64
- André N. Meyer, Earl T. Barr, Christian Bird, Thomas Zimmermann: *Today Was a Good Day: The Daily Life of Software Developers.* IEEE Trans. Software Eng. 47(5): 863-880 (2021)
- Fabio Palomba, Damian Andrew Tamburri, Francesca Arcelli Fontana, Rocco Oliveto, Andy Zaidman, Alexander Serebrenik: *Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells?* IEEE Trans. Software Eng. 47(1): 108-129 (2021)
- Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark G. J. van den Brand, Alexander Serebrenik, Premkumar T. Devanbu, Vladimir Filkov: *Gender and Tenure Diversity in GitHub Teams.* CHI 2015: 3789-3798
- Fiorella Zampetti, Gianmarco Fucci, Alexander Serebrenik and Massimiliano Di Penta. *Self-Admitted Technical Debt Practices: A Comparison Between Industry and Open-Source, Empirical Software Engineering, 2021*