

Eisenbeheer nader bekeken

Evaluatieproject van Serc en Comutable

Ondanks tientallen jaren ervaring met software-ontwikkeling mislukken de meeste ontwikkeltrajecten nog steeds. Een raamwerk als het 'Capability Maturity Model' biedt uitkomst bij het verbeteren van zowel het softwareproduct als het ontwikkelproces. Het beheersen van de eisen van de klant staat daarbij voorop en speciale tools kunnen dit proces ondersteunen, stellen twee medewerkers van het Software Engineering Research Centre.

ROB MAIJERS EN DANNY GREEFHORST

Om de stand van zaken op het gebied van *requirements management tools* (eisenbeheer) te bepalen, organiseert het Software Engineering Research Centre (Serc) in samenwerking met *Comutable* het 'requirements management'-evaluatieproject (zie kader). De deelnemers aan het project evalueren tools aan de hand van een uitgebreid raamwerk van criteria. Dit raamwerk is gebaseerd op een ontwikkeltraject, met het idee dat er één model is, dat de software-engineer vanuit verschillende, aan elkaar gerelateerde gezichtspunten kan bekijken. Deze gezichtspunten zijn onder andere het domeinmodel, de architectuur, het ontwerp, de implementatie en de 'runtime'-omgeving. Eisen vormen ook één van de mogelijke manieren waarop hij naar het model kan kijken. Vanuit deze visie geeft het raamwerk een aantal evaluatiecriteria, waarvan de belangrijkste hieronder punt voor punt aan bod komen. *Ontwikkelproces* - Hoewel de nadruk van eisenbeheer op de analysefase van het ontwikkeltraject ligt, speelt het ook in alle andere fasen een belangrijke rol. Vrijwel iedereen die bij het traject betrokken is, krijgt er mee te maken. In het vervolg noemen we de mensen die te maken hebben met het beheren van eisen 'analisten'. Hun taak is niet alleen het verzamelen en beheren van eisen, ook wijzigingen in eisen elders in het traject kunnen invloed hebben op een bepaalde fase. De precieze wijze waarop eisenbeheer is ingericht binnen een ontwikkeltraject kan per organisatie sterk verschillen. Een productleverancier zal een geheel ander eisenbeheer-proces hebben dan een ontwikkelafdeling binnen een bedrijf. Interessant is om te bepalen op welke wijze tools ondersteuning bieden in dit proces en in hoeverre een tool kan worden toegesneden op een specifieke organisatorische context.

Identificeren en traceren

Identificatie van eisen - Eisen zijn vaak al beschreven in documenten die de analist vervolgens moet invoeren in het tool om die eisen te beheren. Hierbij moet de identificatie van eisen zo veel mogelijk automatisch verlopen door verregaande ondersteuning van het tool. Zo kan het tool controles uitvoeren op de eisen, bijvoorbeeld of er conflicten tussen eisen bestaan en of alle belangrijke eigenschappen van een eis zijn ingevuld.

Traceerbaarheid - Zoals eerder gezegd kan een software-engineer vanuit een bepaald gezichtspunt naar één model kijken. Dit kan op verschillende abstractieniveaus. Zo kan een analist eisen onderverdelen in 'hoog niveau' (product-) eisen, software-eisen, enzovoort. Het is daarbij nodig dat hij eisen op een hoger niveau vertaalt in eisen op lagere niveaus. Daarnaast levert een ontwikkeltraject verschillende tussen- en eindproducten (analyse, architectuur, ontwerp en implementatie), die gerelateerd zijn aan software-eisen. Het is belangrijk de relaties tussen eisen onderling en tussen eisen en andere producten te documenteren om traceerbaarheid te bewerkstelligen. Traceerbaarheid maakt het onder meer mogelijk om de impact van wijzigingen te bepalen. Daarnaast geeft het inzicht in voortgang van het traject. Interessant is om te weten in hoeverre de tools traceerbaarheid tussen eisen onderling en van eisen naar andere producten van het ontwikkeltraject ondersteunen en op welke wijze de tools deze informatie verder gebruiken.

Samenwerking en documentatie

Metrieken - Metrieken geven kengetallen over de eisen die bijvoorbeeld gebruikt kunnen worden om een ontwikkelproces bij te sturen. Idealiter houden de tools metrieken automatisch bij en gebruiken ze deze informatie op een actieve manier.

Samenwerking - 'Requirements management' is een activiteit waarbij veel verschillende personen, mogelijk op verschillende locaties en met verschillende werkgebieden zijn betrokken. Deze verscheidenheid aan personen wordt veroorzaakt door de veelheid aan disciplines die gedurende het gehele ontwikkeltraject nodig is en door de potentieel verschillende partijen die belang hebben bij het traject. Verder is er bij grote projecten een groot aantal mensen nodig voor de uitvoering. Het is daarom belangrijk dat het tool helpt in het verstrekken van toegang tot eisen voor al deze partijen, onafhankelijk van hun locatie en hun precieze systeemconfiguratie. Zo zou het tool ondersteuning kunnen bieden voor discussies over eisen.

Gegevensuitwisseling - Aangezien er meestal veel verschillende personen betrokken zijn bij deze vorm van managen is het van belang dat het gereedschap het mogelijk maakt eisen te synchroniseren met ander gereedschap om die eisen te beheren. Daarnaast zijn eisen uitgangspunt voor andere producten van het ontwikkeltraject, zoals het ontwerp. Daarom is integratie met andere tools, bijvoorbeeld ontwerptools, van belang. Tools zouden dus vormen van

gegevensuitwisseling moeten bieden en het liefst op een zo transparant mogelijke wijze.

Versie- en configuratiebeheer - Nadat eisen zijn opgenomen in het tool zullen ze nog vaak wijzigen. Tools moeten ondersteuning bieden om deze wijzigingen te verwerken. Daarbij is het van belang om te kunnen zien wie, wat, wanneer en waarom heeft gewijzigd. De analist moet een groep eisen als een stabiele versie (baseline) kunnen definiëren. Hiernaar kan dan gerefereerd worden, bijvoorbeeld om te kunnen stellen dat een softwareproduct aan de eisen in een bepaalde *baseline* voldoet.

Rapportage en documentatie - Het is van belang een goed overzicht te kunnen krijgen van alle eisen, hun eigenschappen en hun relaties met andere eisen en producten van het ontwikkeltraject. Bovendien dienen tools documenten met eisen te kunnen produceren. Het is dus belangrijk dat tools uitgebreide ondersteuning bieden voor rapportages en documentatie op maat.

Gebruikersvriendelijkheid - Naast alle functionaliteit die een tool biedt, is het van belang dat een tool gebruikersvriendelijk is. Hierbij spelen vragen als: In hoeverre is het tool instelbaar? Sluit het tool aan bij de taken van de gebruiker? En hoe eenvoudig en toegankelijk is het tool?

Software-eisen

Aanleiding voor het evaluatieproject is de opkomende aandacht voor softwarekwaliteit, waarbij eisenbeheer een belangrijke rol speelt. Daarnaast is het zo dat 'requirements management'-tools nog relatief onbekend zijn. Het evaluatieproject is een manier om inzicht te krijgen in de stand van zaken op het gebied van dit management-gereedschap en de rol die het kan spelen in het ontwikkeltraject.

Om een beeld te vormen van wat 'requirements management' precies betekent, is het nodig eerst eens nader te bekijken wat een software-eis inhoudt en welke rol software-eisen spelen bij het ontwikkelen van programmatuur.

In de eerste fase van een ontwikkeltraject dienen analisten een programma van eisen vast te stellen, dat beschrijft wat het systeem zou moeten doen en welke beperkingen een rol spelen. Een eis is dus een precies gedefinieerde eigenschap of beperking van een softwaresysteem. Merk hierbij op dat er een verschil bestaat tussen wensen, eisen en doelstellingen van een systeem. Een klant kan wel een bepaalde wens hebben, maar een analyse moet bepalen of het ook een eis aan het systeem is. Een eis is dus iets waaraan het systeem moet voldoen en wat iemand kan testen om te bepalen of het systeem er aan voldoet. Doelstellingen geven aan wat men met het systeem wil bereiken. Deze zijn algemener van aard dan eisen en daardoor vaak moeilijk te testen.

Natuurlijke taal

Een analist kan eisen op verschillende wijzen uitdrukken. In veruit de meeste gevallen gebruikt hij daarvoor natuurlijke taal. Daarnaast is het ook mogelijk gestructureerde of formele talen te gebruiken voor het definiëren van eisen. Tenslotte kan hij ook illustraties gebruiken om eisen te beschrijven.

Het programma van eisen moet niet worden onderschat. Voor een ingewikkeld systeem kan het uit duizenden eisen bestaan. Daarnaast moet het programma van eisen zelf ook aan bepaalde criteria voldoen om bruikbaar te zijn bij bijvoorbeeld de realisatie van het systeem. Zo moeten de eisen volledig en consistent zijn. Alles wat het systeem moet doen, dienen de analisten vast te leggen en de eisen mogen uiteraard niet met elkaar in strijd zijn. In de praktijk is dit vaak moeilijk te controleren, vooral als de eisen in natuurlijke taal zijn geformuleerd. Ondersteuning door een tool kan uitkomst bieden.

Het programma van eisen kan, naast het beschrijven van eisen voor de te realiseren software, ook dienen als naslagwerk later in het traject. Om een goed naslagwerk te kunnen vormen in bijvoorbeeld de onderhoudsfase, is een standaard structuur van een programma van eisen nodig.

Wederzijds begrip

Eisenbeheer is een activiteit gedurende het software-ontwikkeltraject die tot doel heeft het bewerkstelligen en onderhouden van een wederzijds begrip tussen de klant en het ontwikkelteam omtrent de eisen waaraan het softwaresysteem moet voldoen. Dit houdt niet alleen het analyseren en documenteren van eisen in aan het begin van het traject, maar ook het beheren van eisen en wijzigingen hierin gedurende het gehele proces. Zo dient het ontwikkelteam de initiële eisen te relateren aan specifieke software-eisen en -producten zoals ontwerp, code en testspecificaties.

Om een beter beeld te krijgen van eisenbeheer is het illustratief te kijken naar de rol van deze managementvorm in het Capability Maturity Model (CMM). Het CMM is een model voor het beschrijven van de volwassenheid van software-ontwikkelorganisaties. Daarnaast kan het dienen als hulpmiddel voor het verbeteren van deze organisaties.

Het model is ontstaan door bestaande ontwikkelorganisaties te bestuderen en deze in categorieën in te delen. Het CMM verdeelt ontwikkelorganisaties in vijf niveaus van volwassenheid, de zogenaamde CMM-'levels'. Organisaties bevinden zich in dit schema minimaal op het eerste niveau. Hogere niveaus zijn gericht op het beheersen, standaardiseren en optimaliseren van het software-ontwikkelproces en het meten hiervan, om het te kunnen bijstellen en de kwaliteit te kunnen borgen. Naast deze gelaagde structuur vormen Key Process Areas (KPA) een belangrijk onderdeel van het CMM. Deze zijn van primair belang om van een bepaald CMM-niveau te kunnen spreken. Per KPA geeft CMM een aantal doelen die de organisatie dient te realiseren en activiteiten (key practices) die invulling geven aan deze doelen. 'Requirements management' is één van de belangrijkste KPA's van CMM niveau 2. Hierbij zijn de doelen het beheersen van de eisen, om te komen tot een 'baseline' van eisen, en het consistent houden van alle producten en activiteiten met deze 'baseline'. Om deze doelen te verwezenlijken dienen de bij het traject betrokken mensen een aantal activiteiten uit te voeren:

- n Op de eerste plaats moeten zij de eisen inspecteren alvorens ze mee te nemen in het ontwikkelproces. Hierbij

- dienen zij de eisen onder andere te controleren op volledigheid, consistentie, haalbaarheid en testbaarheid.
- n Op de tweede plaats moeten zij de eisen gebruiken als basis voor plannen, producten en activiteiten.
- n Als laatste dienen zij alle wijzigingen aan de eisen te inspecteren en mee te nemen in het ontwikkeltraject. Hierbij moeten zij uiteraard kijken naar de impact van wijzigingen op het traject. Het is van groot belang de wijzigingen te overleggen met alle betrokken partijen.

Veel administratief werk

De overige activiteiten zijn algemener van aard en hebben ook betrekking op andere KPA's. Zo dienen de softwaredeskundigen eisen bijvoorbeeld te documenteren, dienen er adequate middelen aanwezig te zijn en moet de organisatie mensen trainen in het beheren van eisen.

Binnen het CMM is eisenbeheer vooral een organisatorische activiteit. Een ontwikkelorganisatie moet standards en richtlijnen definiëren en rollen toewijzen om deze manier van managen vorm te geven. Het goed uitvoeren van eisenbeheer gaat gepaard met een grote hoeveelheid administratie voor het documenteren en structureren van eisen, waardoor ondersteuning door tools onmisbaar is. Het is echter de vraag in hoeverre deze tools meer zijn dan alleen een administratief hulpmiddel, vergelijkbaar met een tekstverwerker of spreadsheet. Op de vraag in hoeverre tools om eisen te beheren de hierboven geschetste grotere rol in het ontwikkeltraject kunnen vervullen, probeert dit 'requirements management'-evaluatieproject een antwoord te vinden.

Danny Greefhorst en Rob Maijers, onderzoekers bij het Software Engineering Research Centre (Serc) te Utrecht.

Evaluatieproject eisenbeheer

Het doel van het 'requirements management'-evaluatieproject is de stand van zaken rond deze tools te peilen, maar ook te onderzoeken welke mogelijkheden eisenbeheer biedt om de kwaliteit van software in het algemeen te verbeteren. Het vier dagen durende evaluatieproject begint met een introductie waarin verscheidene ervaringen uit de praktijk de revue zullen passeren. Tevens wordt een theoretisch overzicht geboden van de kwaliteitsverbetering van de software als gevolg van deze managementvorm.

Paul Hendriks van Serc zal de rol van eisenbeheer binnen het Capability Maturity Model uiteenzetten. Praktijkervaringen komen van Paul Siemons van Signaal Communications en Sjaak Brinkkemper van Baan Company.

Daarnaast zullen de verschillende leveranciers overzichts-presentaties houden van de meest recente versie van hun tool. Aan het project doen de belangrijkste eisenbeheer-tools mee: RequisitePro van Rational Software; Doors van QSS, gedistribueerd door Bergson Software Tools; Caliber-RM van TBI, verspreid door Princeton Softech; en Cradle van 3SL, verkrijgbaar bij SOMS Software Tools.

Na de introductie krijgen de deelnemers de kans om drie van de vier tools nader te bekijken. Hiertoe zullen zij een uitgebreide cursus krijgen over één tool en daarmee een casus uitwerken. De deelnemers evalueren de tools aan de hand van een door Serc opgesteld evaluatie-aanpak en onder begeleiding van leveranciers en medewerkers van Serc. Daarna hebben de deelnemers de mogelijkheid om twee van de andere tools te bekijken. Een presentatie van de bevindingen van de verschillende teams en een paneldiscussie vormen de afsluiting.

Het evaluatieproject vindt plaats van 14 tot en met 17 september 1999. Kosten: 4475 gulden per deelnemer, met 500 gulden korting voor de volgende deelnemers van hetzelfde bedrijf. Abonnees van 'Computable' krijgen 200 gulden extra korting.

Dit evaluatieproject is bij uitstek interessant voor personen die betrokken zijn bij 'requirements management', softwarekwaliteitsverbetering, CMM of voor personen die anderszins meer willen weten over deze managementvorm en de tools zij daarbij kunnen gebruiken.

U kunt zich schriftelijk, telefonisch, per fax of e-mail opgeven bij:

Software Engineering Research Centre

Postbus 424

3500 AK Utrecht

Telefoon: 030-2545412

Telefax: 030-2545948

E-mail: info@serc.nl

Meer informatie is te vinden op de website van Serc: <http://www.serc.nl/>