

De manier waarop software wordt ontwikkeld is continu aan verandering onderhevig. Door veranderende methoden en nieuwe bevindingen evolueert de software steeds. Waar traditioneel gebruikt werd gemaakt van de procedurele ontwikkelmethode wint tegenwoordig de agile denkwijze sterk aan populariteit.

Modelgebaseerde softwareontwikkeling

De revolutie en evolutie van softwarebouw

Anvankelijk werd er gebruik gemaakt van de procedurele ontwikkelmethode. Hierbij worden instructies gecodeerd net zolang tot het eindproduct kan worden getest en opgeleverd. Vervolgens is de iteratieve ontwikkeling ontstaan. Deze methode verdeelt eindproducten onder in deelproducten waarna elke deelproduct op zichzelf ook iteratief wordt ontwikkeld. De deelproducten doorlopen altijd de projectfasen planning, analyse, ontwerp, testen en documentatie.

Tegenwoordig wint de agile methode steeds meer aan populariteit. Bij agile ontwikkelen staat feedback van gebruikers als basis voor de ontwikkeling. Er worden voortdurend kleine delen toegevoegd aan het eindproduct, waarna de gebruikers direct feedback kunnen geven. De ontwikkelcyclus is doorgaans eerder weken dan maanden. Vaak worden delen van software al tijdens het project opgeleverd. Op deze manier evolueert de software.

Huidige Trends

De evolutie van softwareontwikkeling is geïnitieerd door de veranderingen in de business en door technologische vernieuwingen. Bedrijven anticiperen continu op veranderende business processen, steeds korter wordende product-live-cycle en de uitdaging van het integreren van bestaande en nieuwe businesscombinaties. Het IT-landschap verandert ook snel. Voornamelijk de opkomst van het internet heeft ertoe bijgedragen dat een software ontwikkelaar niet meer om onderwerpen als XML, Open source, SOA en AJAX heen kan.

Traditionele problemen

Ondanks de opkomst van nieuwe technieken, zijn

de belangrijkste problemen in de softwareontwikkeling nog steeds niet opgelost. Wijzigingen in functionele specificaties zijn moeilijk inpasbaar in bestaande systemen. Het onderhoud van de broncode gebeurt nog steeds handmatig. Codegeneratoren geven weliswaar een versnelde ontwikkeling bij nieuwbouw, maar de praktijk laat zien dat hergenereren doorgaans niet mogelijk is zonder inmiddels gemaakte wijzigingen in de software ongedaan te maken. Het blijft dan ook bij een 'jump start', waarna onderhoud en aanpassingen handmatig moeten worden uitgevoerd.

Ook services en interfaces naar andere applicaties worden veelal handmatig onderhouden. Elke wijziging heeft impact op het correct functioneren van de applicaties.

Na het ontwerp van het datamodel wordt het datamodel niet getoetst tegen voor te definiëren kennisregels, maar wordt meestal direct begonnen met de bouw van schermen en business functionaliteit.

De architecturen van grotere informatiesystemen zijn complex. Hierdoor verliest men snel het overzicht en zijn wijzigingen steeds moeilijker door te voeren omdat de consequenties ervan niet te overzien zijn.

Model gedreven assembleren

De model gedreven benadering van software ontwikkeling biedt de mogelijkheid om bovenstaande problemen te reduceren of zelfs helemaal weg te nemen. Deze methode legt de nadruk op het modelleren van een informatiesysteem. In dit model worden de diverse objecten van het systeem vastgelegd. Op basis van dit model wordt het eindproduct gerealiseerd. Iedere wijziging wordt in het model vastgelegd, zodat het



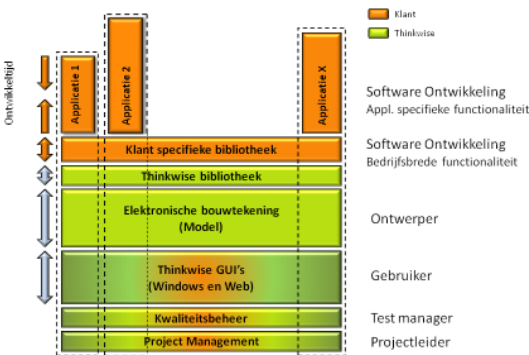
Frank Wijnhout

is creative analist bij Thinkwise Software. Hij is bereikbaar op fwijnhout@thinkwisesoftware.com

eindproduct vervolgens opnieuw kan worden opgebouwd. Er wordt dus niet geprogrammeerd in het eindproduct. De te programmeren code is in zijn geheel onderdeel van het model.

Thinkwise Software Factory

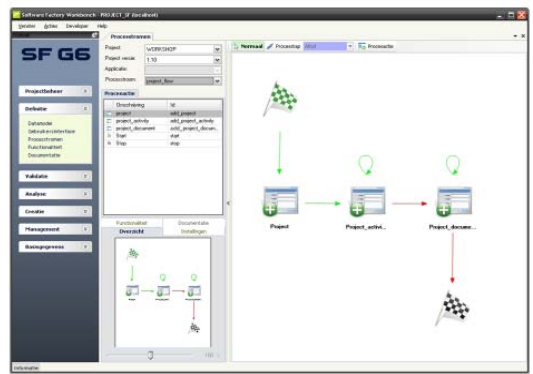
De Thinkwise Software Factory is een software ontwikkel gereedschap dat softwarebouw vanuit een geheel andere optiek benadert. Informatiesystemen bestaande uit gebruikers interfaces, business functionaliteit, documentatie en een database, worden volledig model gedreven ontwikkeld. Gegevens over het informatiesysteem die worden vastgelegd zijn zaken als tabellen, kolommen, relaties, source code templates, menu's, schermgedrag, processtromen en dergelijke. Vanuit dit gevulde model wordt code gegenereerd om een database- en functielaag op te bouwen. Abstracte gebruikersinterface componenten initialiseren zichzelf run time op basis van het model. Deze abstracte componenten zijn in meerdere technologieën beschikbaar, zodat Windows en Web schermen kunnen worden toegepast voor een informatiesysteem. Het model wordt niet vastgelegd in een taal, maar bestaat volledig uit database gegevens. Deze bouwtekening is daardoor platform- en programmeertaal onafhankelijk.



Eindproduct Architectuur.

Ontwikkelproces

De ontwerper modelleert de applicatie met behulp van grafische modelers. De gebruiker voorziet de ontwerper van feedback die de ontwerper doorvoert in de bouwtekening. Automatische validatie zorgt ervoor dat het model foutloos is nog voordat het eindproduct wordt opgebouwd. Ook mag het ontwerp altijd veranderen, zelfs als de ontwikkeling van functionaliteit al is begonnen. Het is zelfs mogelijk om deelsystemen al tijdens het ontwikkeltraject in gebruik te nemen. Gebruikers zijn gedurende het hele ontwikkeltraject betrokken en kunnen continu testen en feedback geven. Deze ontwikkelmethode resulteert in een andere aanpak voor het testen van de applicatie. De tester houdt zich veel minder bezig met het testen het eindproduct, maar legt zijn focus veel meer op het



Grafische workflow modeler.



Geïntegreerde management info in eindproduct.

valideren van de bouwtekening. Het testen start dus al voordat er een eindproduct is.

Daarnaast is geïntegreerde stuurinformatie beschikbaar in de software factory. Deze management informatie is dynamisch door zelf verschillende dimensies en meetwaarden tegen elkaar af te zetten. Vraagstukken als aantal regels code versus aantal regels commentaar, locaties van geweven code in het eindproduct en gewijzigde omvang van een nieuwe versie, zijn continu beschikbaar. Dit vergroot de beheersbaarheid van een ontwikkelproject enorm.

Toekomst

De techniek en de business veranderen zo snel dat een scheiding tussen functionaliteit en techniek niet langer een wens, maar simpelweg een eis is om tot een duurzame oplossing te komen. Bedrijven focussen zich puur op hun dynamische business en software leveranciers richten zich volledig op nieuwe technologieën. Thinkwise Software Factory maakt die scheiding in het softwareontwikkelproces mogelijk. De klant heeft haar eigen technologie-onafhankelijke bouwtekening en de software factory bouwt automatisch het eindproduct.

De scheiding tussen functionaliteit en techniek is nu een eis in plaats van een wens