

De Agile Architect

Joost Lommers, Arjen Uittenbogaard, Ruud van Vliet

Inleiding

Een architect is geen showstopper, geen boe-roeper, geen luchtffietser. Helaas komen we hem maar al te vaak in een of meer van die rollen tegen. Wij menen dat dit heeft te maken met gebrek aan ‘agility’¹! Een agile architect weet dat een architectuur niet in beton gegoten is, dat standaarden aangepast kunnen worden en dat minder regels beter is dan veel regels. Een agile architect weet dat je de effectiviteit van een architectuur alleen maar kunt beoordelen tijdens de toepassing ervan.

Agility en architectuur, een tegenspraak?

De agile architect behoudt de essentie, terwijl de details niet in lijn met het geheel hoeven te zijn. Veel organisaties zijn op zoek naar een meer agile bedrijfsvoering, zodat ze sneller op veranderingen kunnen inspelen. Organisaties willen wendbaarder zijn. Dit betekent dat hun informatievoorziening dat ook moet zijn. Een foute poging om een dergelijke wendbaarheid te realiseren is al jaren bekend: elke wens, elke verandering van inzichten of omstandigheden wordt aangepakt met een nieuw systeem. De ultieme consequentie hiervan wordt beschreven met metaforen als eilandautomatisering en verzuilde monolieten. De kosten voor exploitatie en beheer rijzen de pan uit, en ook nieuwe ontwikkelingen worden in deze situatie steeds duurder.



De Agile Architect behoudt de essentie, terwijl de details niet in lijn met het geheel hoeven te zijn

Architectuur biedt perspectief in deze situatie van wendbaarheid op de korte termijn. Door ook na te denken over de lange termijn, voorkomt de architect dat in de haast van alledag te gemakkelijk verkeerde keuzes worden gemaakt. Het lastige is natuurlijk om niet door te slaan de andere kant op en de wendbaarheid uit het oog te verliezen. Agility en architectuur zijn wat ons betreft een ideale combinatie. Agility zonder architectuur loopt het gevaar zich te verliezen in het hier en nu. Architectuur zonder agility verwordt tot een dogmatische bureaucratie.

De moeilijkheden bij het balanceren van architectuur en agility, en ook de oplossingen die erbij te bedenken zijn, zijn in verschillende categorieën in te delen. Je kunt kijken naar moeilijkheden die te maken hebben met de

¹ Agile betekent letterlijk ‘wendbaar’ maar heeft door het ‘Manifesto for Agile Software Development’ aanvullende betekenis; we verkiezen daarom ‘agile’ boven wendbaar.

technieken van architectuur. Vanzelfsprekend is tooling die spaghetti-code genereert en in beton giet niet de meest geëigende technologie om agile te blijven. Evenzo spreekt het vanzelf dat een organisatiecultuur waarin elke beslissing in drievoud moet worden goedgekeurd door vijf lagen van lijnmanagers niet bevorderlijk is voor wendbaarheid. En volgens Baets (2002) werkt deze structurering zelfs contraproductief. Omdat we techniek een ondergeschikt probleem vinden en cultuurverandering ons iets te ambitieus is, richten we ons in dit artikel op de persoon van de architect. De persoon dus die steeds weer de keuzes moet maken tussen korte- en lange-termijndenken. De persoon die steeds weer het goede voorbeeld moet geven. De persoon die moet durven toegeven dat hij het ook niet weet.

Agility en de architect

Jim Highsmith, een van de voormannen van de agile-beweging, vatte ooit de kern van agile werken samen als een cyclus waarin steeds weer de volgende drieslag wordt herhaald: Speculate-Collaborate-Learn (Highsmith, 1999). Deze cyclus vat de kern van het Agile Manifesto als volgt samen:

- Om wendbaar te zijn moet je durven besluiten, ook al zijn de gevolgen niet volledig te overzien en ook al weet je dat er veranderingen op komst zijn (speculate);
- Door constructief samen te werken, op basis van de genomen besluiten, kom je snel tot goede resultaten (collaborate);
- Je weet dat je besluiten en resultaten niet in een keer goed zullen zijn. Wees bereid om besluiten te heroverwegen als blijkt dat het beter kan en je bent een stapje verder op de goede weg (learn).

Veranderingen opvangen, beoordelen en absorberen vraagt een multidisciplinaire samenwerking van mensen die naar elkaar willen luisteren en van elkaar willen leren. Een agile architect regisseert dit proces en zal zich daartoe ook zelf constructief en luisterend dienen op te stellen.

Een agile architect heeft geen oogkleppen op en werkt niet toe naar één optimum. Zoals een programmamanager meerdere projecten beoordeelt ten aanzien van de organisatiedoelstellingen, neemt ook de architect meerdere projecten in ogenschouw. Hij stelt zich daarbij steeds de vraag wat het bedrijf op dit moment nodig heeft. Het bedrijf bestaat uit verschillende belanghebbenden die het met elkaar eens moeten worden over de gewenste kwaliteiten. Het is aan de architect om deze prioritering te begeleiden en op basis van de uitkomst een visie op een oplossing te creëren. Architectuur is dus altijd situationeel en geeft binnen de huidige context zicht op een oplossing. Gisteren was het anders, morgen is het vast weer anders. De agile architect weet dit en zoekt ondertussen naar de basisconcepten die gisteren, via vandaag, met morgen verbinden.

Instrumenten voor de Agile Architect

Een architect heeft een breed palet aan verschillende instrumenten nodig. Dit palet bevat niet alleen informatiekundige hulpmiddelen als methoden en technieken, maar ook technieken voor portfoliomanagement, investeringsmodellen en krachtenveldanalyses. Wat een architect agile maakt, is zijn vaardigheid om op het juiste moment de juiste

Instrumenten voor de agile architect

Agile Software Development heeft een aantal instrumenten voortgebracht die ook uitstekend bruikbaar zijn voor de architect. Deze instrumenten verhogen de agility van de (IT) organisatie. Onderstaand een selectie.

Werk incrementeel

Werk kan opgedeeld worden in kleinere delen die elk op zich waarde hebben. Door het opdelen kan die waarde incrementeel opgeleverd en getoetst worden. Hierdoor kan de waarde voor de stakeholders eerder gevalideerd worden en kan worden bijgestuurd, indien gewenst. Essentieel is dat de opdeling gestuurd wordt door waarde voor de stakeholders.

Test alles

In een agile omgeving moet al het werk direct gerelateerd kunnen worden aan de waarde die het oplevert voor de stakeholders. Het team moet, zodra iets kan worden opgeleverd, uitgebreid controleren of de gevraagde waarde is bereikt en de stakeholders hierbij betrekken. De tests zijn bedoeld om de waarde vast te stellen en samen met de stakeholders te bepalen welke wijzigingen, wanneer en tegen welke kosten moeten worden doorgevoerd.

Een informatieve werkruimte.

Een Informatieve Werkruimte levert grote, zichtbare 'Information radiators' (Cockburn, 2001) over de belangrijke doelstellingen, planningsinformatie (zoals de datums en inhoud van de volgende iteratie), voortgang en de waarde voor de stakeholders. De lijsten moeten onmiddellijk, zonder verdere uitleg, begrepen kunnen worden en niet voor meerdere interpretaties vatbaar zijn. Het moet in één oogopslag op te nemen zijn door voorbijgangers zonder er vragen bij te hoeven stellen. 'Information radiators' hebben vaak handmatig onderhoud nodig. Een grafiek met het geplande en het gerealiseerde verbruik van uren, gecombineerd met de geplande en behaalde resultaten, is een typische informatie radiator. Met gekleurde post-its kan aanvullende informatie worden gegeven. White boards kunnen worden gebruikt om korte termijn afspraken en lopende discussies weer te geven.

Respond to Change

Om in staat te zijn om op veranderingen in te kunnen spelen moeten plannen en planningen verandering als uitgangspunt nemen. En natuurlijk moeten ze ook gewijzigd worden. De agile aanpak heeft een goed inzicht in de afwijkingen van de plannen en biedt goede ondersteuning om alternatieven door te voeren.

Verwijder vertragende obstakels

De eenvoudigste manier om sneller te gaan is het verwijderen van alle vertragende obstakels. Een vertragend obstakel is elk gedrag, fysieke indeling, procedure of controlepunt dat het werk vertraagt zonder zelf waarde toe te voegen. Een activiteit die waarde toevoegt, kan vertraagd worden door obstakels, maar is op zich geen obstakel. De 'Kunst van het Verwijderen van de Vertragende Obstakels' is dus vooral gelegen in het doordringen tot de kern van de vertraging, zonder waarde te verliezen.

technieken in de juiste mate in te zetten: hij vraagt zich af wat er *nu nodig* is. De agile architect creëert duidelijke beslismomenten om deze afweging regelmatig te maken. Een aantal instrumenten die hem daarbij helpen zijn beschreven in het kader 'Instrumenten voor de agile architect'.

Voorbeelden van een agile aanpak

De agile architect werkt naar de geest van de methoden, en niet naar de letter. Hij heeft de kracht en het lef om te creëren en te innoveren. De kern van agility zit in het durven beslissen. Een beroemde uitspraak die is toegeschreven aan Grady Booch luidt: "(Software) architecture is about making important decisions in the dark". In de praktijk worstelen veel architecten met de, soms onbewuste, drang naar compleetheid. Is er eenmaal een architectuurraamwerk gekozen², dan lijkt het vanzelfsprekend om alle views in zo'n raamwerk in te vullen. Hierbij wordt al snel het vullen van het raamwerk een doel op zich. Echter, je kunt als architect niet compleet zijn, en moet daar ook niet naar streven.

² We stellen hier voor het gemak dat een architectuurraamwerk waarde toevoegt in deze specifieke situatie. De agile architect stelt dit uitgangspunt natuurlijk wel eerst ter discussie.

De agile architect kiest liever een kleiner raamwerk, met alleen views die nuttig zijn voor de problematiek van de eigen organisatie. De Wit, Lommers en Klinkenberg beschrijven zo'n traject (2004). Of de agile architect prioriteert de views in een bestaand raamwerk, en vult alleen de belangrijkste views. Dit kan zowel voor een referentie-architectuur, als voor projectarchitecturen gedaan worden. Op basis van bijvoorbeeld Value Disciplines (Treacy & Wiersema, 1997) of het type project is al snel een prioritering van belangrijke views te maken.

Een agile architect speelt ook met de diepgang van uitwerking. De auteurs van het DYA-raamwerk (Wagter et al., 2001) stellen voor om per architectuurdomein algemene principes, beleidslijnen en modellen uit te werken. Dat is mooi als hier tijd voor is, maar de agile architect kan er ook voor kiezen om alleen de algemene principes uit te werken. Of de algemene principes en beleidslijnen voor de belangrijkste domeinen. Als hij het slim aanpakt, zorgt hij ervoor dat in de projecten de bijbehorende modellen verder worden uitgewerkt. Deze zijn dan vervolgens te "oogsten" na afloop van de projecten, bijvoorbeeld voor opname in de referentie-architectuur.

Een andere manifestatie van compleetheidsdrang toont de architect die betrokken wil zijn bij ieder project dat loopt. Bij een financiële dienstverlener werden de architecten hierdoor steeds minder geliefd. Bij gebrek aan capaciteit sprongen ze in projecten steeds later en steeds minder gedetailleerd bij. Hun toegevoegde waarde werd steeds minder. De agile architect had het niet zo ver laten komen. Hij zorgt voor een checklist waarmee beoordeeld kan worden welke projecten in welke mate architectuurondersteuning nodig hebben. Dit geeft duidelijkheid voor alle betrokken partijen en de architecten kunnen meerwaarde leveren op die projecten waar het daadwerkelijk uitmaakt.

Het manifest voor de Agile Architect

De agile beweging is begonnen met het opstellen van het Agile Manifesto. Met een knipoog hierbij onze variant: het Manifest voor de Agile Architect.

De Agile Architect:

- Weet dat kiezen belangrijker is dan de keuze – en houdt de gevolgen van de keuze nauwlettend bij om zo nodig bij te kunnen sturen;
- Weet wat hij wil – en werkt daarom samen met anderen;
- Weet dat zijn architectuur nooit af is – en zorgt dat iedere versie beter is dan de vorige.

Doe mee op AgileArchitect.nl

Hoe agile zijn wij als schrijvers van dit artikel? Een artikel dat in een boek gepubliceerd is, ligt nu eenmaal vast. Bovendien: hoewel wij als auteurs wel erg agile te werk zijn gegaan tijdens het schrijven (speculate, collaborate, learn), u als lezer was er niet bij betrokken. Toch vinden wij dat net zo min als architectuur en agility in tegenspraak zijn, publiceren en agility dat hoeven te zijn. Dit artikel kan nieuwe inzichten opleveren. Het leren gaat door.

De agile architect weet dat iets nooit af is. Dit artikel is hooguit een momentopname, een baseline. De volgende versie gaan we misschien wel samen met u maken. Meld u aan op AgileArchitect.nl (www.agilearchitect.nl) en denk, doe en beslis mee. We willen graag met u speculeren over hoe het moet. Met u samenwerken om het zo te krijgen. En zo, werkende weg, gezamenlijk leren.

Bio's

Joost Lommers (jlommers@cibit.nl) werkt als senior adviseur bij CIBIT Adviseurs|Opleiders. Hij heeft een HIO-diploma en werkt sinds 1989 in de IT-sector. Noem een methode en Joost kent 'm, heeft er opleiding in gegeven of heeft 'm gebruikt. Joost doet sinds 1995 "aan architectuur". Ook als toetsenist in een band wordt hij wekelijks geconfronteerd met twee waarheden: alleen de toepassing van theorie leidt tot resultaten (in dit geval muziek), en teamwerk leidt tot een product dat meer is dan de som van de afzonderlijke inspanningen.

Arjen Uittenbogaard (auittenbogaard@cibit.nl) werkt als senior docent en adviseur bij CIBIT Adviseurs|Opleiders. Hij geeft al jaren trainingen en advies op het gebied van object-oriëntatie, agile development en aanverwante ontwikkelingen. Langzaam raakt hij ervan overtuigd dat hij met zijn andere passie het vakgebied misschien nog wel meer van dienst kan zijn: improvisatietoneel.

Ruud van Vliet (ruud.van.vliet@caerleon.nl) werkt als Enterprise Architect bij Caerleon. Naast zijn wiskunde en informatica opleiding (WO) is hij ook geschoold in de denktechnieken van Edward de Bono: lateraal denken, het buiten de kaders denken, is een onderdeel van zijn bagage waar hij graag mee bezig is.

Referenties

- Baets, W. (2002). *Wie orde zaait zal chaos oogsten*. Van Gorcum.
- Cockburn, A. (2001). *Agile Software Development*. Addison-Wesley Professional.
- Highsmith, J. (1999). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House.
- Treacy, M. & Wiersema, F. (1997). *The Discipline of Market Leaders*. Perseus Books Group.
- Wagter, R., Van den Berg, M., Luijpers, J., Van Steenberg, M. (2001). *DYA: Snelheid en samenhang in business- en ICT-architectuur*. Tutein Nolthenius.
- De Wit, F., Lommers, J. & Klinkenberg, N. (2004). *Bedrijfseigen raamwerk Ahold*. In: Informatie, november 2004. Ten Hagen Stam.

Het Speculate-Collaborate-Learn forum voor architecten: <http://www.agilearchitect.nl/>
Manifesto for Agile Software Development: <http://agilemanifesto.org/>
Declaration of Interdependence: <http://pmdoi.org/>
