

/trends & hypes

Softwarekeurmerken: een trend of hype?

>>naast verificatie ook validatie noodzakelijk



Wie software laat ontwikkelen, wil een werkend systeem geleverd krijgen. Maar hoe kun je na oplevering weten of het systeem goed is ontwikkeld? Een keurmerk kan hier een basis van zekerheid voor bieden. Hoe gaat softwareontwikkeling in zijn werk en wat toetst een softwarekeurmerk precies?

De ontwikkeling en productie van maatwerksoftware wordt steeds vaker uitbesteed. Dan gaat het niet over een simpele teksteditor, maar over complexe systemen voor banken, ziekenhuizen, verkeersmanagement en dergelijke. Na een functionele aanbesteding wil de opdrachtgever bij oplevering zeker weten dat de software doet wat het moet doen. Vervolgens wordt het systeem jaren gebruikt en onderhouden. Op termijn kunnen nieuwe wensen ontstaan, waardoor de klant extra functies aan het systeem wil toevoegen. Als de software goed is gemaakt, is het relatief eenvoudig om nieuwe functionaliteit toe te voegen. Daarvoor moet de code duidelijk, helder en overzichtelijk zijn. Niet alleen voor degene die de code heeft geschreven, maar ook voor andere ontwikkelaars. Kan je een muurtje bijbouwen zonder dat het hele gebouw in elkaar stort?

Hulpmiddelen

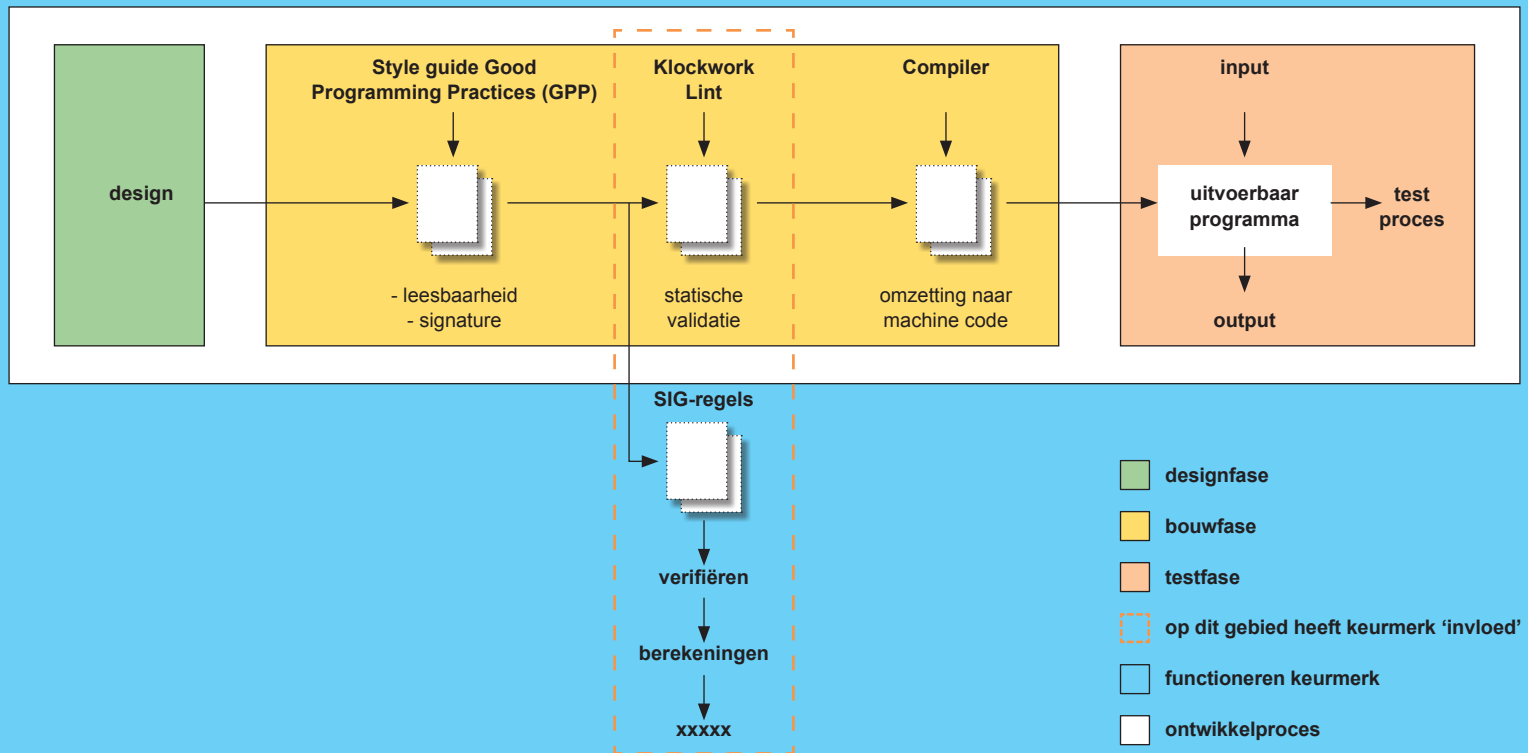
De structuur van de software en de manier waarop het geschreven is (de stijl) bepalen in grote mate de onderhoudbaarheid. Afzonderlijke functies dienen herkenbaar te zijn en snel aanpasbaar, zonder andere functionaliteit te beïnvloeden. Gelukkig bestaan er de nodige hulpmiddelen om duidelijke code te schrijven. Zoals er grammatica en stijlgidsen zijn voor taal, bestaan er regels en afspraken voor het correct opstellen van code. Die op hun beurt weer verwerkt zijn in speciale controleprogramma's als Klockwork en Lint. Deze gereedschappen analyseren de code op bekende valkuilen. "Ik snap wat de code moet doen, ik zie dat je hier een kast maakt van 10 planken en ik zie dat je iets op plank 11 probeert neer te leggen. Wees je daarvan bewust, doe daar wat aan."

Stijl en complexiteit

Andere aspecten waar de hulpprogramma's naar kijken, zijn de stijl en de grootte van de modules. Voor een computer maakt het niet uit hoe de code eruit ziet, zolang alles maar klopt. Maar goede ontwikkelaars zorgen voor overzichtelijkheid. Daar bestaan regels voor (samengevat in de Good Programming Practices) en stijlrichtlijnen. Bijvoorbeeld dat code die functioneel bij elkaar hoort, ook bij elkaar in één module staan. Zo worden functies gedefinieerd die vanuit het hoofdprogramma zijn aan te roepen. Wordt een functie te groot, dan knip je die weer op in aparte functies. Hoe complexer de code, hoe groter de kans op fouten. Een analyse met die insteek telt het aantal beslissingen per functie; de 'if_then' vragen. De analyse waarschuwt als dat er teveel zijn. Het compileren van de code is ook een controle. Pas wanneer alles volgens de regels van de desbetreffende taal is opgeschreven, zet de compiler de code om in machinetaal. De software is nu klaar om te gebruiken. Dan komen de functionele tests: doet het programma wat het moet doen?

Onafhankelijk keurmerk

De klant weet nu dat alles werkt. Maar welke kwaliteit heeft de software? Is de code later makkelijk te onderhouden en aan te passen? Dat is lastiger te bepalen. Precies in die niche is de Software Improvement Group (SIG) gedoken. Dit Nederlandse keuringinstituut geeft een onafhankelijk oordeel over de kwaliteit van softwareproducten in de vorm van een keurmerk volgens de ISO/IEC 9126 normering. Gekeken wordt naar de kwaliteit van



het ontwerp en het gevolgde proces tijdens de ontwikkeling. De architectuur wordt geanalyseerd, waarbij bijvoorbeeld gekeken wordt of de functionaliteit is opgedeeld in modules die begrijpbaar zijn. Maar ook naar hoe de ontwikkelaar tot het ontwerp is gekomen. Daartoe worden interviews met systeemexperts, de architecten en ontwerpers van de code gehouden: wat was je gedachtegang? Hoe heb je het opgesplitst? Wat moet het systeem doen en hoe heb je dat bereikt? En op welke manier test de designer zijn eigen ontwerp? Deze informatie wordt meegenomen bij het bekijken van de code, zodat geëvalueerd kan worden of deze gedachtegang en overwegingen terug te vinden zijn in het eindresultaat.

De kracht van het softwarekeurmerk is de vertaling van complexe code naar een eenduidig kwalificatiemiddel. Een groot systeem kan uit wel duizend modules bestaan die samen een softwareproduct vormen. Die worden door tientallen soorten tests gehaald. Zo veel resultaten kan niemand meer overzien. Het keuringsinstituut vertaalt deze hoeveelheid aan informatie tot een score van één tot vijf sterren, met daarbij eventueel een advies voor verbeteringen.

Goede code = flexibiliteit

Onderhoud is vaak de grootste kostenpost van een systeem, maar wordt nog wel eens over het hoofd gezien bij een uitbesteding. Hoeveel moeite het kost om later nog iets aan te passen, hangt af van de onderhoudbaarheid. In complexe, slecht onderhoudbare code is het veel moeilijker om een aanpassing in te voegen.

Na elke aanpassing is veel testwerk nodig om zeker te weten dat er geen andere functionaliteit is beïnvloed. Met goed onderhoudbare code kan een ontwikkelaar meteen naar de juiste module gaan, waar hij in een compact stukje code een wijziging aanbrengt.

Vakmanschap

Softwarekeurmerken zijn een trend, ze bieden een controlehandvat voor het uitbesteden van complexe softwaresystemen. De hoofdtaak van de klant ligt bij het verzorgen van een oplossing en aan welke functionele eisen deze oplossing moet voldoen. Hoe dit technologisch is opgebouwd, wordt niet tot in detail vastgelegd. Wat vast staat is dat het systeem aan een aantal eisen moet voldoen met betrekking tot onderhoud- en uitbreidbaarheid. Dan biedt een softwarekeurmerk een uitkomst om dit niveau te toetsen. Klanten van Technolution experimenteren met het softwarekeurmerk en hebben meerdere systemen die in ontwikkeling waren via de SIG beproefd. Technolution was niet bekend met de kwalificatiewijze van SIG, maar scoorde desondanks hoog in de SIG-ranking.

Uiteindelijk verifieert het softwarekeurmerk dus of de code netjes en overzichtelijk is geschreven, maar wordt er niet gevalideerd of het opgeleverde systeem ook doet wat ervan verwacht wordt. Daarvoor heb je nog steeds vakmensen nodig die met een gedegen ontwikkelstrategie en meerdere tussentijdse testen de voortgang en het uiteindelijke doel bewaken.