



Afstemmen infrastructuur en gebruikssoftware tijdens ontwerp

Projecten volgens architectuurkaders

Medewerkers aan de Universiteit Twente onderzoeken in het Graal-project de afstemming tussen bedrijf en ict. De auteurs leggen uit hoe het Graal-raamwerk kan worden gebruikt om deze afstemming gestructureerd tot stand te brengen.

Roel Wieringa, Pascal van Eck en Henk Blanken

In het algemeen is het architectuurproces een wederzijds afstemmingsproces. Naast de ict-architecten en fysieke architecten is daarbij nog een derde architect in het spel: de bedrijfsarchitect. Ook deze architect vertaalt knelpunten en doelstellingen in de sociale wereld naar oplossingen. In eerste instantie werkt de bedrijfsarchitect uitsluitend in de sociale wereld. Echter, in onze hoogtechnologische wereld moet ook de bedrijfsarchitect multidisciplinair werken. Ten eerste zal hij of zij het ontwerp van bedrijfsoplossingen moeten afstemmen met mogelijkheden en randvoorwaarden van de symbolische en fysieke wereld. Als de ontworpen bedrijfsarchitectuur vereist dat er één gemeenschappelijk informatiesysteem voor het hele bedrijf is, maar het is te duur om zo'n systeem te realiseren, dan zal de bedrijfsarchitect een andere architectuur moeten ontwerpen. Ten tweede zijn er onderdelen van het bedrijf die in dienst staan van de fysieke of symbolische wereld, zoals de gebouwendienst en de softwarebeheersafdeling. De architecturen van die afdelingen zijn een

afgeleide van de architectuur van de fysieke en symbolische wereld.

Het Graal-raamwerk

Om het wederzijdse afstemmingsproces vorm te geven hanteren bedrijven raamwerken waarin dimensies worden genoemd waarlangs afstemming moet plaatsvinden. Ontwerpbeslissingen in de drie werelden worden dan gestructureerd door ze volgens de dimensies te classificeren. Medewerkers van de Universiteit Twente onderzoeken in het Graal-project* deze afstemmingsbeslissingen. In dit project hanteren we een raamwerk dat alle tot nu toe gevonden raamwerken generaliseert. Door de dimensies van de raamwerken die verschillende bedrijven hanteren op het Graal-raamwerk af te beelden, kunnen we de architectuurbeslissingen bij verschillende bedrijven met elkaar vergelijken. Graal is ontstaan door een uitputtende analyse van raamwerken voor systems engineering, industrieel productontwerp, software engineering en informatiesysteemontwerp (Wieringa, 1996; 2003).

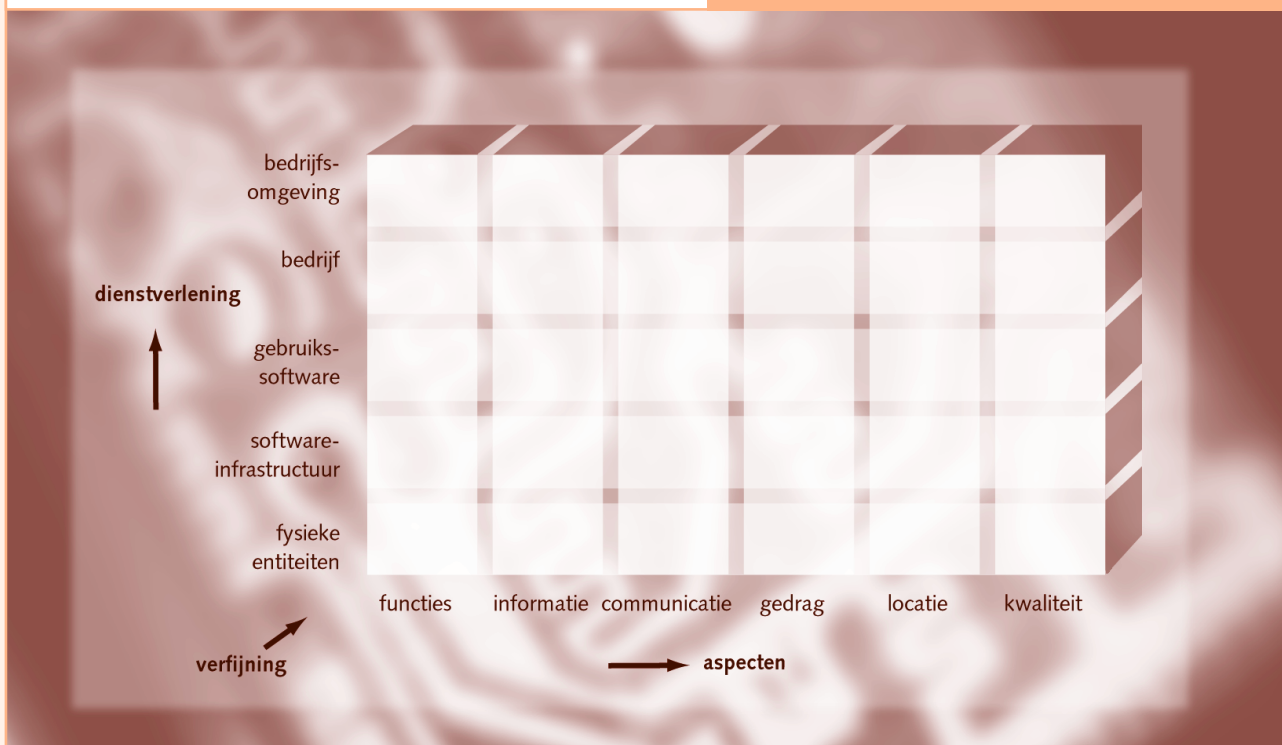
*) Graal staat voor Guidelines Regarding Architecture Alignment. Zie is.cs.utwente.nl/Graal.

Samenvatting

Om de afstemming tussen bedrijf en ict gestructureerd tot stand te brengen is het Graal-raamwerk ontwikkeld. Dit raamwerk identificeert vier dimensies waarlangs deze afstemming tot stand moet komen. De auteurs beschrijven hoe dit raamwerk wordt toegepast voor de afstemming van gebruiksoftware en infrastructuur en behandelen het spanningsveld tussen projectmatig werken en architectuurgestuurd werken.

De belangrijkste dimensies van het Graal-raamwerk

1



Het Graal-raamwerk bestaat uit vier dimensies; drie daarvan zijn in figuur 1 afgebeeld. Verticaal bestaat het raamwerk uit een aantal dienstverleningslagen die we in de een of andere vorm bij alle bedrijven zien. De onderste laag bestaat uit fysieke entiteiten. Daarboven bevinden zich twee lagen uit de symbolische wereld: software-infrastructuur en gebruiksoftware. De bovenste twee lagen betreffen de sociale wereld: bedrijf en bedrijfsomgeving. De betekenis van de lagen is dat entiteiten in een lagere laag diensten (of services) kunnen verlenen aan entiteiten uit een hoger gelegen laag.

- De fysieke laag bestaat uit gebouwen, hardware, bekabeling, draadloze verbindingen en alle andere entiteiten die een locatie in tijd en ruimte hebben en energie verbruiken. Locatie

is een aspect van fysieke entiteiten en daardoor indirect van alle andere entiteiten op hoger gelegen lagen, die allemaal van de fysieke wereld afhangen.

- De software-infrastructuur bestaat uit alle producten die algemeen beschikbare diensten aanbieden, zoals application servers, file servers, databases, workflowmanagement, kantoorsoftware, enzovoort. Infrastructuur moet aan hoge kwaliteitseisen voldoen en heeft niet één specifieke gebruikersgroep.
- Gebruikssoftware zijn producten die voor een bepaalde gebruikersgroep of een bedrijfsproces diensten aanbieden. Hier vallen sommige informatiesystemen onder, alle applicaties, presentatiesoftware, enzovoort. Alle software die specifieke gebruikerstaken ondersteunt,

zoals het uitrekenen van een offerte of het bewaren van casusgegevens in een workflow, behoort tot deze laag. Wanneer gebruiksoftware algemeen gebruikt gaat worden, zoals kantoorprogrammatuur, kan deze naar de infrastructuurlaag verhuizen.

- De bedrijfslaag bestaat uit sociale entiteiten waaruit een bedrijf bestaat, zoals werknemers en afdelingen, en andere entiteiten die uiteindelijk onder verantwoording van het topmanagement staan.
- De bedrijfsomgeving bestaat uit het deel van de sociale wereld waar het bedrijf diensten aan levert.

De lagenstructuur laat alleen de dienstverleningshiërarchie naar de klant zien. Er is een tweede dienstverleningshiërarchie die als doel heeft het bedrijf in stand te houden. In die tweede hiërarchie zijn er onderhoudsmonteurs die diensten verlenen aan de fysieke entiteiten, beheerders die ervoor zorgen dat de software blijft draaien en toeleveranciers die het bedrijf van grondstoffen voorzien. Figuur 1 is niet bedoeld als definitieve vaststelling van een universele dienstverleningshiërarchie, maar als indicatie dat de architect elementen van een dienstverleningshiërarchie ontwerpt, en dus in kaart moet brengen in welke hiërarchie die elementen een functie hebben. Het ontwerpprobleem wordt zo gestructureerd in lagen die middels een doel-middelhiërarchie gerelateerd zijn.

De tweede dimensie in het Graal-raamwerk bestaat uit *aspecten*. Een aspect is een eigenschap van een entiteit. Het belangrijkste aspect in het Graal-raamwerk is de functie van een entiteit. In de documentatie van een bedrijf en zijn software staat minimaal vermeld welke missie een bedrijfsonderdeel heeft, wat de functie van een werknemer is, of wat het doel van een software-systeem is. Een tweede belangrijk aspect is de *informatie* die wordt gemanipuleerd, en een derde belangrijk aspect bestaat uit de *communicatiekanalen* waardoor die informatie stroomt. Het *gedrag* van entiteiten is van belang op alle niveaus, vanaf de netwerkstructuur op infrastructuurniveau tot aan de bedrijfsprocessen waarmee klanten bediend worden. In een mobiele wereld wordt de *locatie* van (de fysieke realisatie van) sociale entiteiten en software steeds belangrijker. *Kwaliteitsaspecten* vinden we eveneens op alle niveaus. Zo is beveiliging een aspect van de fysieke wereld, de software-infrastructuur, applicaties en bedrijfsprocessen.

Het onderscheiden van aspecten is een belangrijk middel om de complexiteit van een architectuur te reduceren. In plaats van alle aspecten van een architectuur in één keer te ontwerpen, kan de architect zich tot enkele aspecten beperken om vervolgens later de aspectontwerpen te integreren. De derde dimensie, *verfijning*, is eveneens een belangrijke techniek om complexiteit te reduceren. Deze dimensie betreft de documentatie over de entiteiten op alle niveaus. Een document kan een entiteit abstract beschrijven, dat wil zeggen met weinig details, of heel verfijnd, dus met heel veel detail. Een abstracte beschrijving van de functies van een softwareproduct zal alleen zijn missie en belangrijkste taken omvatten. Een uiterst gedetailleerde beschrijving zal alle functies omvatten, inclusief hun triggers, precondities, input en output.

De vierde dimensie van het Graal-raamwerk, niet vertoond in het diagram, is de *fase in de levensloop* van een entiteit. Het architectuurproces produceert een continue stroom van documenten met plannen en ontwerpen van systemen in verschillende fasen van hun leven. Terwijl een bepaalde versie van het systeem wordt gebruikt, wordt een volgende versie gepland en wordt tegelijk misschien nagedacht over een vervangend systeem dat in de toekomst als opvolger moet fungeren. Die vierde dimensie hebben we nodig om in kaart te brengen over welke versie van een entiteit het gaat.

Onderzoeksresultaten

Graal doet gevalsstudies, wat betekent dat we met behulp van ons raamwerk architectuurprocessen bij bedrijven analyseren zonder in die processen in te grijpen. Na vier gevalsstudies kunnen we enkele trends herkennen in de architectuur.

Conway

Als het ontwerp van een systeem te groot is om in één hoofd te passen, zijn er meerdere ontwerpers nodig om het ontwerp te maken. Die mensen verdelen het werk op een bepaalde manier. Deze werkverdeling zal zichtbaar worden in de architectuur van het ontworpen systeem. Dit is de wet van Conway, naar de persoon die voor het eerst op dit principe wees (Conway, 1968). De wet geldt in twee richtingen: bij het ontwerpen zal de structuur van de ontwerpgroep worden weerspiegeld in de structuur van het systeem, maar bij het onderhoud zal de structuur van het systeem worden weerspiegeld in de onderhoudsgroep.



We zien de wet van Conway geïllustreerd in de architectuur van gebruikssoftware als die van de infrastructuur. Dus als bijvoorbeeld de gebruikssoftware verdeeld is in domeinen, één voor elke gebruikersgroep, dan zien we dat de beheerafdeling verantwoordelijk voor het onderhoud van de gebruikssoftware, op dezelfde manier verdeeld is in onderdelen, één voor elke gebruikersgroep. En als bijvoorbeeld de architectuur van ict-domeinen wordt aangepast, dan is dit niet alleen een technisch maar ook een organisatorisch probleem. Wie van de huidige it-managers zal welk domein onder zijn of haar hoede krijgen? Omgekeerd zal een herorganisatie van de it-afdeling die indruist tegen de bestaande ict-architectuur van een bedrijf, gedoemd zijn te mislukken, tenzij de bestaande ict-architectuur op de een of andere wijze weer in de nieuwe structuur van de it-afdeling te herkennen is. En als de formele organisatie van de it-afdeling de structuur van het beheerde systeem niet weerspiegelt, zal er een informele structuur ontstaan die die structuur wel weerspiegelt.

Afstemming gebruikssoftware

Afstemming van gebruikssoftware is het domein van bekende informatiesysteemmethoden zoals information engineering en Panfox. Met behulp van deze methoden kunnen we bedrijfsfuncties analyseren, gegevensdomeinen in kaart brengen en een lijst produceren met gewenste informatiesystemen. Dit zijn dus methoden om de architectuur van de gehele applicatie-/informatiesysteemlaag te herontwerpen. In de door ons bestudeerde organisaties vindt architectuurherontwerp echter niet plaats door een analyse van de bestaande bedrijfsfuncties, maar door een fundamentele heroriëntatie op bedrijfsdoelen en een daaruit afgeleide indeling in aandachtsgebieden van het bedrijf. Een verzekeraar identificeert bijvoorbeeld als aandachtsgebieden schadeverzekeringen, hypotheeken, zorgverzekeringen, klantcontacten, claimafhandeling enzovoort. Een dergelijke opdeling in domeinen vereist een theorie van het verzekeringswezen. Elk domein is een kennisgebied met specialisten die kennis van dat gebied hebben. De indeling wordt zo geko-

zen dat gemeenschappelijk te gebruiken diensten in aparte domeinen worden geplaatst. Pas als een indeling in domeinen gevonden is, worden per domein de bedrijfsfuncties en processen in kaart gebracht, zoals door klassieke informatiesysteemmethoden aangegeven, en wordt een gerationaliseerde applicatiearchitectuur per domein gedefinieerd.

Infrastructuurafstemming

Voor de infrastructuurafstemming wordt een soortgelijk proces gevolgd van indeling in domeinen, waarbij elk domein weer een kennisgebied is. Voorbeelden van kennisgebieden zijn operating systems, netwerken, documentscanning en workflowmanagement. Dit zijn technische kennisgebieden die ver van gebruikers of bedrijfsdoelen verwijderd zijn. Infrastructuurdomeinen zijn dus ook veel moeilijker vanuit bedrijfsdoelen of gebruikers aan te sturen. Beslissingen over infrastructuur worden gemotiveerd door een verwijzing naar bedrijfsstrategie, knelpunten, legacy software, aanwezige hardware en marktontwikkelingen in de infrastructuurdomeinen. Marktontwikkelingen lijken echter nog steeds de doorslaggevende factor te zijn, met als gevolg dat infrastructuurcomponenten bestaande uit nieuwe, 'sexy' technologie, worden toegevoegd aan een infrastructuur die reeds uit zijn voegen barst. Zoals gezegd, nieuwe technologie komt zelden in de plaats van oude technologie, maar wordt eraan toegevoegd. Het is de toegevoegde waarde van de infrastructuurarchitect om dit proces vanuit bedrijfsstrategie onder controle te houden.

Zelfs in de gevallen waarin infrastructuur en gebruikssoftware beide vanuit bedrijfsdoelen worden aangestuurd, kunnen deze processen dermate onafhankelijk van elkaar zijn dat de afstemming tussen de gebruikssoftware en de infrastructuur onderling niet goed is. De gebruikssoftware wordt aangestuurd vanuit bedrijfsdoelen, die na detaillering worden vertaald in bedrijfsprocessen, waarbij dan gebruikssoftware wordt ontwikkeld of gekocht. Als de infrastructuurlaag wordt aangestuurd vanuit bedrijfsdoelen, gebeurt dat door deze te vertalen

in abstracte infrastructuurdoelen, die dan worden geconcretiseerd in aanschafbeslissingen per infrastructuurdomein. Deze twee processen kennen onderling weinig afstemming en het resultaat kan zijn dat de infrastructuur de beoogde gebruiksoftware niet kan ondersteunen. De centrale kloof die wij waarnemen is niet die tussen de organisatie en gebruiksoftware, maar die tussen de gebruiksoftware en de infrastructuur.

Project- versus architectuurgestuurd

Een project wordt in het leven geroepen om een resultaat voor een opdrachtgever te bereiken. Die opdrachtgever is meestal een afdeling in een bedrijf en heeft zijn eigen, lokale belangen. Nadat het resultaat geproduceerd is, wordt het project ontbonden. Een architectuur daarentegen wordt in het leven geroepen om een resultaat voor het hele bedrijf te produceren, en nadat de architectuur geproduceerd is, moet deze bewaard blijven. Architectuur is geen project, maar een proces. Dit levert een spanning op tussen projectgestuurd en architectuurgestuurd werken. Waar een projectleider beslissingen neemt op basis van lokale overwegingen, die op het belang van de opdrachtgever gericht zijn, zal een architect beslissingen nemen op basis van globale overwegingen, die op het belang van het hele bedrijf gericht zijn. Waar een projectleider een component zal willen ontwikkelen die precies past bij de behoefte van zijn klant, zal een architect eisen dat algemeen beschikbare componenten hergebruikt worden, zelfs als die niet precies aan de behoefte van dit project voldoen. Er lijkt een natuurwet te zijn in de vorm $P \times A < W$, waarbij P de mate van projectgestuurd werken is, A de mate van architectuurgestuurd werken en W een onbekende constante, die we een willekeurige naam gegeven hebben. Dit vraagt om een centrale toetsing van projecten door te eisen dat ze binnen gestelde architectuurkaders blijven.

Conclusies

Het Graal-raamwerk biedt handvatten voor de analyse van ict-architectuur van zowel individuele ict-systemen als een gehele laag van systemen.

Het laat de samenhang tussen fysieke architectuur en ict-architectuur zien en stelt ons in staat beslissingen over architectuur te structureren. Bij toepassing van het raamwerk op concrete projecten bij grote dienstverlenende bedrijven blijkt dat er een nauwe samenhang is tussen de architectuur van de ict-afdeling en de architectuur van de ict zelf. De ict-architectuur kent een kloof tussen de architectuur van de infrastructuur en de architectuur van de gebruiksoftware, die kan worden verkleind door een betere afstemming tot stand te brengen tussen het infrastructuurontwerpproces en het ontwerpproces van gebruiksoftware. Een tweede kloof die vaak voorkomt is die tussen projectgestuurd en architectuurgestuurd werken. Deze kloof kan alleen maar worden verkleind door vanuit een centrale sturing architectuurkaders aan projecten op te leggen.

Literatuur

- Conway, M.E. (1968). How do Committees invent? *Datamation*, 14(4), april 1968, pp. 28-31.
- Sanden, W. van der & B. Sturm (1997). *Informatiearchitectuur: De infrastructurale benadering*. Panfox.
- Wieringa, R.J. (1996). *Requirements Engineering: Frameworks for Understanding*. Wiley.
- Wieringa, R.J. (2003). *Design Methods for Reactive Systems: Yourdon, StateMate and the UML*. Morgan Kaufmann.
- Wieringa, R.J. e.a. (2003). Aligning Application Architecture to the Business Context. Conference on Advanced Information System Engineering (CAiSE 03). Springer, pp. 209-225.

Roel Wieringa

is hoogleraar Informatiesystemen aan de Universiteit Twente. E-mail: roelw@cs.utwente.nl.

Pascal van Eck

is universitair docent Informatiesystemen aan de Universiteit Twente. E-mail: patveck@cs.utwente.nl.

Henk Blanken

is universitair hoofddocent Databases aan de Universiteit Twente. E-mail: blanken@cs.utwente.nl.