

Met alle aandacht voor nieuwe ontwikkelhulpmiddelen en technologie wordt vaak vergeten dat de levenscyclus van een softwareproduct veel meer omvat dan alleen een eenmalig ontwikkeltraject. Voor de bouw moet het product worden gepland en gespecificeerd. Na de bouw moet de software worden verpakt en bij de gebruikers beschikbaar worden gesteld. Vervolgens moet er ondersteuning zijn bij het gebruik. De problemen die daarbij optreden kunnen leiden tot aanpassingen die vervolgens weer moeten worden "uitgerold". Om zo'n levenscyclus onder controle te krijgen zijn verschillende processen en ondersteunende middelen nodig. Hoe zit het met dit thema vanuit een logistieke invalshoek?



thema

Levenscyclus software meer dan éénmalig ontwikkeltraject

Softwarelogistiek

Softwarelogistiek heeft betrekking op de inrichting en ondersteuning van de definitie, bouw, assemblage, verspreiding, ondersteuning en evolutie van softwareproducten. Terwijl software vanuit het oogpunt van ontwikkeling wordt beschouwd als het eindresultaat, is het vanuit softwarelogistiek gezien een 'object' van manipulatie, administratie, opslag en transport. Softwareontwikkeling is daarbij niet meer dan één van de stappen.

Het goed inrichten van software logistiek is op zich al niet eenvoudig. Maar er zijn diverse actuele ontwikkelingen die de complexiteit nog verder verhogen. De behoefte aan een goed georganiseerde aanpak wordt hiermee alleen maar groter. Een eerste trend is de verschuiving naar een meer iteratieve en evolutionaire projectaanpak, waarbij softwareproducten continu aan wijziging en integratie onderhevig zijn. Daarnaast zien we dat de snelheid waarmee nieuwe (versies van) producten op de markt worden gebracht toeneemt. In de wereld van Internet bijvoorbeeld volgen nieuwe releases elkaar in sneltreinvaart op. Dit alles vergt natuurlijk een goede administratie van alle gemaakte en uitgeleverde software elementen. Naast de projectmethodiek is ook de ontwikkelmethodiek aan het veranderen. Hierbij speelt componentgebaseerd ontwikkelen een belangrijke rol. Een goede modularisatie van het softwareproduct levert niet alleen beter onderhoudbare systemen op, er zijn ook voordelen voor ontwikke-

ling. Gegeven een goede opsplitsing in componenten ligt het voor de hand dat deze op verschillende locaties en/of door verschillende ontwikkelteams kunnen worden gebouwd. Daarnaast is ook het hergebruik van componenten een belangrijke motivatie. Dit alles leidt tot een situatie waarin softwarecomponenten in verschillende projecten worden gebruikt en gebouwd en in verschillende producten worden uitgeleverd.

Coördinatie en synchronisatie van de ontwikkeltrajecten en een goede administratie over welke (versies van) componenten waar worden gebruikt zijn hierbij cruciaal.

Als laatste aspect signaleren we een toenemende variatie in softwareproducten. Waar vroeger één product werd gebouwd, gaat het nu veelal om een familie van producten, waarbij elk element van de familie al dan niet bepaalde kenmerken of functionaliteit biedt. Een bekend voorbeeld hiervan zijn ontwikkelhulpmiddelen, die vaak in een 'personal', 'professional' en 'enterprise' variant worden geleverd. Het beheersen van deze variaties stelt hoge eisen aan de inrichting en ondersteuning van logistiek.

MODEL VOOR SOFTWARELOGISTIEK Het SERC heeft de afgelopen tijd gewerkt aan een generiek model voor softwarelogistiek dat inspeelt op de hierboven gesignaleerde trends. Een globaal overzicht van dit model is weergegeven in de poster in deze uitgave. Belangrijk bin-

Poster

In dit nummer van Software Release Magazine is een poster bijgevoegd die een stroomschema bevat over het onderwerp software logistics. Dit artikel bevat een toelichting hierop.

nen dit model zijn de verschillende processen die een rol spelen (definitie, bouw, assemblage, verspreiding, ondersteuning). Binnen deze processen vindt een aantal activiteiten plaats die betrekking hebben op een aantal soorten gegevens. Deze gegevens worden opgeslagen in speciaal daarvoor ontwikkelde opslagplaatsen. We zullen de verschillende processen met bijbehorende activiteiten en gegevens hier globaal toelichten:

- *Definitie*

Binnen dit proces worden de productfamilies, producten, onderdelen, en de eisen die hieraan worden gesteld gedefinieerd. De eerste stap hierbij is het opstellen van de eisen aan producten. Op hoog niveau kunnen deze worden gepresenteerd als features; belangrijke eigenschappen van producten. Producten met veel gemeenschappelijke features kunnen worden gegroepeerd in een productfamilie. Dit heeft als voordeel dat alle gemeenschappelijke onderdelen slechts één keer gedefinieerd en gebouwd hoeven te worden. Het is hierbij ook van belang de verschillen tussen (de features van) producten in kaart te brengen. De definitie van productfamilies, producten en onderdelen behelst verder het opstellen van de architectuur waarbij verantwoordelijkheden van de individuele onderdelen in kaart worden gebracht. Ook wordt hierbij aangegeven hoe de verschillende features in onderdelen worden gerealiseerd. De resultaten van het definitieproces worden geversioneerd opgeslagen in een magazijn, waardoor tevens toekomstig hergebruik van deze definities mogelijk wordt.

- *Bouw*

Voor de bouw van de verschillende onderdelen wordt gebruik gemaakt van de definities die in het definitieproces zijn opgesteld. Het gaat hierbij dus om het vertalen van deze definities naar software. Belangrijk hierbij is dat

onderdelen voldoen aan hun definitie aangezien deze fungeert als contract waarin de diensten die andere onderdelen kunnen gebruiken zijn gespecificeerd. Het bouwproces kan bestaan uit verschillende soorten activiteiten. Zo kunnen onderdelen op maat worden gemaakt, maar is het ook mogelijk bestaande commerciële onderdelen in te kopen. Daarnaast speelt het aanpassen/onderhouden van bestaande onderdelen een belangrijke rol. Het resultaat van al deze activiteiten dient te worden getoetst aan de definitie, waarna het onderdeel kan worden opgeslagen in een onderdelenmagazijn. Ook hier wordt de versiehistorie van de individuele items bijgehouden. Voor de goede orde merken we op dat individuele onderdelen uit allerlei artefacten (bestanden) kunnen bestaan. Deze worden natuurlijk – geversioneerd – opgeslagen bij de onderdelen, maar zijn verder logistiek van minder groot belang.

- *Assemblage*

Als de individuele onderdelen zijn gebouwd kan een product geassembleerd worden. Hiervoor moet er een selectie worden gemaakt uit de verschillende versies van de onderdelen zodat een consistent geheel ontstaat. Met deze versies kan dan aan de hand van een bouwvoorschrift het uiteindelijke product worden gebouwd. De specifieke versie van het product die hierbij ontstaat wordt aangeduid als een release. Het is van belang deze release te testen om te kijken in hoeverre deze functioneel en technisch voldoet aan de definitie van het product. Als de release de test is gepasseerd dan kan deze worden opgeslagen in het productenmagazijn, dat dient als catalogus van beschikbare producten.

- *Verspreiding*

Gebruikers kunnen hun relevante producten uit het productenmagazijn selecteren. Er moet dan natuurlijk gecon-

Logistiek van componenten en configuratiemanagement

Op 23 mei heeft een seminar plaatsgevonden met als thema 'Logistiek van Componenten & Configuration Management'. Op dit seminar is aandacht besteed aan een aantal zaken dat rondom dit thema speelt. Voor het effectief kunnen ontwikkelen van goede software is niet alleen de kwaliteit van de programmatuur van belang. Ook het inzichtelijk krijgen van aspecten rond configuratiemanagement (CM) is minstens zo belangrijk, zoals een overzicht van alle ontwikkelde productdelen (software, maar ook andere onderdelen), het onderling consistent houden van die delen, het om kunnen gaan met verschillende versies van onderdelen en het (her)produceren van releases. Daarnaast moet door steeds korter wordende releasecycli en veranderde distributiemogelijkheden rekening worden gehouden met de logistieke aspecten van software, zoals de verspreiding en installatie van software(componenten). In vergelijking tot CM ligt hier de nadruk op software als object voor manipulatie of transport en niet op software als eindresultaat van het ontwikkelingsproces. In zijn presentatie spreekt Gert Florijn de verwachting uit dat op het gebied van softwarelogistiek veel verandering zullen plaatsvinden. Nu ligt de focus met name op ontwikkeling en de daarbij behorende configuratieaspecten. De markt ontwikkelt zich echter van aanbodgericht naar vraaggericht. Softwareleveranciers zullen om moeten

kunnen gaan met specifieke wensen van een klant ten aanzien van functionaliteit, het leveren van nieuwe versies of updates en diversiteit in omgevingen waarin het product moet functioneren. Hierdoor worden aspecten als definitie van softwareproducten en componenten, installatie en distributie steeds complexer en belangrijker. Spreker Emile van de Logt van CMG richt zich op het capability maturity model (CMM) waarin CM als een aandachtsgebied is opgenomen en vertelt over de implementatie hiervan bij Signaal Communications. Van de Logt wijst onder andere op het gevaar bij de implementatie van CM te vervallen in regels en procedures die niet worden nageleefd. Het is van belang iedereen te betrekken om samen een nieuwe manier van werken aan te leren. Een belangrijk resultaat van de invoering van CM is dat iedereen weet wat CM inhoudt en betekent, waardoor discussies tot het verleden behoren. CM binnen softwareontwikkeling en beheer van software hebben duidelijke raakvlakken. Maarten Wijsman van Quint Wellington Redwood gaat in op beheer in het algemeen en specifiek op ITIL (IT Infrastructure Library): een praktische handleiding met procedures en werkinstructies voor het opzetten van een beheersorganisatie. De in ITIL gedefinieerde processen change management, software control & distribution en configuratiemanagement verschillen weinig met hoe men vanuit soft-

troleerd worden in hoeverre de configuratie bij de gebruiker geschikt is voor het product. Mogelijk heeft de gebruiker ook nog andere producten of onderdelen nodig. Na het passeren van deze test kan het product worden ingepakt om vervolgens bijvoorbeeld via het netwerk naar het systeem van de gebruiker te worden gedistribueerd. Hier kan dan de installatie van het product plaatsvinden, waarna het tot de beschikking van de gebruiker staat. Uiteindelijk kan de gebruiker het product weer van zijn systeem verwijderen. Belangrijk bij deze activiteiten is administratie van gegevens over de systeemconfiguraties van gebruikers. Deze wordt bijgehouden in de zogenaamde configuratie database.

- *Ondersteuning*

Bij het gebruik van het product kunnen er allerlei onverwachte situaties optreden; de gebruiker weet niet wat hij moet doen of loopt tegen fouten in het product aan. Hier toe heeft de gebruiker contact met een helpdesk die verantwoordelijk is voor eerstelijns ondersteuning. De helpdesk registreert de afwijkingen van het normale gedrag in de issue database. Op basis van deze afwijkingen worden de problemen in het product geïnventariseerd. Deze problemen zijn op hun beurt weer uitgangspunt voor de wijzigingsverzoeken die worden geformuleerd door de change control board. Impact-analyse speelt hierbij een belangrijke rol door de impact van wijzigingen op het product, zijn onderdelen en alle andere relevante gerelateerde gegevens te laten zien. Hiervoor is behoefte aan een centrale plaats waar verwijzingen naar de gegevens in de verschillende magazijnen zijn opgeslagen. De enterprise repository is verantwoordelijk voor deze administratie en biedt een gebruikersinterface voor bijvoorbeeld impact-analyse, maar ook voor andere doeleinden zoals

faciliteren van hergebruik of het leveren van managementinformatie.

Tussen bovenstaande processen zijn allerlei relaties gedefinieerd. Zo is er een natuurlijke workflow van definitie naar ondersteuning, die overigens niet lineair doorlopen hoeft te worden, en zijn er terugkoppelingen van ondersteuning naar eerdere processen. Een probleem dat optreedt kan bijvoorbeeld veroorzaakt zijn door een programmeerfout in een onderdeel, maar ook door een fout in de definitie van het product.

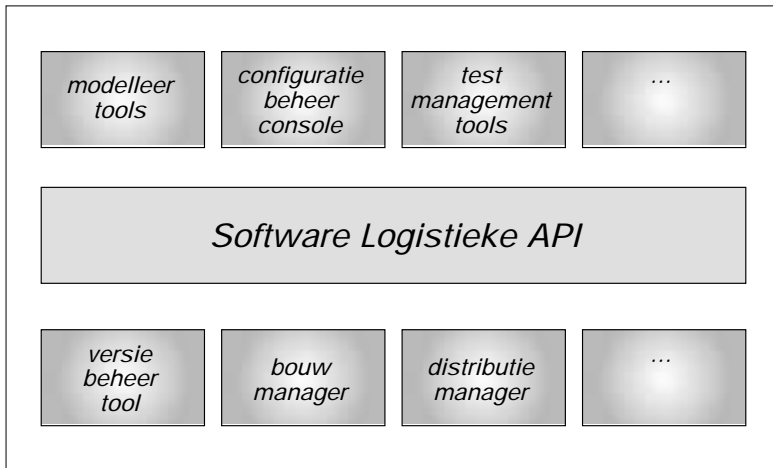
ONDERZOEK NAAR SOFTWARELOGISTIEK Softwarelogistiek is zeker nog geen volledig uitgewerkt terrein. Natuurlijk kunnen we bestaande kennis en ervaring incorporeren uit gebieden als softwareconfiguratiebeheer en de beheerwereld (onder meer ITIL). Maar er zijn ook diverse aspecten die nader onderzoek vergen. We noemen een paar voorbeelden.

SERC is in het kader van een Europees project betrokken bij onderzoek naar softwarelogistiek voor product-families en product-lines. In dat onderzoek wordt gewerkt aan een verdere detaillering van het gepresenteerde softwarelogistieke model. Een belangrijk aspect van het onderzoek richt zich op het bepalen van de impact van grote variaties in producten. Uitgangspunt hierbij is het gebruik van features voor het in kaart brengen van gemeenschappelijkheden en verschillen tussenproducten. Deze features kunnen worden gebruikt voor aansturing van de verdere logistieke processen.

Een ander belangrijk onderzoeksgebied is het administreren van relaties tussen items om te komen tot traceerbaarheid. Deze traceerbaarheid is van belang voor impact-

wareontwikkeling tegen logistiek en CM aankijkt. Wijsman spreekt echter ook over een kloof tussen beheer en ontwikkeling. Ericsson heeft een proces voor CM opgezet dat binnen de organisatie succesvol blijkt. Peter van de Gaag vertelt dat Ericsson een heldere organisatie en procedures heeft opgezet voor het beheren van producten, halfproducten en documenten. Centraal staat de change control board die verantwoordelijk is voor CM. Zij wordt in haar werkzaamheden ondersteund door processen voor onder andere change requests, problem reports en releases. Deze processen worden ondersteund door tools en databases. Van de Gaag benadrukt verder dat naast processen en tools de mensen een cruciale rol spelen en om deze reden duidelijk aandacht verdienen. Na de succesverhalen en afdwalingen naar beheer, wordt het publiek in de presentatie van Jan van de Sandt (Ordina) weer met de neus op de feiten gedrukt. Van de Sandt schetst een aantal typische problemen waartegen zijn ontwikkelteam is aangelopen gedurende een grootschalig ontwikkeltraject. Voorbeelden zijn het onderhouden van meerdere versies die bij de klant uitstaan terwijl tegelijkertijd nieuwe versies worden ontwikkeld en het distribueren van nieuwe versies of updates naar ruim 800 vestigingen in het hele land. Na de ontwikkeling van basisfunctionaliteit blijkt dat de vestigingen specifieke wensen hebben ten aanzien van het

softwareproduct. Problemen hierbij zijn hoe aan deze wensen kan worden voldaan en hoe het overzicht over de uiteenlopende versies behouden blijft. Mark van der Velden van Baan richt zijn aandacht op logistiek van softwareontwikkeling voor multi-site development. Baan heeft haar softwareontwikkeling zo georganiseerd dat teams in Amerika, Nederland en India één softwareproduct kunnen ontwikkelen. In het proces is een aantal taken duidelijk gescheiden. Iedere programmeur ontwikkelt in zijn eigen omgeving. De code wordt vrijgegeven in een gedefinieerde omgeving waarbij bijvoorbeeld gebruikte compilers en platform zijn geregistreerd. Pas daarna kan de softwarecomponent worden opgestuurd naar de build-groep. In deze groep komen alle componenten samen en wordt het complete product samengesteld. Hiervandaan kunnen releases naar de markt worden vrijgegeven. De build-groep test de software in een zoveel mogelijk geautomatiseerde omgeving. De ontwikkelaars krijgen hier vandaan hun feedback via intranet of e-mail. Met dit proces is men in staat binnen twee dagen een volledige build te maken en heeft men goede controle over de productie. De discussie tussen deelnemers en sprekers resulteert onder meer in de conclusie dat het toepassen van CM niet complex is maar alleen discipline vereist. Hoewel de praktijk vaak anders is zou CM dus een non-issue moeten zijn.



AFBEELDING: Tool-integratie-architectuur

analyse, maar kan ook voor andere doeleinden nodig zijn, bijvoorbeeld voor het bepalen van de voortgang van een project of het kunnen aantonen van vervulling van eisen omtrent safety. Belangrijke vragen zijn welke informatie er nodig is voor traceerbaarheid, hoeveel informatie nog hanteerbaar is, en hoe er tot deze informatie gekomen wordt.

Een laatste punt van onderzoek is de ondersteuning van het softwarelogistieke traject. Uiteraard is er een veel-

voud van tools die bepaalde onderdelen van het softwarelogistieke traject invullen, maar onderlinge integratie van deze tools laat vaak te wensen over. Bovendien is er vaak geen expliciete ondersteuning voor het ontwikkelen van producten in een productfamilie-aanpak. Ons onderzoek richt zich op de definitie van een integratielaag die een objectmodel beschikbaar stelt volgens het logistieke model (zie de afbeelding). Deze laag integreert aan de ene kant tools die opslag en bewerking van gegevens voor hun rekening nemen, en aan de andere kant tools die gegevens ontsluiten richting de eindgebruiker. Hierbij fungeert de laag als broker waardoor de vervanging van tools geen aanpassing behoeft in andere tools.

HOOGWAARDIGE PRODUCTONTWIKKELING Duidelijk is dat een goede organisatie van de processen - die een rol spelen in softwarelogistiek - essentieel is voor de realisatie van kwalitatief hoogwaardige productontwikkeling. De inrichting van softwarelogistiek vergt dan ook het aanpassen van de processen en activiteiten aan de organisatorische situatie en het selecteren, aanpassen en integreren van tools hiervoor.

Gert Florijn, Danny Greefhorst en Hugo Boer
zijn werkzaam bij het Software Engineering Research Centre te Utrecht.