

“Betrouwbare software op Tijd binnen Budget”

door Ir.J.A.Sassenburg, Alert Automation Services b.v.

Inleiding

Van eind tachtiger jaren af werd duidelijk dat toepassing van technologische hulpmiddelen in onze samenleving een enorme vlucht zou gaan nemen. Deze trend kenmerkt zich onder meer door revolutionaire ontwikkelingen in de software industrie. Echter steeds grotere problemen worden ervaren met het beheersen van projecten. Budgetten worden fors overschreden, levertijden worden niet gehaald, het uiteindelijke produkt blijkt niet aan verwachtingen te voldoen en bovendien veel fouten te bevatten. Managers zijn vaak radeloos en vragen zich af hoe deze problemen kunnen worden opgelost. In dit artikel worden de belangrijkste ontwikkelingen om te komen tot betrouwbare software op tijd binnen budget geschetst.

De software crisis

De ontwikkelingen binnen de software industrie zijn stormachtig. Naar verwachting zal in 'embedded software' - software ingebed in produkten als televisie, audio, telefoon, medische apparaten, telecommunicatie - de omzet in de komende jaren sterk toenemen. Binnen Europa een groei van 2 naar 20 miljard gulden, zo luiden voorspellingen. Haaks op deze ontwikkelingen is er binnen ondernemingen een nimmer aflatende druk om te komen tot:

- het verkorten van doorlooptijden van projecten door efficiënter te werken, zodat produkten sneller op de markt kunnen worden geïntroduceerd;
- het verhogen de betrouwbaarheid van het eindprodukt;
- het verbeteren van klantgerichtheid door vooraf gemaakte zo exact mogelijke afspraken ten aanzien van functionaliteit, kwaliteit, kosten en leveringstijden na te komen.

De spanningsvelden die zo ontstaan, zijn niet nieuw. In het verleden zijn reeds tal van pogingen ondernomen een gepast antwoord te vinden. Men zocht dit onder meer in het toepassen van geavanceerdere methoden en technieken. Resultaten waren echter teleurstellend. Eind jaren tachtig ontstond ineens overdreven aandacht voor de ISO-9001 standaard met bijbehorende ISO/9000-3 richtlijn. Veel organisaties hebben getracht zo snel mogelijk te worden gecertificeerd. Dit is vaak doel op zich geworden en dat certificaat wordt voornamelijk als commercieel visitekaartje gebruikt. Wat dan?

A. Procesbenadering (Ref1)

Zelfs in ongedisciplineerd werkende organisaties komt men van tijd tot tijd projecten tegen, die uiterst succesvol zijn verlopen. Het succes van dergelijke projecten is echter meestal toe te schrijven aan heroïsche inspanningen van een projectteam in plaats van het volgen van een voor de organisatie vastgelegde gedisciplineerde werkwijze. Door afwezigheid van een duidelijk vastgelegd proces zijn toekomstige resultaten volledig afhankelijk van het beschikbaar zijn van diezelfde mensen voor een volgend project. Dit kan geen basis voor continuïteit en verbetering op lange termijn. Dit kan alleen worden bereikt door het voor de hele organisatie vastleggen en voortdurend verbeteren van die processen die bij software ontwikkeling een rol spelen.

In 1986 is het Software Engineering Institute (Pittsburgh/USA), destijds onder leiding van Watts S. Humphrey, begonnen met ontwikkeling van een 'process maturity framework' om organisaties te helpen hun software ontwikkelingsprocessen te verbeteren. Na jaren ervaringen met dit raamwerk, middels onderzoeken bij vele bedrijven te hebben opgedaan, leidde dit in 1991 tot vrijgave van het **Capability Maturity Model**. Het CMM maakt onderscheid tussen een vijftal niveaus van volwassenheid, waarop een organisatie zich kan bevinden.

De verschillende volwassenheidsniveaus laten zich als volgt karakteriseren:

Niveau 1: “Initial”

Het software ontwikkelingsproces verloopt ad hoc en soms chaotisch (met name vlak voor het moment van verwachte oplevering). Slechts weinig processen zijn goed gedefinieerd en

behaalde succesvolle resultaten zijn sterk afhankelijk van kwaliteiten van individuele leden van een projectteam.

Niveau 2: "Repeatable"

Processen die betrekking hebben op projectmanagement zijn vastgelegd en ingevoerd. Hierdoor is men in staat projecten binnen zekere grenzen beheerst uit te voeren. In een projectteam is voldoende discipline voorhanden om behaalde resultaten in vergelijkbare projecten in de toekomst te evenaren.

Niveau 3: "Defined"

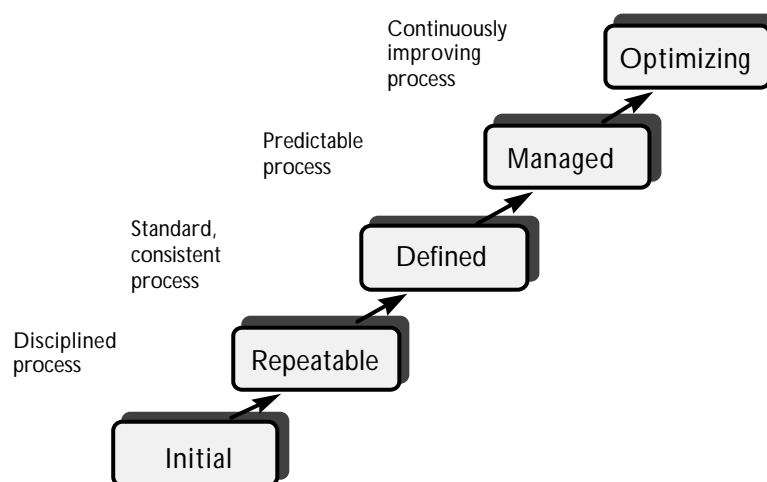
Alle processen met betrekking tot projectmanagement en software-engineering-activiteiten zijn gestandaardiseerd en vastgelegd voor de hele organisatie. Alle projecten binnen de organisatie worden conform dit standaardproces uitgevoerd, slechts op weloverwogen en overeengekomen punten afwijkend.

Niveau 4: "Managed"

Op dit niveau worden gedetailleerde metingen uit het softwareproces en de resulterende softwareproducten verkregen. Aan de hand van dit statistische materiaal is men in staat objectieve beslissingen te nemen.

Niveau 5: "Optimizing"

Continue procesverbetering wordt mogelijk gemaakt door kwantitatieve terugkoppeling uit het ontwikkelingsproces en door innovatieve ideeën en nieuwe technieken uit te proberen.



Figuur 1: Structuur van het Capability Maturity Model

B. Vakmanschap (Ref 2)

Uit onderzoek blijkt, dat een gemiddelde automatiseerder per 10 regels programmatuur 1 fout maakt. Deze fouten worden in het vervolgtraject (inspecties, testen) zo veel mogelijk verwijderd. Gemiddeld wordt 95% van alle fouten door middel van inspecties en testen gevonden en geëlimineerd. Een dure oplossing, want het in een later stadium herstellen van een fout is aanzienlijk duurder dan het vroeger signaleren of zelfs voorkomen van een fout. De gegeven ervaringscijfers betekenen, dat een applicatie van 8 miljoen regels programmatuur (bijvoorbeeld een Boeing 747) nog steeds 40.000 fouten zou bevatten. Gelukkig wordt dit soort applicaties veel intensiever getest, zodat het aantal fouten in de praktijk beduidend lager zal zijn. Maar foutvrije applicaties van deze omvang bestaan niet en het volledig testen ervan is onmogelijk. Dit is een steeds grotere reden tot ongerustheid, gezien de spectaculaire groei van het aantal producten waar software in zit. Voorbeelden: ziekenhuisapparatuur, financiële transactiesystemen, vliegtuigen, telefooncentrales, satellieten, televisies.

Waar alle nieuwe modellen en aanpakken geen aandacht aan lijken te schenken, is het oplossen van de fouten daar waar zij ontstaan: de bron. Men tracht te komen tot foutvrije software door steeds intensiever te testen. De bron van het probleem, het huidige prestatieniveau van de eigen automatiseerders, is zelden bekend, laat staan dat er wordt gekeken hoe dit kan worden verbeterd. Men accepteert het prestatieniveau vreemd genoeg als normaal, zich niet realiserend dat dit steeds vaker tot levensbedreigende situaties kan leiden. Ook de afnemer van software produkten heeft boter op het hoofd. Niet werkende software produkten worden aangeschaft en de afnemer is zelfs bereid om voor de kosten te betalen om het produkt werkend te krijgen.

Het **Personal Software Process** is een methode die aandacht schenkt aan het verbeteren van het vakmanschap van automatiseerders. Het is wederom ontwikkeld door Watts Humphrey in samenwerking met SEI en is een raamwerk ter ondersteuning van prestatieverbetering van de individuele automatiseerder. Aan de hand van oefeningen wordt iedere deelnemer inzicht gegeven in zijn huidige manier van werken en de mogelijkheden tot verbetering. De eerste oefeningen verschaffen inzicht in:

- de schattingsvaardigheid (duur werkzaamheden, grootte eindprodukt);
- het aantal fouten dat men maakt;
- de eigen produktiviteit (aantal regels programmatuur of functiepunten per tijdseenheid).

In volgende oefeningen wordt men stapsgewijs geconfronteerd met een steeds meer gedisciplineerde werkwijze. Deze bestaat uit het maken van schattingen aan de hand van resultaten van vorige oefeningen, het verwijderen van fouten in eigen werk door kritische inspecties en het opdelen van de werkzaamheden in overzichtelijke delen. Vervolgens laten de resultaten van de uitgevoerde oefeningen de deelnemer zelf tot het inzicht komen dat een gedisciplineerde werkwijze leidt tot grote verbeteringen. Bovendien ervaart men, hoe belangrijk het is om van fouten uit het verleden te leren en ze te vermijden. Het feit dat men alle aspecten aan den lijve ondervindt is zonder meer de sleutel tot het succes van deze methode.

Niveau	Aandachtsgebieden
stap 1: PSP0	Inzicht verkrijgen huidige werkwijze
stap 2: PSP1	Verbeteren schattingsvaardigheid
stap 3: PSP2	Verwijderen fouten door reviews
stap 4: PSP3	Decompositie van complexere problemen

Figuur 2: Structuur PSP

C. Metrics (Ref3)

Binnen automatiseringsorganisaties zijn in het algemeen weinig ervaringscijfers bekend over het verloop van projecten uit het verleden. Tengevolge hiervan is het maken van kwantitatieve prognoses voor nieuwe projecten uiterst moeilijk. In de praktijk gebeurt meestal het volgende. Bij het plannen van automatiseringsprojecten wordt in het algemeen begonnen met het vaststellen van de einddatum: de datum waarop het resulterende produkt gereed dient te zijn. Hierbij gaat men uit van de op dat moment bekende wensen en berekent vervolgens wat het zal kosten in termen van inspanning (mensen) en geld om die functionaliteit te kunnen realiseren. Er worden eenvoudige lineaire verbanden verondersteld tussen de parameters doorlooptijd, benodigde inspanning en hoeveelheid te realiseren functionaliteit. Men maakt hierbij een aantal standaard fouten:

- er wordt onvoldoende rekening gehouden met uitbreiding van en wijziging van de oorspronkelijk vastgestelde functionaliteit;
- er wordt al te vaak geëxperimenteerd met het ondoordacht toepassen van relatief onbekende technologieën (Windows, Java) en nieuwe methoden en technieken (Object Oriëntatie, RAD), waar nog geen enkele ervaring mee opgedaan is;
- de kwaliteit van het eindprodukt wordt niet of onvoldoende gespecificeerd;
- men verzuimt afgesloten projecten te evalueren om op deze wijze een ervaringsdatabase op te bouwen, die kan worden gebruikt bij het inrichten van toekomstige projecten;
- omdat men de niet-lineaire verbanden tussen verschillende parameters niet kent, wordt geen gefundeerde afweging gemaakt wat het effect zou kunnen zijn van een kortere/langere

doorlooptijd of een beperking/uitbreiding van de functionaliteit op het budget (inspanning en kosten).

Met name dit laatste aspect is interessant, omdat er niet-lineaire verbanden tussen deze parameters bestaan. Zo zal het verminderen van een hoge tijdsdruk door het verlengen van de doorlooptijd kostenbesparend werken, waarbij bovendien de kwaliteit van het eindproduct beter kan worden gegarandeerd. Voorbeeld: het verlengen van de doorlooptijd van 8 tot 13 maanden, kan een factor 5 aan kostenbesparing opleveren.

In figuur 3 zijn relaties tussen een aantal parameters weergegeven. Het onderzoeken van de niet-lineaire verbanden tussen deze parameters staat al decennia lang in de aandacht. Hoewel men door het evalueren van duizenden automatiseringsprojecten indrukwekkende resultaten heeft bereikt, worden de uitkomsten hiervan nog zelden toegepast.

Figuur 3 laat meer zien. Het in de hand houden van de functionaliteit (streven naar minimale produktgrootte) heeft een positief effect op doorlooptijd, budget en kwaliteit. Dit is voor de hand liggend, maar ervaringscijfers leren dat automatiseringsprodukten gemiddeld 30 tot 50% ongebruikte functionaliteit bezitten.

Hiernaast is het van groot belang de produktiviteit van een organisatie te verhogen. Enerzijds kan dit gebeuren door te als organisatie te investeren in procesverbetering (bv. aan de hand van het Capability Maturity Model). Anderzijds verdient het vakmanschap van de individuele aandacht (bv. op basis van het Personal Software Process).

Driver	Doorlooptijd	Inspanning (kosten)	Kwaliteit
Produktgrootte	+	+	-
Produktiviteit	-	-	+
Tijdsdruk	-	+	-

Figuur 3: Relatie tussen parameters

Huidige Situatie

Uiteraard zal de lezer benieuwd zijn wat de huidige situatie in Nederland is. Deze verschilt niet veel van de situatie in andere landen. Ongeveer 75-80% van alle software bedrijven bevindt zich op CMM niveau 1, het PSP wordt slechts sporadisch toegepast en het verzamelen van en sturen op kwantitatieve gegevens staat nog in de kinderschoenen. Dit lijkt wellicht een pessimistisch beeld, echter de laatste jaren heeft een aantal bedrijven in Nederland toch een behoorlijke stap in de goede richting gemaakt. Zowel in de technische als de administratieve automatiseringsbranche staat het CMM breed in de belangstelling. Enkele bedrijven gebruiken het CMM al jaren als referentie om hun verbeteringsactiviteiten richting te geven. Voorbeelden zijn: Baan Development B.V., diverse produktdivisies binnen Philips, Hollandse Signaalapparaten en Ericsson Telecommunicatie B.V. Met name Ericsson heeft in korte tijd spectaculaire resultaten weten te bereiken. Het PSP vindt inmiddels langzaam haar intrede in Nederland. Binnen Baan wordt momenteel een aantal mensen opgeleid en ook andere bedrijven hebben belangstelling. Ten aanzien van metrics is er ook veel gaande. Steeds meer bedrijven onderkennen dat het nemen van beslissingen op gevoel te risicovol is en investeren in het opzetten van een ervaringsdatabase om toekomstige projecten beter in te kunnen richten. Binnen het Automatiseringscentrum van de Belastingdienst in Apeldoorn wordt hier momenteel veel aandacht aan gegeven.

Om bedrijven die bezig zijn met deze materie met elkaar in contact te brengen, is in 1995 een netwerk opgericht onder de naam SPIDER. Dit netwerk organiseert elke kwartaal een landelijke bijeenkomst om rondom een vastgesteld thema ervaringen uit te wisselen. Hiernaast zijn enkele

wergroepen opgericht. Indien men geïnteresseerd is om nader kennis te maken met SPIDER, kan men contact opnemen met de auteur van dit artikel (E-mail: hanss@aa.nl).

Alert Automation Services b.v. werkt met een aantal toonaangevende bedrijven samen om op de genoemde gebieden concrete vooruitgang te boeken. Hiernaast verzorgt Alert Automation Services b.v. trainingen op het gebied van het CMM, het PSP en Metrics.

Auteur

Ir.J.A.Sassenburg is directeur van Alert Automation Services b.v. te Veldhoven. Hij is gastdocent aan de Technische Universiteit Eindhoven en voorzitter van het landelijke SPI netwerk SPIDER.

Literatuur

Ref1 "Capability Maturity Model for Software, Version 1.1", M.C.Paulk, B.Curtis, M.B.Chrissis, C.V.Weber, Technical Report, CMU/SEI-93-TR-24, SEI, Carnegie Mellon, Pittsburgh, USA..

Ref2 "Introduction to the Personal Software Process", W.S.Humphrey, ISBN 0-201-54809-7.

Ref3 "Measures for Excellence: Reliable Software on time within budget", L.H.Putnam, W.Myers, ISBN 0-13-567694-0.