Software maken blijft mensenwerk

De integrale benadering volgens het TOP concept

Erik Philippus

Het maken van software gaat zelden van een leien dakje. Veel producenten van software-intensieve apparaten willen de kwaliteit van de afgeleverde producten structureel verhogen. De daarvoor benodigde investering in menskracht kan vaak niet worden gedaan, omdat een aanzienlijk deel van de engineers druk bezig is om de zaak bij de klant draaiend te houden. In de meeste ontwikkeltrajecten is nauwelijks tijd voor bezinning. Er wordt zo druk gezaagd, dat er geen tijd is om de zaag te slijpen[1]. Verhoging van de productiviteit van het ontwikkelteam verschaft ruimte om uit die spiraal te komen. Het hier geïntroduceerde *TOP concept* helpt daarbij.

De eindgebruiker uit beeld

Bij het uitgroeien van het vak van softwaremaken tot een engineering discipline dreigt de eindgebruiker grotendeels uit het beeld te verdwijnen als direct belanghebbende. Een 'verstokte' software engineer is niet gewend in de huid te kruipen van de eindgebruiker, en heeft bovendien de neiging om op de eerste plaats de eigen problemen op te lossen, niet die van de eindgebruikers[2]. De lokkende mogelijkheden van de softwaretechnologie verdringen de wensen en de beleving van de gebruikers naar het tweede plan. Deze ontwikkeling heeft verregaande consequenties voor het software ontwikkeltraject.

Het prominent in beeld brengen van de eindgebruiker als mens van vlees en bloed maakt het vak van software engineering volwassen. Het uit het oog verliezen van 'de mens' als invloedrijke en bepalende dimensie heeft niet alleen een negatieve invloed op het ontwikkeltraject, maar ook schadelijke gevolgen voor het eindproduct. Onvoldoende begrip van wat de opdrachtgever feitelijk voor ogen staat, gecombineerd met tekortschietend inlevingsvermogen ten aanzien van de toekomstige gebruikers, staat aan de basis van veel problemen rond ontwikkeling, invoering en gebruik van software intensieve systemen.

De belangrijkste productiviteitsfactor

Het creëren van een gemeenschappelijke visie tussen eindgebruiker, opdrachtgever en ontwikkelaars is verre van eenvoudig. Op het bordje van de softwareproducent ligt het omzetten van het pakket van eisen in een maakbare oplossing, die vervolgens getoetst kan worden aan de belangen van de (soms wispelturige of ongeduldige) opdrachtgever. Niet lang geleden werd softwaretechnologie gezien als *de* oplossing voor alle problemen. Maar alle technologische vooruitgang ten spijt, kwamen software projecten toch nog regelmatig in de problemen. Hierdoor kwam er steeds meer aandacht voor de procesmatige kant. Een voorbeeld hiervan is het Rational Unified Process, bedoeld om steeds complexe ontwikkeltrajecten onder controle te houden. Het procesdenken nam een hoge vlucht onder invloed van de Capability Maturity Modellen.

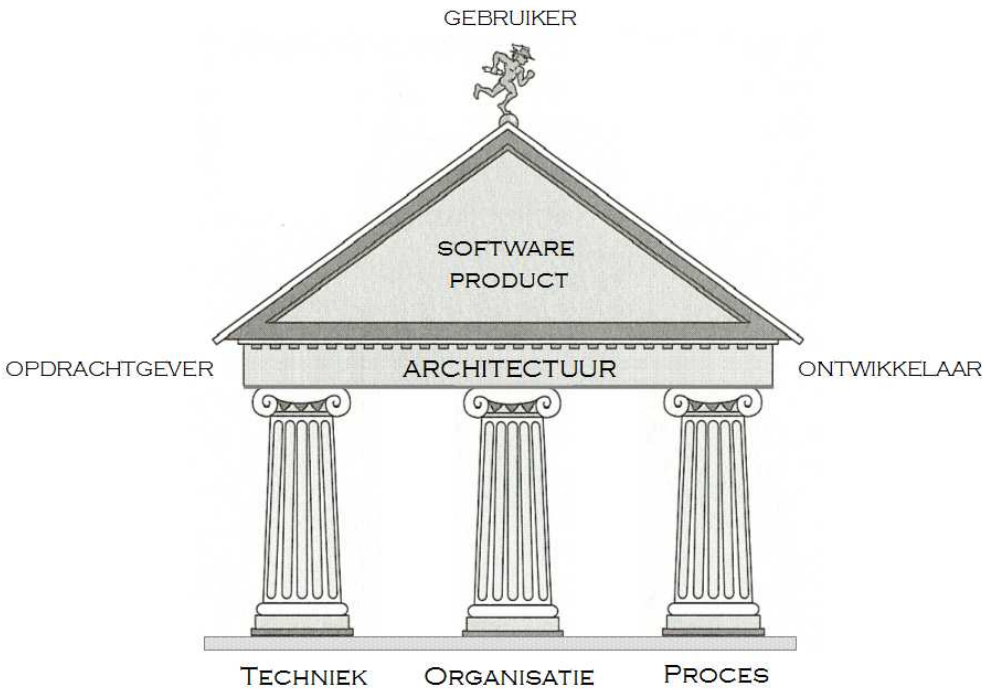
Toch laten de meeste industriële software ontwikkeltrajecten zich nog steeds het beste omschrijven als een lappendeken. Er wordt gewerkt aan een optimale integratie van gereedschappen en processen, met de metafoor van de 'software fabriek' als leidraad. Helaas wordt hierbij over het hoofd gezien dat persoonlijke eigenschappen en vaardigheden van de individuele software ontwikkelaars altijd een doorslaggevende rol zullen blijven spelen. Het maken van software is en blijft een mensenzaak. Menselijk gedrag laat zich niet egaliseren of anonimiseren door techniek of proces[3].

Dit is geen nieuw gezichtspunt. In feite zat Barry Boehm al in de 80-er jaren op hetzelfde spoor toen hij de invloed van diverse factoren op de doorlooptijd van software ontwikkeltrajecten in kaart bracht. Productiviteit bleek samen te hangen met de complexiteit van producten, de volwassenheid van de ontwikkelprocessen, maar op de eerste plaats met relevante competenties van de betrokken ontwikkelaars[4]. Keer op keer blijkt dat individuele ontwikkelaars, maar ook complete ontwikkelteams een opmerkelijk groot verschil tot een factor 10 in productiviteit kunnen vertonen[5]. Kortom: de menselijke factor heeft meer invloed op productiviteit en software kwaliteit dan enige andere factor.

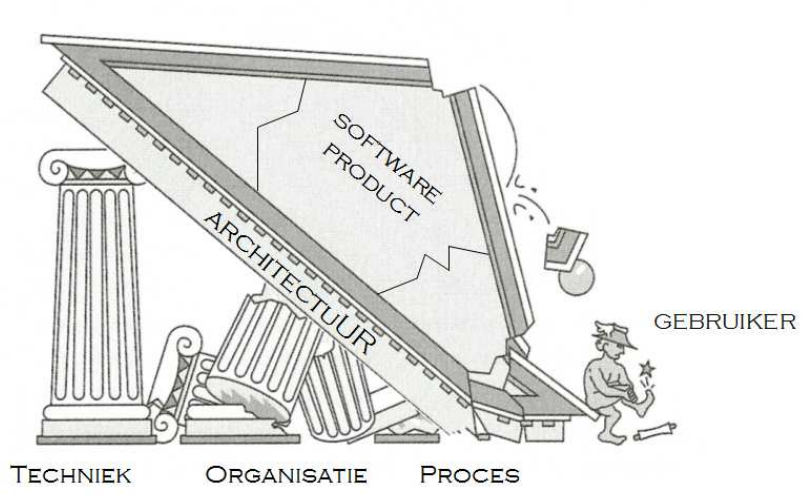
Ook de organisatiecultuur kan een onverwacht effect hebben op de productiviteit van individuele ontwikkelaars, zowel in positieve als negatieve zin[6]. Alistair Cockburn, die aan de wieg heeft gestaan van de Agile beweging, beschouwt mensen als 'niet-lineaire, eerste-orde elementen in software ontwikkeling'[7]. Ook Martin Fowler ziet ontwikkelaars niet als 'plug-and-play objecten', en stelt bovendien dat ontwikkelgereedschap en processen zich dienen aan te passen aan de ontwikkelaar, in plaats van andersom[8].

Een kwestie van balans

Toch wordt er op ons vakgebied relatief weinig aandacht geschonken aan de factor die het grootste potentieel tot productiviteitsverbetering bezit: 'de mens'. Waarschijnlijk voelen we ons als techneuten niet gemakkelijk op het gladde ijs van menselijk gedrag. Toch getuigt het van (vakinhoudelijke) volwassenheid om – zonder het belang van technologie of processen uit het oog te verliezen – de technische bril eens af te zetten. Cruciaal daarbij is om vast te stellen om welke zaken het dan gaat, hoe maak je deze zichtbaar en concreet, en wat is de meetbare invloed ervan op het traject en/of het eindproduct?[9]. Als kapstok voor deze discussie dient het z.g. **TOP concept**.



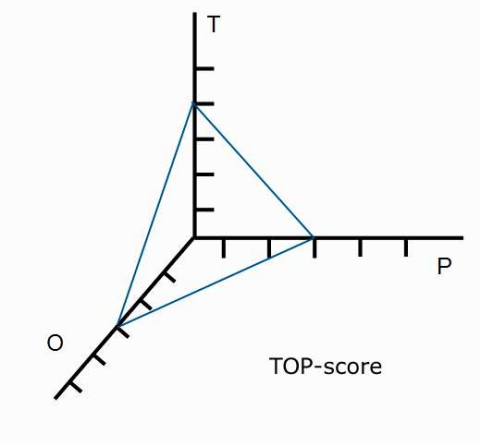
De ruggengraat van dit concept is de software architectuur, die de brug slaat tussen de opdrachtgever en de ontwikkelaar. De architectuur en de daarop gebaseerde softwareproducten worden gedragen door drie gelijkwaardige peilers, te weten **T**echniek, **O**rganisatie en **P**roces. Uitgangspunt is dat deze factoren in onderlinge samenhang de doelmatigheid van een ontwikkeltraject bepalen. Om te voorkomen dat er 'zand tussen de tandwielen komt, dienen belangrijke beslissingen in software ontwikkel projecten genomen te worden met inachtneming van de juiste balans tussen deze drie dimensies.



Het bereiken van een zo hoog mogelijke score op één aspect is ineffectief. Er ontstaat pas een evenwichtig bouwwerk als de drie peilers ongeveer even hoog zijn. Zo heeft het nauwelijks zin om geavanceerd ontwikkelgereedschap naar binnen te kruien als het team de afgesproken werkwijze niet volgt. Het is water naar de zee dragen om een zwaar ontwikkelproces op te dringen aan een team dat de vereiste technische know-how mist. En iedereen die te maken heeft met problematiek rond platformontwikkeling of legacy systemen, weet wat er kan gebeuren als er een blinde vlek is voor de invloed van niet-technische factoren.

Meten is weten

Het TOP concept heeft als doel het inzichtelijk en tastbaar maken van de dimensies **T**echniek, **O**rganisatie en **P**roces. Zonder meetbare doelen is elke verbetering een toevalstreffer. Als het gaat om productiviteitsverbetering, zijn er verschillende 'knoppen' die de manager kan bedienen. De z.g. **TOP score** geeft een aanwijzing welke 'knop' het meeste effect zal sorteren.



De TOP score is een momentopname van invloedrijke factoren op de productiviteit. Het is de samengestelde score op de drie dimensies, b.v. '4-3-3'. De score fungeert als 'nulmeting' voor een ontwikkel- of verbetertraject. Een herhaalde meting geeft inzicht of het project zich beweegt in de richting van de gestelde doelen. De TOP score kan worden bepaald middels een specialistische assessment, bestaande uit (code-)inspecties, effectmetingen, reviews, interviews, checklijsten, en etnografische methoden.

De globale betekenis van de assen is als volgt:

Techniek
Hiermee wordt gedoeld op inhoudelijk vakmanschap. De score is gerelateerd aan het vermogen van het team om het vereiste product te realiseren in technische zin. Van belang is hierbij het (cumulatieve) niveau van ervaring en kennis met betrekking tot de toegepaste ontwikkelomgeving. De kwaliteit van deze omgeving zelf is een belangrijke voorwaardenscheppende factor. Naast opleiding en ervaring, laat het technisch vakmanschap zich tot op zekere hoogte aflezen aan de kwaliteit van bestaande code, designs en documentatie.

Organisatie
Hierbij gaat het om factoren die mede bepalend zijn voor de doelmatigheid van de ingezette methoden, technieken en tools. Hier komt dus de menselijke factor in beeld. Naast de 'niet-technische' vaardigheden van software ontwikkelaars, wordt hierbij ook gedoeld op zaken zoals bedrijfscultuur, (informeel) organigram, wijze van aansturing en evaluatie, bevoegdheden, etc. Hierbij kan het gaan om bekrachtigende, positieve invloeden, maar ook om belemmerende, negatieve gedrags-patronen[10].

Proces
Hiermee wordt aangegeven in hoeverre de afgesproken ontwikkelprocessen wordt gevolgd. Daarnaast is de kwaliteit en meerwaarde van de toegepaste processen zelf ook een graadmeter. Bieden de gekozen processen de ontwikkelaars optimale ondersteuning bij hun primaire taken, of is het 'procesdenken' doorgeslagen?

Conclusie
Het TOP concept is gebaseerd op een mensgerichte benadering van software ontwikkeling, waarbij een plaats is ingeruimd voor de noden van zowel gebruikers als ontwikkelaars. Uitgangspunt is dat software systemen afgestemd dienen te zijn op eindgebruikers in plaats van andersom[11]. Het TOP concept omvat weliswaar geen ontwerprichtlijnen om dat tot stand te brengen, maar geeft wel een handvat voor het creëren van de juiste omstandigheden.
Hiermee wordt tevens voorkomen dat de ontwikkelomgeving een contraproductief keurslijf wordt voor het ontwikkelteam. Deze benadering schept de basis voor een significante verhoging van de (kwantitatieve en kwalitatieve) output van ontwikkelaars. Doordat de gemaakte systemen beter aansluiten bij de verwachtingen van alle belanghebbenden neemt het projectrendement en de productkwaliteit toe, en kan er aanzienlijk worden bespaard op rework en onderhoud. Het TOP concept is complementair aan de Agile methode, die eveneens wordt gekenmerkt door een mensgerichte aanpak en een intensieve samenwerking tussen alle partijen.

Verwijzingen

1. Steven Covey, *The Seven Habits of Highly Effective People,* Fireside, 1990

2. Alan Cooper (interview), *Why We Don't Build Software for Users*, dec.2002
   www.ftponline.com/vsm/2002_12/online/the

3. Alistair Cockburn en Jim Highsmith, *Agile Software Development: The People Factor*, Computer, pp 131-133, November 2001

4. B.W. Boehm, *Software Engineering Economics*, Prentice-Hall, 1981

5. Steve McConnell, *Rapid Development*, Microsoft Press, 1996
   (tevens inspiratiebron voor de twee illustraties van het TOP-model)

6. Tom DeMarco en Timoty Lister, *Peopleware, Productive Projects and Teams*, Dorset House Publishing Co, 1999

7. Alistair Cockburn, *Characterizing People as Non-Linear, First-Order Components in Software Development*,
   http://alistair.cockburn.us/crystal/articles/cpanfocisd/characterizingpeopleasnonlinear.html

8. Martin Fowler, *The New Methodology*,
   http://www.martinfowler.com/articles/newMethodology.htmlSoftware

9. Quantifying Soft Factors, IEEE Software, December 2001
   http://www.stevemcconnell.com/ieeesoftware/eic14.htm

10. David M. Dikel, David Kane en James R. Wilson, *Architecture: Organizational Principles and Patterns*, Prentice Hall, 2001, ISBN 0-13-029032-7

11. Menselijke Maat en IT, www.it4humans.org