Publicatie : AG Connect
Datum : 31 dec 2021
cm2 : 1350
Pagina : 66
AVE : € 12250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 7700

Software-ecosysteem net zo kwetsbaar als evenknie in de natuur

THE GOOD, THE BAD AND THE UGLY

EEN SOFTWARE-ECOSYSTEEM IS EEN GROTE VERZAMELING VAN ELKAAR AFHANGENDE SOFTWAREPROJECTEN, GEZIEN IN EEN BREDE SOCIO-TECHNOLOGISCHE CONTEXT. IN DE OPENSOURCEWERELD LEVEN WERKELIJK GIGANTISCHE SOFTWARE-ECOSYSTEMEN. JURGEN VINJU EN TIJS VAN DER STORM ZIEN DE NADELEN EN DE RISICO'S MAAR ZEKER OOK DE ENORME VOORDELEN VAN SOFTWARE-ECOSYSTEMEN VOOR ONTWIKKELAARS EN ONDERZOEKERS.

door Jurgen Vinju en Tijs van der Storm beeld Shutterstock

Dit artikel is uitsluitend voor intern gebruik binnen: Centrum Wiskunde & Informatica

Publicatie : AG Connect
Datum : 31 dec 2021
cm2 : 1350
Pagina : 66
AVE : € 12250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 7700

DE MEESTE ECOSYSTEMEN LATEN ZICH DEFINIËREN DOOR (a) een specifieke programmeertaal (zoals Java of Python) of een run-time omgeving (zoals de JVM), (b) een groot aantal online broncode repositories (zoals Github of Gitlab), (c) een centrale database waarin alle uitgaven (releases) te vinden zijn van applicaties en bibliotheken bedoeld voor hergebruik (zoals NPM of Maven Central), en (d) alles daar omheen: mailing lijsten, issue trackers, documentatiebestanden, gebruikersfora, Slack-kanalen, ga zo maar door.
Dit soort ecosystemen leven voornamelijk in de wereld van open source: de broncode en de uitgeleverde applicaties en bibliotheken zijn vaak vrij beschikbaar, en de ontwikkelaars zijn openlijk te benaderen voor vragen, bug reports, of verzoeken ter verbetering. Ook is het vaak mogelijk bij te dragen aan de software zelf "van buiten af" (een zogenaamde "pull request").

Er zijn ook gesloten software-ecosystemen, zoals die van Google, die zich kenmerken door een werkelijk gigantisch versiebeheersysteem waarin alle software die Google-medewerkers schrijven, wordt opgeslagen. Ook Facebook, Apple en Microsoft hebben dergelijke ecosystemen, die de stijl en manier van werken van hun softwareontwikkeling bepalen. Maar ook grote softwaregedreven organisaties zoals banken hebben privé software-ecosystemen, met hun eigen bibliotheken, programmeertalen (of dialecten van bestaande talen), processen, rapportagesystemen, en programmeeridiomen en -conventies. Maar wat zijn de voor- en nadelen aan software-ecosystemen, hun risico's en bijbehorende kansen?

THE GOOD:

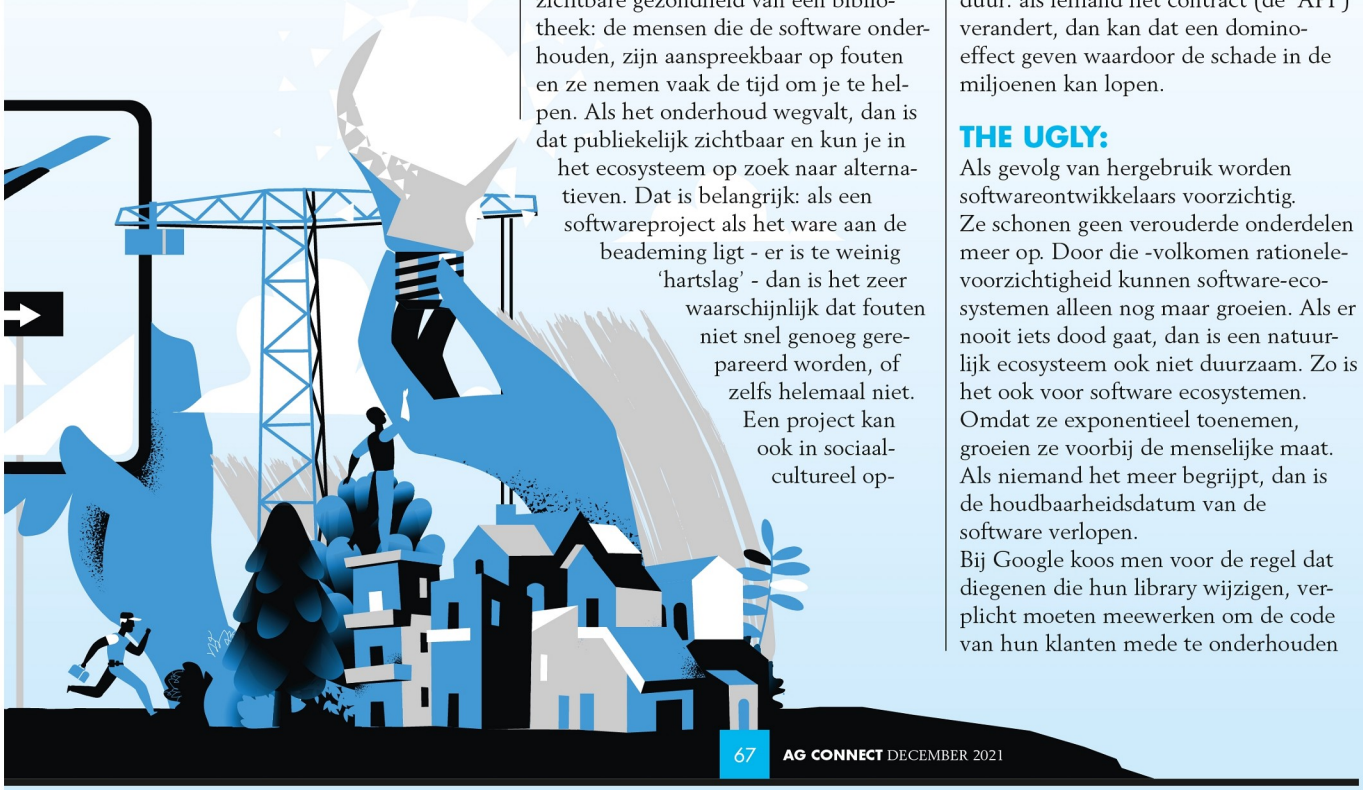
Software ecosystemen zijn de geïncarneerde belofte van modulaire softwareontwikkeling uit de jaren '90. Ecosystemen maken herbruikbare software vindbaar en snel te installeren.
Er zijn positieve sociale aspecten aan de zichtbare gezondheid van een bibliotheek: de mensen die de software onderhouden, zijn aanspreekbaar op fouten en ze nemen vaak de tijd om je te helpen. Als het onderhoud wegvalt, dan is dat publiekelijk zichtbaar en kun je in het ecosysteem op zoek naar alternatieven. Dat is belangrijk: als een softwareproject als het ware aan de beademing ligt - er is te weinig 'hartslag' - dan is het zeer waarschijnlijk dat fouten niet snel genoeg gerepareerd worden, of zelfs helemaal niet. Een project kan ook in sociaal-cultureel opzicht vooroplopen; door bijvoorbeeld een gebruiksvriendelijke licentie te bieden, of door een vriendelijke sfeer op de mailinglist. Kortom, het publieke gezicht van software-ecosystemen draagt bij aan het maken van verstandige keuzes binnen softwareprojecten.

THE BAD:

Software-ecosystemen zijn kwetsbaar. Een geval waarin de analogie met echte ecosystemen wél door te trekken is. Vooral op 'de bodem', waar de vaakst gebruikte bibliotheken leven, kan maar beter niks misgaan. Een zero-day veiligheidsrisico op zo'n plek kan het hele ecosysteem besmetten, en dan hebben we het over honderdduizenden projecten met minstens zoveel software engineers. Met het zo nagestreefde hergebruik komt onvermijdelijk de prijs van afhankelijkheid. Denk aan hoe in 2016 de left-pad module in het Javascript-ecosysteem het halve internet op z'n knieën kreeg, toen het uit de NPM-module database werd verwijderd. Andere wijzigingen zijn mogelijk net zo duur: als iemand het contract (de 'API') verandert, dan kan dat een domino-effect geven waardoor de schade in de miljoenen kan lopen.

THE UGLY:

Als gevolg van hergebruik worden softwareontwikkelaars voorzichtig. Ze schonen geen verouderde onderdelen meer op. Door die -volkomen rationele- voorzichtigheid kunnen software-ecosystemen alleen nog maar groeien. Als er nooit iets dood gaat, dan is een natuurlijk ecosysteem ook niet duurzaam. Zo is het ook voor software ecosystemen. Omdat ze exponentieel toenemen, groeien ze voorbij de menselijke maat. Als niemand het meer begrijpt, dan is de houdbaarheidsdatum van de software verlopen.
Bij Google koos men voor de regel dat diegenen die hun library wijzigen, verplicht moeten meewerken om de code van hun klanten mede te onderhouden



Dit artikel is uitsluitend voor intern gebruik binnen: Centrum Wiskunde & Informatica

Publicatie : AG Connect
Datum : 31 dec 2021
cm2 : 1350
Pagina : 66
AVE : € 12250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 7700

AUTEUR

TIJS VAN DER STORM
is senior onderzoeker en groepsleider bij het Centrum Wiskunde & Informatica (CWI) en deeltijd hoogleraar software engineering aan de Rijksuniversiteit Groningen.

AUTEUR

JURGEN VINJU
is senior onderzoeker bij het Centrum Wiskunde & Informatica (CWI) en deeltijdhoogleraar automated software analysis aan de TU Eindhoven.
Jurgen.vinju@cwi.nl
Foto: CWI

(automatisch). Een ander voorbeeld is het R-ecosysteem. R is een programmeertaal voor statistische berekeningen met een groot ecosysteem: CRAN. Als de ontwikkelaars van R een nieuw onderdeel uitbrengen of een nieuwe versie van de taal, dan moeten de bibliotheken evolueren; anders worden ze verwijderd uit het ecosysteem. Zo wordt CRAN gezond gehouden. Het vertoont geen exponentiële groei meer, want het kan niet groeien voorbij het minimale aantal mensen die de bibliotheken onderhouden.

BOOST VOOR EMPIRISCH ONDERZOEK

Binnen de academische gemeenschap van software-engineering wordt onderzoek gedaan naar het gedrag van software-ecosystemen, en naar hoe (semi-) automatische gereedschappen kunnen helpen zo'n ecosysteem te navigeren en gezond te houden. Het bestaan van software-ecosystemen heeft een boost gegeven aan de empirische methoden. We kunnen nu niet alleen broncode observeren, zoals voorheen, maar ook hoe die zich ontwikkelt en wat activiteiten van software engineers zijn. Verschillende vakgroepen in Nederland lopen wereldwijd voorop met empirisch onderzoek. Het ecosysteem is 'big data' en de onderzoeksvragen in software engineering: het wat, hoe en waarom van (on) betrouwbare, (on)betaalbare en (niet-) duurzame software, zijn deels te beantwoorden door die data te analyseren.
Een praktische vraag is: hoe besluit ik een bibliotheek te gaan gebruiken?
Er zijn veel factoren die bijdragen aan deze beslissing, zowel technisch, als sociaal: wat is de kwaliteit in termen van functionaliteit? Wat zijn de performancegaranties? Hoe actief is het project? Hoe zit het met de codekwaliteit? Staat het project open voor contributies? Hoe vriendelijk is de cultuur?
In recent onderzoek is uitgezocht hoe een automatisch aanbevelingssysteem ontwikkelaars kan ondersteunen bij dit soort keuzes. Op basis van analyse van broncode, versiegeschiedenis en andere informatie in het Java-ecosysteem kan dan een dashboard getoond worden voor een afgewogen keuzeproces.

NIEUW GEREEDSCHAP

De problemen van software ecosystemen zijn ook aanleiding tot het ontwikkelen van nieuw gereedschap. De gegevens die geleerd kunnen worden uit de analyse van een heel software-ecosysteem kunnen naar degene gebracht worden die een bibliotheek onderhoudt, of naar de gebruiker van die bibliotheek. Een belangrijk aspect van hergebruik is

Platforms als Github zijn vliegwiel voor interactie

bijvoorbeeld compatibiliteit: 'past' een nieuwe versie van een bibliotheek nog? Stel je bent afhankelijk van een bibliotheek X versie 1.0, en op een gegeven moment komt versie 2.0 uit: moet je nu je code aanpassen zodat je mee kunt gaan naar de nieuwe versie? Stel je kiest hiervoor, wat is dan de schade? Hoeveel moet er worden aangepast en hoeveel werk is dat? Zijn er alternatieve oplossingen? Dit zijn allemaal belangrijke vragen, want je wil niet dat je softwareproject achteropraakt. Hoe langer je het upgraden uitstelt, hoe groter de kans op niet-gedichte veiligheidslekken, hoe meer nieuwe features je mist, en hoe hoger de kosten worden om uiteindelijk toch mee te liften.

VLIEGWIEL

Het is duidelijk dat software-ecosystemen niet meer weg te denken zijn uit de huidige software-ontwikkelingspraktijk. De open-source-ecosystemen dragen bij aan hergebruik van bibliotheken op grote schaal en platforms als Github en Stackoverflow zijn een vliegwiel voor openheid, transparantie en interactie. Niettemin zijn ecosystemen kwetsbaar: afhankelijkheid brengt risico met zich mee, en de gezondheid van een ecosysteem is niet gegarandeerd.
Onderzoek naar software-ecosystemen heeft een vlucht genomen. Door alle bronnen van informatie in een ecosysteem te analyseren, kunnen ontwikkelaars betere beslissingen maken en geholpen worden met onderhoud.
De metafoor van ecosystemen blijkt voor software uitstekend te passen: ze zijn waardevol, complex, dynamisch, en kwetsbaar. Net als in de natuur.

Dit artikel is uitsluitend voor intern gebruik binnen: Centrum Wiskunde & Informatica