Een Brug Te Ver?

De analogie tussen de fysieke en digitale bouwwereld

Erik Philippus
Improvemen**T**
erik.philippus@improvement-services.nl

Te pas en te onpas worden projecten in de civiele bouwwereld als analogie opgevoerd om (grote) software ontwikkel trajecten te verduidelijken (en zo mogelijk te verbeteren). Dat deze vergelijking echter met een gepaste korrel zout genomen moet worden, bewijst nadere bestudering van het artikel 'De bouw als voorbeeld voor ICT' van Tobias Kuipers en Gerjon de Vries, verschenen in het december nummer van 'Management Scope'. Het blijkt dat bij het al te gretig overnemen van concepten uit de fysieke bouwwereld, er nog wel wat addertjes onder het gras kunnen zitten, zoals:

Wat zijn nu precies de cruciale verschillen tussen de fysieke en digitale bouwwereld, en hoe zinnig is de vergelijking met de fysieke bouwwereld in de praktijk? Wat is nu precies een 'software bouwmanager', wat is de plek, de rol en hoe zit het met de verantwoordelijkheid en het risico? En hoe zit het met zijn onafhankelijkheid? Ofwel: door wie wordt de software bouwmanager betaald? Is de software bouwmanager in staat om voldoende aandacht te schenken aan een zeer belangrijke invloed op het succes van een software traject: de menselijke factor? En ten slotte: waar is in dit verhaal de IT architect?

Op twee benen

Ervan uitgaande dat de problematiek bij het managen van grote ICT projecten direct gerelateerd is aan het scala van onzekerheden bij het bouwen van software systemen, wordt gesteld dat de tijd rijp is voor een nieuwe vorm van ICT dienstverlening, namelijk 'software-bouwmanagement'. Deze term is overgenomen uit de bouwwereld, waarin bouwmanagement al jaren geleden als aparte discipline is ingesteld, met als doel de beheersbaarheid en voorspelbaarheid van complexe bouwopdrachten te verbeteren. Immers, de bouw van een nieuw pand kost de meeste opdrachtgever behoorlijk wat hoofdbrekens en frustraties. Maar al te vaak lopen de kosten uit de hand, worden gebouwen te laat opgeleverd of blijkt het pand gedeeltelijk te voldoen aan de behoeften.

Natuurlijk - de vergelijking met de meeste software projecten is treffend. De kennelijke overeenkomst in symptomen betekent echter niet dat ook de oplossingsrichting naadloos kan worden overgenomen van de bouwwereld. Hoewel het voorgestelde concept van software bouwmanagement zeker positieve aspecten bevat, hinkt het verhaal van Kuipers en de Vries op verschillende plaatsen toch op twee benen.

In het artikel wordt als belangrijkste oorzaak voor de problematiek gewezen naar een software ontwikkelproces, gebaseerd op de werkwijze in de fysieke bouw. Achtereenvolgens worden de fases van specificatie, ontwerp, bouw en test doorlopen, afgerond door ingebruikname en opleiding van gebruikers.

Zoals de auteurs terecht signaleren, kost het een hoop tijd om een dergelijk formeel gefaseerd proces te volgen, en kunnen bovendien de stappen onafhankelijk van elkaar mislukken.

Vervolgens worden drie nadelige gevolgen van deze (in softwarekringen overbekende) 'waterval' methodiek geschetst. Hieruit blijkt dat deze werkmethode, afgeleid van de fysieke bouw, grote problemen veroorzaakt bij het managen van grote ICT projecten. Dit geeft al aan dat er kennelijk een essentieel verschil is tussen de fysieke en de digitale bouwwereld. Daarom is het opmerkelijk dat er vervolgens een oplossing wordt aangedragen, nota bene afkomstig uit diezelfde bouwwereld.

Verschillen in focus

Want er zijn nogal wat cruciale, in deze context relevante verschillen tussen het bouwen van huizen en van software. Zo wordt in de fysieke bouwwereld het overgrote deel van de inspanning geleverd in de constructiefase: slechts circa 10% van de totale kosten wordt gespendeerd aan ontwerp activiteiten.

Een traditionele bouwmanager houdt zich dan ook nauwelijks inhoudelijk bezig met het ontwerp, maar richt de aandacht op de constructie: het oplossen van problemen en storingen tijdens de bouw, het bewaken van de voortgang en opstellen van prognoses, het terugkoppelen van informatie, etc. Kortom: de bouwmanager zal met grote regelmaat afdalen in de bouwput van projecten.

Kijken we naar het software creatie proces, dan ligt dit beeld totaal anders: hier is juist de designfase verreweg het meest arbeidsintensieve onderdeel, zo'n 85%. Zelfs als alle testactiviteiten worden geschaard onder de noemer 'constructie', blijft ontwerpen minstens de helft van het werk. Bovendien is er een tendens om typische software constructie activiteiten zoveel mogelijk te automatiseren, b.v. door codegeneratie.

Een software bouwmanager die zich enkel beperkt tot de constructie, zou niet veel verschillen van de reeds bekende, traditionele 'build manager'. Kortom: de focus van de software bouwmanager is fundamenteel anders dan die van de bouwmanager in de civiele bouwwereld. Op zich is dat verschil nog niet zo bezwaarlijk, maar dan moeten wel de elementaire verschillen tussen de constructiefase en de ontwerpfase worden meegenomen. Ontwerpen is in essentie een creatieve activiteit, waardoor het – zeker in vergelijking met de constructie fase - moeilijk te plannen is. Ontwerpers en bouwers beschikken dan ook over verschillende talenten en competenties, en zullen ook op een andere wijze aangestuurd moeten/willen worden.

In het artikel wordt een rol van de software bouwmanager geschetst, waarvan het twijfelachtig is of die zonder problemen wordt geaccepteerd door een software ontwikkelteam. Het voorschrijven van universele softwaretechnische conventies is op zich een goede zaak, maar het is zeer de vraag of een ervaren, hoog opgeleide ICT-er bereid is om dat terrein prijs te geven. De ervaring leert dat de eigengereidheid van software engineers niet alleen legendarisch is, maar ook bijzonder weerbarstig – vaak tot wanhoop van de opdrachtgever…

De menselijke factor

Hierbij stuiten we op een aspect van software ontwikkeling waaraan in het algemeen – en in het bijzonder ook in het artikel – wordt voorbijgegaan, namelijk de verregaande invloed van de menselijke factor. Uiteraard willen mensen (en zeker managers) graag voorspelbare projecten – maar het is een gegeven dat het juist de mens is die onvoorspelbaarheid brengt in elk ontwikkeltraject. Er is geen technologie, ontwikkelproces of organisatiestructuur, hoe geavanceerd en doordacht ook, in staat om de variabiliteit van mensen in een projectteam uit te schakelen. En op de keper beschouwd is dat maar goed ook.

Het is een gegeven dat mensen neigen tot inconsistentie. Keer op keer blijkt dat het voor de mens bijzonder moeilijk is om een bepaalde activiteit dagelijks op identieke wijze uit te voeren – dat is haast net zo moeilijk als het veranderen van een gewoonte. En als het gaat om het maken van software, dan hebben we te maken met nogal wat gewoontes, om niet te spreken van opmerkelijke, diep gewortelde gedragspatronen…

Toch begint in ICT kringen langzaam maar zeker het besef door te dringen, dat de invloed van menselijk gedrag een cruciale factor is in het voorspellen van de voortgang en de uitkomst van een software project. Het succes van de mensgerichte aanpak van de laatste jaren sterk in opkomst zijnde Agile ontwikkelmethoden is hiervan een sprekend voorbeeld.

De moraal: menselijk gedrag vormt een niet te onderschatten factor in het werkveld van de software bouwmanager. Het slechts lijdzaam beamen hiervan is onvoldoende: het inventariseren van relevante gedragspatronen, het herkennen van signalen en drijfveren en het onderkennen van (tegenstrijdige) belangen, en hoe hier adequaat mee om te gaan, vormen allen een noodzakelijk onderdeel van het uitgebreide pallet van vaardigheden die de succesvolle software bouwmanager in z'n rugzak dient te hebben.

Wie draagt het risico?

De auteurs slaan de spijker op de kop met het signaleren van de weerbarstige 'uurtje-factuurtje' mentaliteit van vele ICT-ers. Ook nu nog werken veel softwarebouwers nog voornamelijk op basis van nacalculatie, en zijn derhalve nauwelijks gebaat bij het optimaliseren van het software ontwikkelproces. Aangezien het grootste deel van de kosten van een software systeem zitten in het onderhoud, zou prioriteit geven aan het maken van software van hoge kwaliteit leiden tot inkomstenderving van de software bouwer/onderhouder.

Deze pijnlijke belangentegenstelling lijkt mede verantwoordelijk te zijn voor de trage gang naar volwassenheid in de software industrie.

Terecht wijzen de auteurs op het opmerkelijke feit dat softwarebouwers nauwelijks worden gedwongen om aan risicomanagement te doen. Afwijkend van de fysieke bouwwereld, is het bij het maken van software nog steeds gebruikelijk om het risico zoveel mogelijk bij de opdrachtgever neer te leggen.

Dit roept echter wel meteen de vraag op wat nu de verantwoordelijkheid de software bouwmanager is. Hebben we het alleen over een toezichthouder, of over een mede-drager van het projectrisico? Direct hieraan gerelateerd: wie gaat eigenlijk de software bouwmanager betalen? Zo op het eerste gezicht zal dat (noodgedwongen!) de opdrachtgever zijn, en in dat geval is er geen sprake van onafhankelijkheid. Hoewel een loffelijk streven, lijkt de tijd vooralsnog niet rijp voor het instellen van een - door alle partijen erkende – regulerende en toezichthoudende instantie. Voordat het zover is, zal er toch eerst een algemene classificatie van software systemen moeten worden opgesteld, met bijbehorende productievoorschriften en productnormen.

Over communicatie gesproken

Naast de inherente onzekerheden bij het bouwen van softwaresystemen, ligt gebrekkige (of zelfs ontbrekende) communicatie tussen opdrachtgevers, gebruikers en ontwikkelaars aan de basis van veel problemen. Ontwikkelaars hebben gewoonlijk veel moeite om in de huid te kruipen van toekomstige gebruikers van het te ontwikkelen softwaresysteem, opdrachtgevers zijn vaak niet goed in staat hun business doelstellingen en het daarvan afgeleide pakket van eisen duidelijk te definiëren - kortom: verwarring en misverstanden tussen belanghebbenden zijn aan de orde van de dag.

En alsof dat nog niet genoeg uitdaging biedt, hebben we zonder uitzondering ook nog te maken met veranderende requirements. Niet voor niets wordt gezegd: “Lopen op water en het omgaan met requirements is eenvoudig, zolang beide maar bevroren zijn”.

De software bouwmanager lijkt tussen de opdrachtgever en opdrachtnemer in te staan, en dit betekent dus een extra schakel in een al bestaande, maar moeizame samenwerking.

Dat kan natuurlijk goed uitvallen, maar dat is zeker geen vanzelfsprekendheid. In ieder geval staat de software bouwmanager op een cruciale plek als het gaat om de uitwisseling van informatie tussen partijen.

Een dergelijke positie stelt zware eisen aan ervaring en vaardigheden van de bouwmanager op gebied van communicatie, inlevingsvermogen en onderhandelen. Over de invloed van de menselijke factor gesproken…

Meten is weten

De functie van bouwmanager valt en staat met de meetbaarheid van alle relevante eigenschappen van het opgeleverde systeem. Alleen door te meten kan de bouwmanager te weten komen of het geleverde product aan de gestelde eisen voldoet. In het artikel wordt gesteld dat er momenteel mogelijkheden zijn om software 'op de juiste manier inzichtelijk te maken'. Inderdaad zijn er tegenwoordige geavanceerde gereedschappen om aan software te kunnen meten, maar dit geldt zeker niet voor alle relevante aspecten van software systemen. Kwaliteitsfactoren zoals onderhoudbaarheid en aanpasbaarheid zijn slechts tot zekere hoogte af te leiden uit aspecten van het ontwerp of de code.

In ieder geval kan de onderhoudbaarheid van software niet met dezelfde mate van nauwkeurigheid worden bepaald als b.v. de levensduur van een bepaalde coating.

Ook dit is een cruciaal verschil tussen de fysieke en digitale bouwwereld.

Een opdrachtgever die vraag om onderhoudsvrije software geeft blijk van weinig realiteitsgehalte. Ook al gooit een afgeleverd systeem de hoogste ogen wat betreft onderhoudbaarheid, dan nog kunnen de omstandigheden zo snel en onvoorspelbaar wijzigen dat de beloofde aanpasbaarheid maar deels kan worden waargemaakt in de toekomst.

Hoe het ook zij, sommige kwaliteitsaspecten zijn goed te meten, andere een stuk minder goed. Het gevaar dat hierbij op de loer ligt, is dat de software leverancier zich vooral gaat richten op het behalen van een hoge score op alleen de meetbare aspecten. Hierdoor kan bij de opdrachtgever een scheef beeld ontstaan over de overall kwaliteit van het geleverde systeem – iets om als bouwmanager terdege rekening mee te houden.

Dit laat onverlet dat productkwaliteit bij het maken van software nog steeds een onderbelichte rol speelt, terwijl juist kwaliteit de smeerolie kan zijn in de communicatie tussen opdrachtgever en softwarebouwer.

Multidisciplinaire ontwikkeling

Dat het bouwen van software zich steeds vaker afspeelt in een multidisciplinaire ontwikkelomgeving, komt helaas in het artikel niet aan de orde. Toch is dit juist een aspect van software ontwikkeling waarbij de metafoor van de fysieke bouwwereld van toepassing lijkt.

De in sneltreinvaart toenemende miniaturisering van processorkracht maakt de weg vrij voor zeer complexe embedded software toepassingen. Hierbij zal in toenemende mate een intensieve samenwerking met andere engineering disciplines nodig zijn, b.v. om geavanceerde Systems-On-Chip (SOC) toepassingen mogelijk te maken. Maar ook grotere apparaten worden voorzien van steeds complexere software, en de integratieproblematiek groeit nu al menige machinebouwer boven het hoofd.

In ieder geval heeft het vervagen van de traditionele scheidingswanden tussen software en disciplines zoals elektronica, fysica, optica, chemie, etc. ingrijpende gevolgen voor de het software ontwikkeltraject. Systeemontwikkeling zal steeds minder betrekking hebben op 'software-only' systemen, en dientengevolge zal het bouwen van (technische) software steeds meer een multidisciplinair karakter krijgen. Verschillende engineering disciplines komen bij elkaar in een complex krachtenveld, en juist hierin zou onafhankelijke software bouwmanager een grote toegevoegde waarde kunnen hebben. Een belangrijke taak zou dan liggen op het gebied van integratie, in alle fases van het project. Uiteraard dient een dergelijke manager van diverse markten thuis te zijn, hoewel een stevige achtergrond in software engineering het meest geschikt lijkt voor het succesvol uitoefenen van een dergelijke, uitdagende functie.

Waar is de architect?

In de fysieke bouwwereld komt bouwmanagement normaliter pas ter sprake als het pakket van eisen op tafel ligt, en op basis daarvan een ontwerp is gemaakt. De bouwmanager zal zonder al te veel problemen het stokje kunnen overnemen van de architect. Kijken we echter naar de digitale bouwwereld, dan zien we een veel grotere mate van onzekerheid met betrekking tot de specificaties van het op te leveren systeem. Dit heeft grote gevolgen voor het (politieke) besluitvormingsproces, en daarmee ook voor de betrokkenheid van de architect. Kortom: de grote afwezige in het verhaal over de software bouwmanager is de architect.

Overigens roept deze hele discussie over de software bouwmanager herinneringen op aan het pleidooi voor onafhankelijke IT architecten, met Jaap van Rees als belangrijkste pleitbezorger voor de functiescheiding tussen architect en softwarebouwer.

Is er sprake van oude wijn in nieuwe zakken, of stapt de software bouwmanager in het gat dat is achtergelaten door het droombeeld van de onafhankelijke informatie architect? De toekomst zal het leren.