

OPEN SOURCE-SOFTWARE PRODUCEREN

Met succes een open source-softwareproject runnen

Karl Fogel

*Dit boek is opgedragen aan twee dierbare vrienden
zonder wie het niet tot stand zou zijn gekomen:
Karen Underhill en Jim Blandy.*

Colofon

Deze vertaling van het boek Producing Open Source Software is mogelijk gemaakt door Stichting Kennisnet, SURFfoundation, SURFdiensten en SURFnet.



Er zijn geen inhoudelijke wijzigingen in de tekst van de auteur aangebracht, met uitzondering van het onderdeel 'vrij' versus 'open' in Hoofdstuk 1. De tekst is hier ingekort, omdat het beschreven probleem in het Engels wel, maar in het Nederlands niet van toepassing is. Dit is gedaan in overleg met en met toestemming van de auteur.

Oorspronkelijke titel

Producing Open Source Software

How to Run a Successful Free Software Project

Auteur Karl Fogel

Vertaling Tekom Vertalers BV.

Ontwerp en opmaak Vrije Stijl grafisch ontwerp

Beeld MegapixelMedia, Istockphoto.com

Druk Drukkerij Libertas



Copyright © 2005, 2006, 2007 Karl Fogel, onder een licentie van Creative Commons Attribution-ShareAlike (3.0)

INHOUDSOPGAVE

VOORWOORD 9

EEN WOORD VOORAF 10

Waarom schrijf ik dit boek?
Wie zou dit boek moeten lezen?
Bronnen
Dankbetuiging
Disclaimer

1. INLEIDING 17

- 1.1 Geschiedenis 20**
 - De opkomst van propriëtaire software en open source-software
 - Bewuste weerstand
 - Onbewuste weerstand
 - 'Vrij' versus 'Open source'
- 1.2 De huidige stand van zaken 26**

2. AAN DE GANG 29

- Maar eerst kijkt u rond
- 2.1 Beginnen met wat u hebt 32**
 - Een goede naam kiezen
 - Zorg voor een duidelijke missieverklaring
 - Vermeld dat het project vrij is
 - Lijst met functies en vereisten
 - Ontwikkelingsstatus
 - Alfa en Bèta
 - Downloads
 - Toegang tot versiecontrole en bug tracker
 - Communicatiekanalen
 - Richtlijnen voor ontwikkelaars
 - Documentatie
 - Beschikbaarheid van documentatie
 - Documentatie voor ontwikkelaars
 - Voorbeelden van output en screenshots
 - Screenshots
 - Canned hosting
- 2.2 Een licentie kiezen en toepassen 44**
 - De 'Alles mag'-licenties
 - De GPL
 - Hoe u een licentie toepast op uw software
- 2.3 De toon zetten 46**
 - Voorkom privédiscussies
 - Grofheden in de kiem smoren
 - De code nakijken op verdachte onderdelen
 - Wees bij het openen van een reeds gesloten project alert op de impact van de wijziging
- 2.4 Aankondigen 53**

3. TECHNISCHE INFRASTRUCTUUR 57

| | |
|--|----|
| 3.1 Projectbenodigdheden | 59 |
| 3.2 Mailinglijsten | 60 |
| Spampreventie | |
| Filteren van posts | |
| Adressen in archief verbergen | |
| Management van identificatie en headers | |
| Het grote 'reply-to'-debat | |
| Twee fantasieën | |
| Archivering | |
| Software | |
| 3.3 Versiebeheer | 71 |
| 'Versie' versus 'revisie' | |
| Verklarende woordenlijst versiebeheer | |
| Een versiebeheersysteem kiezen | |
| Het versiebeheersysteem gebruiken | |
| Houd alles onder versiebeheer | |
| Doorzoekbaarheid | |
| E-mails committen | |
| CIA: Nog een manier om veranderingen te publiceren | |
| Branches gebruiken om bottlenecks te voorkomen | |
| Enkelvoudigheid van informatie | |
| Autorisatie | |
| 3.4 Bug tracker | 83 |
| Interactie met mailinglijsten | |
| De bug tracker voorfilteren | |
| 3.5 IRC / Realtime chatsystemen | 89 |
| Paste-sites | |
| Bots | |
| IRC archiveren | |
| 3.6 RSS Feeds | 92 |
| 3.7 Wiki's | 92 |
| 3.8 Website | 94 |
| Canned hosting | |
| Een canned hosting site kiezen | |
| Anonimiteit en betrokkenheid | |

4. SOCIALE EN POLITIEKE INFRASTRUCTUUR 99

| | |
|--|-----|
| Afsplitsbaarheid of forkability | |
| 4.1 Vriendelijke dictators | 101 |
| Wie is geschikt als vriendelijke dictator? | |
| 4.2 Consensusdemocratie | 102 |
| Versiecontrole betekent dat u kunt relaxen | |
| Als geen consensus bereikt wordt, stem dan | |
| Wanneer te stemmen? | |
| Wie mag er stemmen? | |
| Opinie peilen versus stemmen | |
| Veto's | |
| 4.3 Alles opschrijven | 109 |

5. GELD 113

| | |
|---|-----|
| 5.1 Soorten betrokkenheid | 115 |
| 5.2 Aanstelling voor de lange termijn | 117 |
| 5.3 Kom over als individuen, niet als een blok | 118 |
| 5.4 Wees open over uw beweegredenen | 119 |
| 5.5 Met geld kun je geen liefde kopen | 121 |
| 5.6 Aanbesteding | 123 |
| Beoordeling en aanvaarding van wijzigingen | |
| Casestudy: het protocol voor CVS-wachtwoordauthenticatie | |
| 5.7 Financiering van niet-programmeringsactiviteiten | 126 |
| Kwaliteitsgarantie (oftewel professioneel testen) | |
| Juridisch advies en bescherming | |
| Documentatie en bruikbaarheid | |
| Hosting/bandbreedte bieden | |
| 5.8 Marketing | 130 |
| Onthoud dat u in de gaten gehouden wordt | |
| Kraak concurrerende open source-producten niet af | |

6. COMMUNICATIE 135

| | |
|---|-----|
| 6.1 U bent wat u schrijft | 136 |
| Structuur en opmaak | |
| Inhoud | |
| Toon | |
| Onbeleefd gedrag herkennen | |
| Het gezicht | |
| 6.2 De gebruikelijke valkuilen omzeilen | 145 |
| Plaats geen nutteloze posts | |
| Productieve versus niet-productieve threads | |
| Hoe makkelijker het onderwerp, des te langer de discussie | |
| Voorkom heilige oorlogen | |
| Het effect van de 'luidruchtige minderheid' | |
| 6.3 Moeilijke mensen | 152 |
| Omgaan met moeilijke mensen | |
| Casestudy | |
| 6.4 Omgaan met groei | 156 |
| Opvallend gebruik vanarchieven | |
| Behandel alle bronnen alsarchieven | |
| Named Anchors en ID-attributen | |
| Gewoontes voor codificeren | |
| 6.5 Geen discussies in de bug tracker | 164 |
| 6.6 Publiciteit | 166 |
| Aankondigen van beveiligingskwetsbaarheden | |
| Het rapport ontvangen | |
| De fix in stilte ontwikkelen | |
| CAN/CVE-nummers | |
| Vooraankondiging | |
| De fix publiceren | |

7. DOWNLOADPAKKETTEN MAKEN, RELEASES EN DAGELIJKSE ONTWIKKELING 177

| | |
|--|-----|
| 7.1 Releasenummers | 178 |
| Componenten van releasenummers | |
| De simpele methode | |
| De even-/onevenmethode | |
| 7.2 Release-branches | 184 |
| De werking van release-branches | |
| 7.3 Een release stabiliseren | 187 |
| De eigenaar van de release als dictator | |
| Stemmen over veranderingen | |
| Het managen van coöperatieve releasestabilisatie | |
| Releasemanager | |
| 7.4 Het maken van downloadpakketten | 191 |
| Format | |
| TAR-bestanden | |
| Naam en lay-out | |
| Wel of geen hoofdletters | |
| Voorreleases | |
| Compilatie en installatie | |
| Binaire pakketten | |
| 7.5 Testen en uitbrengen | 198 |
| Kandidaat-releases | |
| Releases aankondigen | |
| 7.6 Het onderhouden van verschillende releaselijnen | 200 |
| Beveiligingsreleases | |
| 7.7 Releases en dagelijkse ontwikkeling | 202 |
| Het plannen van releases | |

8. HET MANAGEN VAN VRIJWILLIGERS 207

| | |
|---|-----|
| 8.1 Het meeste uit uw vrijwilligers halen | 208 |
| Delegeren | |
| Onderscheid maken tussen informeren en toewijzen (assign) | |
| Follow-up geven aan delegeren | |
| Zien waar mensen in geïnteresseerd zijn | |
| Complimenten en kritiek | |
| Territoriumdrang voorkomen | |
| De automatiseringsratio | |
| Geautomatiseerd testen | |
| Regressietests | |
| Behandel iedere gebruiker als een mogelijke vrijwilliger | |
| 8.2 Zowel managementtaken als technische taken delen | 221 |
| Patchmanager | |
| Vertaalmanager | |
| Internationalisatie versus lokalisatie | |
| Documentatiemanager | |
| Issuemanager | |
| FAQ-manager | |
| 8.3 Transities | 229 |
| 8.4 Committers | 232 |
| Committers kiezen | |
| Commit access intrekken | |

Gedeeltelijke commit access
Slapende committers
Voorkom geheimzinnigheid

| | |
|----------------------|-----|
| 8.5 Erkenning | 236 |
| 8.6 Forks | 238 |
| Omgaan met een fork | |
| Een fork initiëren | |

9. LICENTIES, AUTEURSRECHTEN EN PATENTEN 245

| | |
|---|-----|
| 9.1 Terminologie | 245 |
| 9.2 Aspecten van licenties | 249 |
| 9.3 De GPL en licentiecompatibiliteit | 250 |
| 9.4 Een licentie kiezen | 251 |
| De MIT / X Window System-licentie | |
| De GNU General Public-licentie | |
| Is de GPL gratis of niet gratis? | |
| En de BSD-licentie? | |
| 9.5 Toekennen van auteursrecht en eigendom | 254 |
| Niets doen | |
| Contributor-licentieovereenkomsten | |
| Overdracht van auteursrecht | |
| 9.6 Tweevoudige licentieprogramma's | 257 |
| 9.7 Patenten | 258 |
| 9.8 Verdere bronnen | 261 |

APPENDICES 265

| | |
|---|-----|
| A. Open source-versiebeheersystemen | 265 |
| B. Gratis bug trackers | 270 |
| C. Waarom zou ik me druk maken over de kleur van het fietsenhok? | 273 |
| D. Voorbeeldinstructies voor het rapporteren van bugs | 278 |
| E. Auteursrecht | 281 |

VOORWOORD



Als je verandering wilt in de wereld, begin dan bij jezelf. Onder dat motto ben ik dit jaar begonnen met de uitvoering van het politiek breed gesteunde Actieplan Nederland Open in Verbinding. Overheden en de publieke sector zijn nu verplicht om open standaarden te gebruiken en worden sterk aangeraden om voor open source software te kiezen.

Het gebruik van open source software stimuleert innovatie, omdat softwareontwikkelaars kunnen voortbouwen op software die eerder is ontwikkeld door concurrenten. Daarnaast maakt het klanten minder afhankelijk van leveranciers, omdat ze de vrijheid hebben om bestaande software uit te breiden en niet vastzitten aan jarenlange contracten. Ten slotte zorgt de beschikking over de broncode ervoor dat de toepassing in de toekomst altijd gebruikt en uitgebreid kan worden, ongeacht of de softwareleverancier nog bestaat.

In de praktijk blijkt echter dat onbekendheid met dit type software potentiële klanten afschrikt. Elk softwareproject kent zijn uitdagingen en die van open source zijn nu eenmaal anders dan die van gesloten software. Karl Fogel vertelt in zijn boek precies wat je moet doen om deze uitdagingen te overwinnen.

Ik ben dan ook blij met het initiatief van SURF en Kennisnet om het boek van deze ervaren softwareontwikkelaar in het Nederlands te vertalen en deze kennis vrij beschikbaar te stellen via internet. De kracht van het boek is, wat mij betreft, vooral dat het zo leesbaar is. Doordat het de voor- en nadelen van open source software-ontwikkeling zo helder uitlegt, is het een goede remedie tegen drempelvrees.

Ik weet daarom zeker dat dit boek u zal aanmoedigen om open source software te maken en te gebruiken.

Frank Heemskerk
Staatssecretaris van Economische Zaken

EEN WOORD VOORAF

WAAROM SCHRIJF IK DIT BOEK?

Op feestjes kijken mensen mij niet langer met een niet-begrijpende blik aan als ik zeg dat ik open source-software schrijf. “Oh, ja, open source, zoals Linux?” zeggen ze. Ik knik geestdriftig ter instemming. “Ja, inderdaad! Dat is wat ik doe.” Het is prettig geen compleet randverschijnsel meer te zijn. In het verleden was de volgende vraag altijd erg voorspelbaar: “Maar hoe verdien je je boterham daarmee?” Ik antwoordde dan met een samenvatting van de logica van open source: dat er organisaties zijn in wiens belang het is dat er bepaalde software bestaat, maar dat het niet noodzakelijk is er exemplaren van te verkopen. Ze willen gewoon zeker weten dat de software beschikbaar is en onderhouden wordt, meer als gereedschap dan als verhandelbaar product.

De laatste tijd gaat de vervolgvraag echter niet altijd over het geld. De business case voor open source software¹ is niet meer zo geheimzinnig als voorheen en veel niet-programmeurs begrijpen inmiddels (of zijn op zijn minst niet meer verbaasd) dat er mensen zijn die hier een fulltime baan aan hebben. In plaats daarvan is de vraag die ik steeds vaker hoor “Oh, en hoe gaat dat in zijn werk?”

Ik had daar geen bevredigend antwoord op en hoe meer ik m'n best deed om een antwoord te vinden, des te meer ik mij realiseerde hoe ingewikkeld dit onderwerp eigenlijk is. Een open source-softwareproject runnen verschilt nogal van leiding geven aan een bedrijf. Stel je voor dat je constant over de hoedanigheid van je product moet onderhandelen met een groep vrijwilligers van wie je de meeste nog nooit hebt ontmoet! Om allerlei redenen lijkt het evenmin op het leiden van een traditionele non-profitorganisatie of een overheid. Er zijn overeenkomsten met al deze zaken, maar ik kom langzaam maar zeker tot de conclusie dat open source-software een geval apart is. Er zijn veel zaken waarmee het vergeleken kan worden, maar geen enkele is echt gelijkwaardig. En inderdaad, zelfs de veronderstelling dat open source-softwareprojecten ‘gerund’ kunnen worden, doet de werkelijkheid enigszins geweld aan. Een open source-softwareproject kan worden gestart en worden beïnvloed door geïnteresseerde partijen, vaak tamelijk sterk. Maar de waardevolle eigenschappen ervan kunnen niet het bezit worden van een enkele eigenaar en zo

lang er mensen zijn die ergens, waar dan ook, geïnteresseerd zijn in het voortzetten van het project, kan het niet eenzijdig stopgezet worden. Iedereen heeft onbegrensde macht en iedereen heeft geen macht. Ingrediënten voor een interessante dynamiek.

En dat is waarom ik dit boek wilde schrijven. Open source-softwareprojecten hebben zich ontwikkeld tot een specifieke cultuur, een levensopvatting waarbij de vrijheid om software datgene te laten doen wat je wilt dat software moet doen een dogma is. Toch is het resultaat van deze vrijheid niet een groepje losstaande individuen die hun eigen weg gaan met de code, maar enthousiaste samenwerking. En inderdaad, het vermogen om samen te werken op zich is een van de meest gewaardeerde vaardigheden in de wereld van de open source-software. Om deze projecten te kunnen managen moet iemand deel gaan uitmaken van een soort opgezwollen samenwerking, waar iemands vermogen om niet alleen samen te werken met een ander maar ook nieuwe manieren te bedenken voor deze samenwerking kan resulteren in een tastbaar voordeel voor de software. Dit boek probeert de technieken te beschrijven waarmee dit gedaan kan worden. Het is in geen geval compleet, maar het is in ieder geval een begin.

Goede open source-software is een waardig doel op zich en ik hoop dat lezers die zoeken naar een manier dit te realiseren blij zullen zijn met wat ze hier vinden. Maar daarnaast hoop ik iets over te brengen van het grote plezier dat je kunt beleven aan het werken met een gemotiveerd team van open source-ontwikkelaars en aan de interactie met gebruikers op de heerlijk directe manier die open source stimuleert. Deelnemen in een succesvol open source-softwareproject is leuk; uiteindelijk is dat wat het hele systeem in stand houdt.

WIE ZOU DIT BOEK MOETEN LEZEN?

Dit boek is bedoeld voor ontwikkelaars en managers van open software die overwegen een open source-project te beginnen, en voor degenen die al een project gestart zijn en zich afvragen hoe ze verder moeten. Het kan ook handig zijn voor mensen die alleen willen participeren in een open source-project, maar dit nooit eerder gedaan hebben.

De lezer hoeft geen programmeur te zijn, maar moet wel de technische basisconcepten van software kennen, zoals broncode, compilers en patches. Ervaring met open source-software, als gebruiker of ontwikkelaar, is niet nodig. Mensen die al aan open source-softwareprojecten hebben meegewerkt, zullen sommige delen van het boek waarschijnlijk als een open deur beschouwen en deze waarschijnlijk willen overslaan. Omdat er enorm grote verschillen kunnen zijn wat betreft de ervaring die de lezers hebben, heb ik geprobeerd de secties van duidelijke koppen te voorzien en te vermelden wanneer iets overgeslagen kan worden door degenen die al bekend zijn met deze materie.

BRONNEN

Veel van het basismateriaal voor dit boek heb ik verzameld gedurende de vijf jaar dat ik aan het Subversion-project (<http://subversion.tigris.org/>) gewerkt heb. Subversion is een open source-versiecontrolesysteem, dat helemaal nieuw geschreven is. Het is bedoeld om CVS te vervangen als het door de open source-gemeenschap geprefereerde versiecontrolesysteem. Het project is begin 2000 gestart door mijn werkgever, CollabNet (<http://www.collab.net/>). Godzijdank begreep CollabNet meteen vanaf het begin hoe dit project moest worden gerund als een werkelijk coöperatief en breed gedragen project. In het begin boden veel vrijwillige ontwikkelaars zich aan; op dit moment zijn er zo'n vijftig ontwikkelaars werkzaam aan het project, van wie slechts enkele medewerkers van CollabNet zijn.

Subversion is in veel opzichten een klassiek voorbeeld van een open source-project en ik voelde me er uiteindelijk veel meer door geïnspireerd dan ik oorspronkelijk verwacht had. Dit was gedeeltelijk een kwestie van gemak: telkens wanneer ik een voorbeeld nodig had van een bepaald fenomeen, schoot me er meestal meteen een van Subversion te binnen. Maar het was ook een kwestie van verificatie. Alhoewel ik op verschillende manieren betrokken ben bij andere open source-software-projecten en met vrienden en kennissen praat die betrokken zijn bij veel andere projecten, realiseerde ik me snel dat beweringen geverifieerd moeten worden op feitelijke juistheid als ze zwart op wit komen te staan. Ik wilde geen uitspraken doen over gebeurtenissen betreffende andere projecten die alleen gebaseerd zijn op wat ik kon lezen in de archieven van hun openbare mailinglijsten. Als iemand dat met Subversion zou doen, dan weet ik dat hij de helft van de tijd goed zou zitten en de andere helft fout. Dus wanneer ik inspiratie of voorbeelden gebruikte van een ander project waar ik geen directe ervaring mee had, dan probeerde ik eerst te praten met een informant van dat project, iemand waarvan ik zeker wist dat hij me kon vertellen wat er echt speelde.

Ik heb de afgelopen vijf jaar aan Subversion gewerkt, maar ik doe dit werk al twaalf jaar. Andere projecten die dit boek hebben beïnvloed zijn:

- Het *GNU Emacs text editor-project* van de Free Software Foundation, waarvan ik van een paar kleine onderdelen het onderhoud verricht.
- *Concurrent Versions System (CVS)*, waaraan ik in 1994-1995 intensief gewerkt heb met Jim Blandy, maar waarbij ik sindsdien slechts met tussenpozen betrokken ben.
- De verzameling open source-projecten die bekend staat als de *Apache Software Foundation* en in het bijzonder de *Apache Portable Runtime (APR)*- en *Apache HTTP-server*.
- *OpenOffice.org*, de *Berkeley Database* van Sleepycat en de *MySQL Database*. Ik ben niet persoonlijk betrokken geweest bij deze projecten, maar ik heb ze geobserveerd en, in sommige gevallen, gesproken met de mensen daar.
- *GNU Debugger (GDB)* (idem).
- *Het Debian-project* (idem).

Dit is uiteraard geen volledige lijst. Zoals de meeste open source-programmeurs houd ik op afstand verschillende projecten in de gaten, gewoon om een idee te hebben van de stand van zaken. Ik zal ze niet allemaal noemen, maar ze worden in de tekst vermeld wanneer dat van toepassing is.

DANKBETUIGING

Het schrijven van dit boek duurde vier keer zo lang als ik aanvankelijk dacht. Vaak voelde het als het zwaard van Damocles dat, waar ik ook was, boven mijn hoofd hing. Zonder de hulp van vele mensen zou ik niet in staat geweest zijn het boek af te maken en tegelijkertijd mijn verstand te bewaren.

Andy Oram, mijn redacteur bij O'Reilly, was de droom van elke schrijver. Behalve grondige kennis van de branche (hij opperde veel van de onderwerpen), bezit hij de zeldzame gave om te begrijpen wat je wil zeggen en je te helpen de juiste manier te vinden om het te zeggen. Het was een eer met hem te mogen werken. Mijn dank gaat ook uit naar Chuck Toporek, die dit plan direct naar Andy stuurde.

Brian Fitzpatrick heeft de revisie gedaan van bijna al het materiaal dat ik schreef. Dat heeft het boek niet alleen beter gemaakt, maar me ook aan het schrijven gehouden op de momenten dat ik overal elders wilde zijn behalve achter mijn computer. Ben Collins-Sussman en Mike Pilato hielden eveneens de voortgang in de gaten en waren altijd bereid om - soms zeer uitvoerig - te discussiëren, ongeacht het onderwerp dat ik die week onder handen had. Ze merkten het ook als mijn tempo omlaag ging en plaagden me er zo nodig op een vriendschappelijke manier mee. Bedankt jongens.

Biella Coleman schreef haar proefschrift in dezelfde periode dat ik dit boek schreef. Ze weet wat het is om te gaan zitten en iedere dag te schrijven en ze fungeerde voor mij als een inspirerend voorbeeld en een gewillig oor. Ze heeft ook een fascinerende antropologische kijk op de open source-softwarebeweging en ze heeft me ideeën en referenties aan de hand gedaan die ik voor het boek kon gebruiken. Alex Golub, nog een antropoloog met een voet in de open source-softwarewereld, die ook op dat moment zijn proefschrift schreef, was daarvoor al buitengewoon behulpzaam geweest.

Micah Anderson maakte ondanks zijn eigen schrijfklus altijd een opgewekte indruk, hetgeen op een misselijk en jaloers makende manier inspirerend was, en stond altijd klaar met zijn vriendschap, een gesprek en (in ieder geval één keer) technische ondersteuning. Bedankt Micah!

Jon Trowbridge en Sander Striker gaven zowel bemoedigende als concrete hulp. Dankzij hun brede ervaring op het gebied van open source-software leverde materiaal op waar ik op geen enkele andere manier aan had kunnen komen.

Mijn dank ook aan Greg Stein, niet alleen voor zijn vriendschap en aanmoediging op het juiste moment, maar ook voor het feit dat hij aan het Subversion-project liet

zien hoe belangrijk regelmatige evaluatie van de code is bij het opzetten van een programmeergemeenschap. Mijn dank gaat ook uit naar Brian Behlendorf, die het belang van openbare discussies tactvol in onze hoofden gestampt heeft. Ik hoop dat dit in het hele boek naar voren komt.

Dank aan Benjamin ‘Mako’ Hill en Seth Schoen, voor diverse gesprekken over open source-software en de daarbij horende politiek; aan Zack Urlocker en Louis Suarez-Potts voor het vrijmaken van tijd voor een interview ondanks hun volle agenda; aan Shane op de Slashcode-lijst voor zijn toestemming om zijn posts te mogen citeren en aan Haggen So voor zijn enorme behulpzaamheid bij het vergelijken van canned hosting-sites.

Mijn dank aan Alla Dekhtyar, Polina en Sonya voor hun niet-aflatende en geduldige aanmoedigingen. Ik ben erg blij dat ik onze avonden niet langer vroeg hoeft te beëindigen (of beter gezegd, zonder succes vroeg probeer te beëindigen) om naar huis te gaan en aan ‘Het Boek’ te werken.

Mijn dank aan Jack Repenning voor zijn vriendschap, gesprekken en koppige weigering om een makkelijke foutieve analyse te accepteren als er een moeilijkere goede voorhanden is. Ik hoop dat er iets van zijn lange ervaring met zowel het ontwikkelen van software als de software-industrie aan dit boek is blijven plakken.

CollabNet was uitzonderlijk royaal en stond me een flexibel rooster toe om te schrijven en klaagde niet toen het veel langer ging duren dan oorspronkelijk de bedoeling was. Ik ken de fijne kneepjes niet van hoe management tot zulke beslissingen komt, maar ik vermoed dat Sandhya Klute en later Mahesh Murthy er iets mee te maken hebben gehad. Mijn dank aan beiden.

Het hele Subversion-ontwikkelingsteam is de afgelopen vijf jaar een grote bron van inspiratie voor me geweest en veel van wat in het boek staat, heb ik geleerd door met hen samen te werken. Ik zal ze hier niet allemaal met naam en toenaam bedanken, omdat het er gewoon teveel zijn, maar ik verzoek elke lezer die iemand die aan Subversion meewerkt tegen het lijf loopt om deze onmiddellijk een drankje naar zijn of haar keuze aan te bieden. Ik ben dat zeer beslist van plan.

Vaak zeurde ik tegen Rachel Scollon over de stand van zaken van het boek. Ze was altijd bereid te luisteren en speelde het op de een of andere manier klaar om het probleem kleiner te laten lijken dan het leek voordat we praatten. Dat heeft enorm geholpen, bedankt.

Mijn dank aan (alweer) Noel Taylor. Hij zal zich, gezien al mijn geklaag de vorige keer, wel afgevraagd hebben waarom ik per se nog een boek wilde schrijven. Dankzij zijn vriendschap en leiding van Golosá is er muziek en kameraadschap in mijn leven gebleven, zelfs in de drukste tijden. Mijn dank ook aan Matthew Dean en Dorothea Samtleben, vrienden en lankmoedige muziekpartners, die begripvol bleven ook al stapelden de smoesjes waarom ik niet geoefend had zich op. Megan Jennings steunde me voortdurend en was oprecht geïnteresseerd in het onderwerp, ook al was ze er niet mee vertrouwd. Een geweldige elixer voor de onzekere schrijver. Bedankt, maatje!

Ik beschikte voor dit boek over vier goed geïnformeerde en toegewijde redacteurs: Yoav Shapira, Andrew Stellman, Davanum Srinivas en Ben Hyde. Als ik in staat was geweest om al hun uitstekende suggesties over te nemen, dan was dit een beter boek geworden. Helaas kon ik door tijdsgebrek alleen de krenten uit de pap pikken, maar de verbeteringen zijn al met al toch aanzienlijk. De resterende fouten komen geheel op mijn conto.

Mijn ouders, Frances en Henry, hebben me zoals altijd erg gesteund en omdat dit boek minder technisch is dan het voorgaande hoop ik dat ze het wat leesbaarder vinden.

Ten slotte wil ik degenen bedanken aan wie dit boek is opgedragen, Karen Underhill en Jim Blandy. Karens vriendschap en welwillendheid betekenen alles voor me, niet alleen tijdens het schrijven van dit boek, maar sowieso de afgelopen zeven jaar. Ik had het boek eenvoudigweg niet af kunnen krijgen zonder haar hulp. Hetzelfde geldt voor Jim, een ware vriend en een hackers hacker, die me de eerste beginselen bijbracht van open source-software, op een manier zoals een vogel een vliegtuig iets zou leren over vliegen.

DISCLAIMER

De gedachten en opvattingen die tot uitdrukking gekomen zijn in dit boek zijn geheel de mijne. Ze geven niet per definitie de opvattingen weer van CollabNet of van het Subversion-project.