



Online verwachtingen overtreffen, offline het verschil maken

Beantwoording
Marktconsultatie OpenZaak

Inhoudsopgave

- [A. Inleiding](#)
- [1. Single shared codebase](#)
- [2. Shared codebase governance](#)
- [3. Bugs en bugfixes](#)
- [4. Helpdesk en support](#)
- [5. Standards compliance](#)
- [6. Codebase marketing](#)

A. Inleiding

Ruim tien jaar geleden begon Mintlab B.V. met de gemeente Bussum aan de ontwikkeling van het eerste echte Open Source zaaksysteem van Nederland. Waar iedereen ons voor gek verklaarde, hadden wij een droom. Een softwareoplossing die volledig in Open Source samen met gemeenten wordt ontwikkeld. Dit bleek niet eenvoudig te zijn, maar het is ons gelukt. Ruim 45 organisaties maken elke dag gebruik van Zaaksysteem.nl. Software die na tien jaar nog steeds gezamenlijk wordt ontwikkeld onder de EUPL en waarvan de broncode voor iedereen beschikbaar is.

Zowel wij als gemeenten hebben al deze jaren een sterke behoefte om de informatiehuishouding van gemeenten drastisch te veranderen. Een verandering waarbij gemeenten werken op basis van API's, makkelijk verschillende toepassingen kunnen implementeren en dit alles met een belangrijke rol voor Open Source software. Zo hebben wij de afgelopen jaren gemeenten geholpen om naast Zaaksysteem.nl met de implementatie en het beheer van software die hieraan voldoet.

Sinds vorig jaar lijkt er eindelijk een doorbraak te zijn! De ontwikkelingen van Common Ground worden worden bestuurlijk gedragen en diverse organisaties starten allerlei initiatieven om tot nieuwe oplossingen te komen die passen in de architectuur van Common Ground. De ontwikkelingen van de Zaakgericht werken API's en OpenZaak zijn voor ons hier bij de meest belangrijke. Moeten wij dit als een bedreiging zien? Of juist als een enorme kans?

Wij hebben voor het laatste gekozen. Mintlab is een organisatie die met behulp van moderne software de dienstverlening en bedrijfsvoering van de overheid willen verbeteren. Maar bovenal een organisatie met een bijzondere schare mensen die met een enorme passie werken aan Open Source software. Naast Zaaksysteem.nl bieden wij vanaf dit jaar ook de koppeldienst Koppel.app aan en eigenlijk past OpenZaak perfect in dit plaatje. Wij hebben alles reeds in huis om dit te realiseren. Een volledig vernieuwd SaaS-platform van AWS wat dit jaar in gebruik is genomen (en beschikt over alle nieuwe technieken), een ontwikkelteam wat al in Python en Open Source ontwikkelt en allerlei mensen die gemeenten kunnen ondersteunen bij de implementatie en het beheer.

Wij hebben met plezier de vragen beantwoord, en hebben natuurlijk al even OpenZaak gedownload en uitgerold op ons SaaS-platform. Naast het schriftelijk beantwoorden van de vragen, zouden wij graag onze kennis en ervaring rondom het ontwikkelen, implementeren en beheren van een Open Source oplossing en community met jullie delen. Rest ons de vraag: morgen in productie?

Peter Moen en Michiel Ootjers

1. Single shared codebase

Veronderstellingen

We willen een open community op schaal faciliteren, inclusief samenwerking op een vendor-neutrale codebase en shared branching model, waarbij uiteenlopende forks worden vermeden. Dit willen we opnemen in de contracten die gemeenten hebben met leveranciers, bijvoorbeeld door het benoemen van afspraken over branches en een nadruk op continuous integration.

Vragen

1. *Hoe zou u dit principe willen opnemen in contracten die gemeenten met u afsluiten?*

Zaaksysteem.nl is een oplossing die reeds in Open Source is ontwikkeld met een single shared codebase. In de overeenkomsten die wij met gemeenten afsluiten staat beschreven dat alle software die wij ontwikkelen voor gemeenten aangeboden wordt in dezelfde codebase op basis van de EUPL. Het ligt dan ook voor de hand om een soortgelijk artikel te benoemen.

In het geval van OpenZaak zou dat betekenen dat wij een artikel opnemen in onze overeenkomst waarin staat vastgelegd dat alle broncode die wordt ontwikkeld, aangeboden wordt in de centrale repository.

2. *Hoe stelt u voor om 'soft forks' up-to-date te houden van elkaar? Moet er een centrale repository zijn die als master kan worden beschouwd?*

Hierboven lieten we al even de term “centrale repository” doorschemeren. Dit zou wel onze voorkeur genieten. Wanneer leveranciers committen om mee te gaan met deze repository, voorkom je dat er soft forks ontstaan en moedig je leveranciers tevens aan om softwareverbeteringen “upstream” aan te bieden. Dit plaatje past naar ons idee het beste bij de principes van een landelijke standaard, maar opent wel het risico van een logge innovatie-snelheid waar de maintainers van de centrale repository goed op moeten letten.

Wij hebben het model “volg de centrale repository letterlijk” mogen ervaren bij een leverancier van het cloud-platform OpenStack. Het voordeel is dat de versies van OpenStack snel opgevolgd kunnen worden omdat de dienst van de leverancier niet verweven is met maatwerk wijzigingen, maar je bent wel afhankelijk van de functionaliteiten van de “centrale repository”. Gelukkig ligt de ontwikkelsnelheid van de centrale repository van OpenStack hoog, en komen nieuwe functionaliteiten snel beschikbaar.

Leveranciers kunnen in hun overeenkomst met gemeenten opnemen dat een oplossing voor productiedoeleinden altijd gebruik moet maken van de master repository zodat de voor- en eventuele nadelen van dit model voor zowel leverancier als gemeente duidelijk is.

3. *Welke uitdagingen zijn er voor u om in zo'n multi-branch omgeving te werken? Hoe kunnen deze uitdagingen worden beperkt om de bedrijfsrisico's voor u te verminderen?*

Financieel verwachten we dat de risico's afhankelijk zijn van het gekozen verdienmodel. Als Open Source leverancier besef je dat het bestaansrecht van de organisatie afhankelijk is van aanvullende en ondersteunende diensten op OpenZaak. Als we uitgaan van het voorgestelde model uit de voorgaande vragen kan je je namelijk niet onderscheiden met het ontwikkelen van functionaliteiten aan de software OpenZaak, deze worden immers "landelijk" aangeboden en zijn voor elke leverancier bruikbaar.

Wat betreft doorontwikkeling nodigt dit model leveranciers mogelijk onvoldoende uit tot het verbeteren of innoveren van de software OpenZaak. Om dit aan te moedigen denken wij aan een aan de centrale repository gebonden ontwikkelbudget, een centraal geregeld budget, waar elke leverancier uit kan putten wanneer deze werkt aan de doorontwikkeling van de centrale repository. Dit is de meest transparante oplossing voor zowel leverancier als gemeenten. Op deze manier wordt voorkomen dat leveranciers hun prijzen voor het beheer verhogen om ontwikkeling te kunnen vergoeden, en resulteren vergoedingen aan ontwikkelaars vanuit dit centrale "budget" direct in verbeteringen aan OpenZaak.

Om een vlotte doorontwikkeling te garanderen, zouden wij adviseren dat het beheer van de repository breed belegd wordt, zodat aangeboden wijzigingen (pull requests) snel kunnen worden gepubliceerd. Op deze manier zou het risico van de "logge innovatie" beperkt kunnen worden.

4. *Heeft u ooit eerder op deze manier gewerkt?*

Ja, wij hebben hier ervaring mee. Wij hebben twee softwareoplossingen waarbij het voorkomt dat de verschillende scrum-teams wijzigingen willen aanbieden bij de softwareoplossing 'van de overkant' en krijgen wij externe pull requests aangeboden.

Daarnaast worden alle functionaliteiten aan ons zaaksysteem centraal gereleased nadat een grote pool van gemeenten hun ervaringen hebben gedeeld in de acceptatieperiode. Bewust geen maatwerk, en elke gemeente hetzelfde OpenZaak platform.

5. *Is de huidige architectuur en make-up van de codebase bevorderlijk voor multi-vendor samenwerking en modulaire implementatie van componenten?*

Dit zou naar onze mening geen probleem hoeven te zijn. Er zijn diverse Open Source oplossingen waarbij meerdere ontwikkelbedrijven samen het product ontwikkelen. In die zin doen wij dit nu ook. Er zijn meerdere teams die aan Zaaksysteem.nl werken, en op gezette momenten worden externe bedrijven ingehuurd om te ontwikkelen.

Zaaksysteem.nl is daarmee het bewijs dat het kan.

Verder is dit wellicht de plek in deze marktconsultatie om aan te geven dat wij verwachten dat er wijzigingen nodig zijn binnen de architectuur om de applicatie op een schaalbare manier als SaaS-oplossing te kunnen aanbieden. Wij zien de performancetesten als een goede start, maar 5000 gelijktijdige gebruikers is slechts een fractie van wat Zaaksysteem.nl momenteel gelijktijdig verwerkt. Daarbij leert de ervaring dat de praktijk er altijd anders uitziet en er meer knelpunten in het proces kunnen zitten. Wij zouden willen aanmoedigen om, gezamenlijk, kritisch te kijken hoe bepaalde componenten losgetrokken kunnen worden en als losse service aangeboden kunnen worden om monolietvorming te voorkomen.

2. Shared codebase governance

Veronderstellingen

Onze visie is om samenwerking mogelijk te maken tussen meerdere gemeenten en leveranciers, die samen aan de codebase werken. Hiervoor is een gedeeld bestuur nodig voor de codebase. Dit kan onder meer het volgende behelzen: a) een open en gedeelde issue en bug tracker, b) een open en gedeelde feature roadmap, en c) een open en gedeelde technical roadmap. Dit omvat ook engineering en contributing guidelines, evenals open communicatiekanalen.

Vragen

1. *Hoe vindt u dat governance moet worden georganiseerd binnen de community?*

Dit is een vraagstuk waar wij als Open Source oplossing al tien jaar mee bezig zijn en waarin wij diverse successen en lessons learned hebben. Als Open Source oplossing hebben wij een aantal onderdelen waarvan wij vinden dat het bijdraagt aan het doel om gezamenlijk de oplossing te werken:

- a. Een gebruikersvereniging met een betrokken bestuur;
- b. Een gezamenlijke en open roadmap;
- c. Een community-platform om kennis en ervaringen te delen;
- d. Een gezamenlijke issue en bug-tracker;

Wat wij verder hebben geleerd, is dat het belangrijk is dat gemeenten naast diverse 'digitale' samenwerking ook in fysieke vorm graag samenwerking. Met de huidige coronavirus wellicht even niet, maar dat niet meegerekend, hebben wij vastgesteld dat het tevens belangrijk is om fysieke bijeenkomsten te organiseren waarin diverse projecten rondom ontwikkeling, implementatie en beheerder worden besproken en/of uitgevoerd.

2. *Wat zijn de uitdagingen die u voorziet bij het onderhouden van open backlogs bij uw klanten?*

De grootste uitdaging die wij verwachten bij het onderhouden van een (open) backlog is het afstemmen van de juiste prioritering. Uit onze ervaring weten wij dat veel gemeenten met wensen komen en iedere gemeente haar ontwikkelproject het meest belangrijk vindt. Het is dus heel mooi dat gemeenten zelf projecten kunnen initiëren, maar als er meer projecten worden aangeleverd dan dat er capaciteit is, dan is het van belang om een gezamenlijke methode te hebben voor het prioriteren van wijzigingen.

Een methode die bij ons goed werkt, is het vaststellen van een gezamenlijke visie en ontwikkelstrategie. Deze visie en strategie vormt de basis voor de gezamenlijke Roadmap. Het helpt ons om gemeenten op één lijn te krijgen wanneer er verschillen

ontstaan in de prioritering van projecten die op de gezamenlijke roadmap staan.

3. *Wie moet verantwoordelijk zijn voor het bijwerken van de shared roadmap? U, de gemeenten, of een gedeelde verantwoordelijkheid?*

Bij voorkeur wordt een roadmap door een Product Owner beheerd. Omdat de rol Product Owner reeds verantwoordelijk is voor de Product Backlog, kan deze rol ook daadwerkelijk invloed uitoefenen bij het ontwikkelen van de roadmap. In het geval van Zaaksysteem.nl is de rol Product Owner verantwoordelijk voor de shared roadmap én de Product Backlog.

Echter, in bovenstaand model van een centrale repository waar meerdere leveranciers ontwikkelen aan de broncode en meerdere gemeenten een actieve rol hebben denken wij dat het voorgestelde Product steering team een goed model is. Wij zouden het Product Team noemen, maar hé 'what's in a name'.

Wij hebben (nog) geen ervaring met een Product Steering Team, maar wij zouden adviseren dat dit team wordt bemand met mensen vanuit de gemeenten, maar ook met mensen vanuit de leveranciers. Door de verschillende partijen in dit team te plaatsen, blijft er een goede balans tussen wat de oplossing moet doen, en wat realistisch is om te ontwikkelen en te beheren.

4. *Hoe wilt u de gedeelde feature en technical roadmap kunnen beïnvloeden?*

Wij zouden graag zien dat het mogelijk is om een voorstel in te dienen, en dat het duidelijk is aan welke voorwaarden het moet voldoen en hoe het besluitvormingsproces hiervan verloopt. Verder zou iedereen een voorstel moeten kunnen indienen (gemeenten, zzp-er of leverancier) zolang het aan de voorwaarden voldoet. Met voorwaarden bedoelen wij dat het helder moet zijn wat het doel van een project is, op welke manier het ontwikkeld wordt, wat het effect op beheer is en meer van dat soort vragen. Bij voorkeur zou het voorstel en de besluitvorming daarvan volledig openbaar moeten zijn.

Zo hebben wij ervaring met het 'marktplaats-model' waarbij elke gemeente een project kan indienen. Nadat het project beoordeeld is, maken wij een specificatiedocument en kunnen gemeenten 'doneren' op het project totdat het aan alle voorwaarden voldoet en op de roadmap kan worden geplaatst.

Dit jaar hebben we het marktplaats-model gewijzigd naar de featurewaaier. Omdat er steeds meer gemeenten gebruikmaken van onze oplossing, verzamelt onze Product Owner alle wensen en voegt deze zoveel mogelijk samen in ontwikkelprojecten die in lijn zijn met de visie en strategie, en een groot gezamenlijk belang hebben. Gemeenten kunnen vervolgens zelf bepalen aan welke projecten zij actief willen deelnemen.

5. *Hoe wilt u dat de community omgaat met het ontwikkelen, eens worden over en onderhouden van gedeelde engineering en contributing guidelines?*

Wij kunnen ons prima vinden in een model waarbij dit wordt belegd bij het Technical Steering Team.

6. *Welke risico's ziet u in het werken in zo'n omgeving?*

Het "no objections" decision making model binnen het technical steering team lijkt ons afdoende. De risico's die wij zien van het product steering team is wanneer er geen gelijke balans is tussen leveranciers en gemeenten. Waar gemeenten de leveranciers veel kunnen leren over hun informatiehuishouding, hebben leveranciers op hun beurt veel ervaring in het bouwen van schaalbare en generieke oplossingen die voor iedereen optimaal werken.

Het gevolg van het missen van deze gezonde balans is dat er een klassieke 'kant-leverancier-relatie' gaat ontstaan waarbij vanuit de gemeente de doelstellingen van de gemeente voorop komen te staan, en vanuit de leverancier het bedrijfsbelang. Dit terwijl het doel is om gezamenlijk een oplossing te ontwikkelen, te gebruiken en te beheren. Hierbij is het van belang dat er een goed evenwicht is van de verschillende rollen in de besluitvorming.

3. Bugs en bugfixes

Veronderstellingen

Er zal een centrale, openbare en geprioriteerde issue tracker zijn voor features en bugs. Veel van deze problemen kunnen kleine hoeveelheden werk (minder dan 80 uur) omvatten. Daarnaast zullen veel gemeenten issues en bugs hebben die verband houden met hun lokale implementatie en context.

Vragen

1. *Moet het werken op zowel lokale implementaties en de centrale repository worden opgenomen in dezelfde of aparte contracten?*

Wij verwachten dat het beter is om dit in aparte contracten onder te brengen. Onze ervaring leert dat lokale implementaties en saas-oplossingen verschillende karaktereigenschappen hebben. Zo is het denkbaar dat een lokale implementatie zelfstandig wordt aangepast door een gemeente, al dan niet met behulp van een leverancier. Het is aan de gemeente zelf om deze wijzigingen wel of niet direct op productie te plaatsen.

Voor een SaaS-oplossing ligt dit anders. Er moet met meer gemeenten rekening worden gehouden alvorens de wijziging op een productieplatform kan worden gereleased. Zo zijn er nog meer voorbeelden te noemen waarbij de taken, verantwoordelijkheden en bevoegdheden verschillen.

Overigens willen wij de mogelijkheid niet onbenut laten om te benoemen dat naar onze mening een centrale oplossing vanuit een SaaS-oplossing altijd de betere keuze is. Met name omdat dit het gezamenlijk ontwikkelen en gebruiken van dezelfde oplossing bevordert.

2. *Hoe wilt u dat de contracten zo gestructureerd zijn dat u kunt werken aan lokale implementaties en branches en ook kunt bijdragen aan de centrale repository?*

Naar onze mening zou het goed zijn om in alle gevallen in het contract vast te leggen dat wijzigingen (of deze nu lokaal of vanuit SaaS-model zijn) moeten worden aangeboden aan de centrale repository.

3. *Hoe moeten bugfixes in de hele community worden geïmplementeerd? Kunnen leveranciers hotfixes van andere leveranciers overnemen voordat deze zijn samengevoegd in de centrale repository? Of moeten leveranciers pas wijzigingen doorvoeren nadat ze zijn samengevoegd in de centrale repository?*

Vanuit ons Open Source standpunt zouden leveranciers OpenZaak vanuit de centrale repository moeten installeren. Maar het succes hiervan valt of staat met de snelheid waarmee het Technical Steering team de bugfixes en functionaliteiten accepteert in de centrale repository. Want in principe moet iedereen de mogelijkheid

hebben om de software te verbeteren en zouden er geen belemmeringen moeten zijn voor een andere organisaties om gebruik te maken van een nieuwe wijziging.

Dit neemt niet weg dat een zekere mate van flexibiliteit gewenst is. Wanneer er een beveiligingslek zou zijn gevonden, wil iedereen natuurlijk zo snel mogelijk een hotfix plaatsen en daarna eventueel nog een update. Maar eerst dat lek dichten.

4. *Wat voor soort aansprakelijkheidsvraagstukken zouden er kunnen zijn indien 'hot fixing' onverwachte schade of verdere problemen oplevert? Levert dit een probleem op voor uw contractuele verplichtingen aan uw klant?*

Als SaaS-leverancier zijn wij gewend dat wij verantwoordelijk zijn voor het leveren van een oplossing met hoge betrouwbaarheidseisen. Het is aan ons als leverancier om een juiste afweging te maken bij het wel of niet plaatsen van een nieuwe versie. De huidige contracten voorzien hier voldoende in. Echter, wanneer we CI/CD in ogenschouw nemen, dan voorzien wij de huidige contracten hier onvoldoende in, en zou het ons helpen als we hier met de community richtlijnen voor kunnen maken.

4. Helpdesk en support

Veronderstellingen

Veel gemeenten zullen de voorkeur geven aan één enkele leverancier die ze kunnen bellen voor al hun behoeften. Dit varieert van vragen, bugs, ideeën voor doorontwikkeling, technische integratie, training, configuratie van machtigingen / autorisatie, meldingen, enzovoort. Onze veronderstelling is dat in veel gevallen, geen enkele leverancier al deze diensten kan of wil leveren.

Vragen

1. *Is het mogelijk om één helpdesk op te zetten voor alles wat te maken heeft met het gebruik van OpenZaak door een gemeente? Of zijn er voordelen voor verschillende leveranciers die verschillende helpdesks onderhouden voor verschillende soorten vragen (bijvoorbeeld om hen in staat te stellen hun serviceaanbod te differentiëren?)*

Ja dat kan. Dit is wat wij als organisatie elke dag doen voor onze gemeenten. We bieden Zaaksysteem.nl als SaaS-oplossing aan, voeren onderhoud uit, bieden ondersteuning bij de implementatie, verzorgen trainingen functioneel beheer en ontwikkelen samen nieuwe functionaliteiten. Wij zien dit als de typische ondersteunende en aanvullende diensten die horen bij een Open Source product. Wat ons betreft is de veronderstelling dus niet juist.

In het geval van OpenZaak zou dit precies hetzelfde zijn. De enige voorwaarde die wij zullen stellen is dat de oplossing als SaaS-geleverd wordt. Wij zijn niet de aangewezen partij voor het leveren van deze diensten voor een lokale implementatie.

2. *Kunnen meerdere marktpartijen zich verenigen om hun aanbod te combineren?*

Dit behoort ook tot de mogelijkheden. Ook dit is een model wat bij ons al praktijk is. Zo bieden wij producten en diensten via geselecteerde partners. In het geval van OpenZaak zou dit bijvoorbeeld kunnen betekenen dat wij wel het beheer en de implementatie van de software doen, maar niet de ontwikkeling. Of juist andersom. Het grootste gedeelte van ons ontwikkelteam ontwikkelt in Python. Wij zouden wel de ontwikkelingen kunnen uitvoeren, maar het beheer elders beleggen.

3. *Kunnen gespecialiseerde 'helpdeskaanbieders' werk uitbesteden aan andere aanbieders?*

Wellicht is dit mogelijk. Wij hebben hier onvoldoende ervaring mee. Wij geloven in het ondersteunen van onze gemeenten bij alles wat nodig is voor onze oplossingen. Vanuit die visie hebben wij onze helpdesk, ontwikkelteam, consultancyteam en beheerteam (DevOps) altijd in dezelfde organisatie gehad.

5. Standards compliance

Veronderstellingen

De OpenZaak API's zijn gebaseerd op een VNG standaard, de API's voor zaakgericht werken. De standaard wordt continu onderhouden en verbeterd en OpenZaak moet daaraan blijven voldoen. Deze updates moeten worden gepropageerd naar de centrale repository-branch, evenals repositories die marktpartijen gebruiken om hun diensten te leveren (zoals backend-as-a-service).

Vragen

1. *Hoe zorgen we ervoor dat updates van de VNG-standaard op de juiste manier worden doorgevoerd in lopende OpenZaak-implementaties? Moeten deze vereisten worden opgenomen in contracten met marktpartijen?*

Wanneer we het “landelijke repository” model hanteren, dan zou het niet nodig moeten zijn om dit contractueel te verplichten: een leverancier wordt immers verleid door gemeenten om de master van deze repository in productie aan te bieden, omdat deze de laatste functionaliteiten bevat.

Het toevoegen van de VNG standaard aan de “landelijke repository” kan dan geprioriteerd worden door het centrale product steering team.

2. *Wat vindt u van versiebeheer van OpenZaak ten opzichte van andere toepassingen? Hoeveel backwards-compatible versies moeten worden ondersteund?*

Als het hier gaat om het ondersteunen van verschillende API koppelvlakken willen we voorstellen om het aantal versies te beperken tot maximaal 2 of 3. In tegenstelling tot StUF is het met JSON API's eenvoudiger om “backwards compatible” wijzigingen aan te brengen. Denk aan het toevoegen van velden zonder verplichtheidsaanduiding. Ook dit kan eventueel meegenomen worden.

3. *Moeten leveranciers in het kader van hun contracten worden gevraagd om wat ze hebben geleerd bij het implementeren en ondersteunen van OpenZaak te laten bijdragen aan de VNG-standaard?*

Wij vinden dat leveranciers (en gemeenten!!!) verleid moeten worden om kennis en ervaringen te delen. Wanneer er een goede community aanwezig is, dan is onze ervaring dat dit een automatisch gevolg is. Het verplichten van dit middels contracten vinden wij slecht passen bij Open Source.

6. Codebase marketing

Veronderstellingen

Om de community effectief te laten schalen, gaan we ervan uit dat er een set neutraal marketingmateriaal moet zijn, waaronder logo, website, productdocumentatie, casestudies, business case materiaal, enzovoort. Wij zijn van mening dat dit leveranciers- en gemeente-neutraal moet zijn om een gedeelde set storytelling resources mogelijk te maken.

Vragen

1. *Op welke wijze zou u willen dat dit marketingmateriaal wordt gemaakt?*

Het zou mooi zijn als dit materiaal ook met een open source / create commons licentie wordt vrijgegeven en ter beschikking wordt gesteld.

2. *Wat voor invloed wilt u hebben op dit materiaal en de website?*

Als het gaat om logo's e.d. lijkt het ons minder wenselijk dat iedereen daar invloed op heeft. Als het gaat om casestudies en modeldocumenten dan is het wellicht fijn dat er modellen zijn die iedereen kan invullen. Op die manier kan het neutraal blijven, en beschikt elk document over dezelfde informatie. Als leverancier zou het dan prettig zijn dat je een business case of casestudie kan uploaden.

3. *Moeten leveranciers contractueel worden gevraagd om, wat ze hebben geleerd bij het implementeren en ondersteunen van OpenZaak, toe te voegen aan het OpenZaak marketingmateriaal (inclusief bijvoorbeeld user testimonials, business cases, etc)?*

Ook hier geldt weer dat wij dit niet goed bij een Open Source cultuur vinden passen. Het is beter om zowel gemeenten als leveranciers te verleiden dan te verplichten om hier aan bij te dragen.