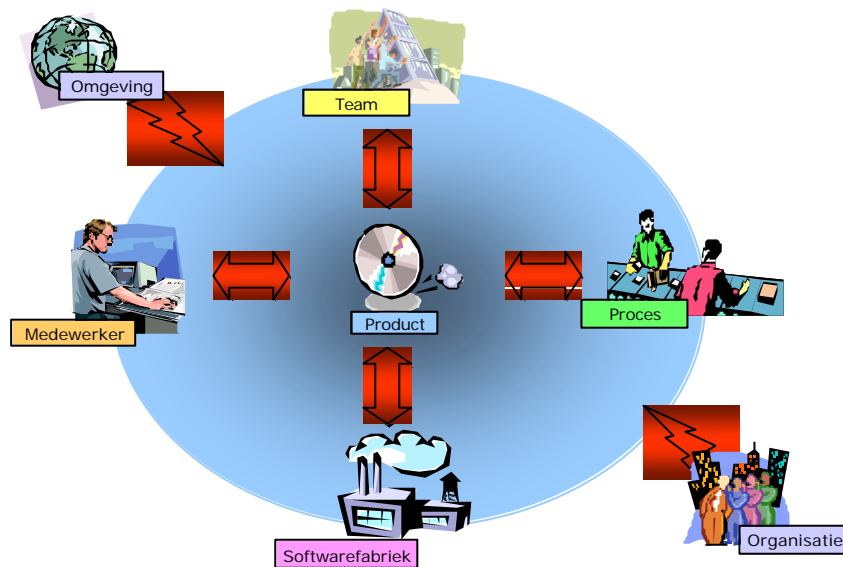


SOFTWARE QUALITY MODEL – SQM

Het valt niet te ontkennen dat het gebruik van softwarekwaliteitsmodellen gezorgd heeft voor aanzienlijke verbeteringen binnen organisaties. De meeste softwarekwaliteitsmodellen zijn ervaringen die in de praktijk hun waarden hebben bewezen. Daar ligt meteen ook de grootste kracht van deze modellen. Men hoeft niet meer te bewijzen dat de aanpak werkt, men hoeft er ‘alleen’ maar voor te zorgen dat de aanpak met verstand wordt gebruikt. De kracht van deze softwarekwaliteitsmodellen is verder dat er een poging wordt gedaan om een visie te geven van een ideaaltypische software engineering proces.

De meeste softwarekwaliteitsmodellen zijn typische exponenten van de procesbenadering van kwaliteit¹. In die opvatting is geen ruimte voor een expliciete aandacht voor overige factoren zoals productkwaliteit, immers een goed proces levert vanzelf goede producten. Helaas zijn er in de literatuur en binnen Info Support voorbeelden te over die het tegenovergestelde laten zien.

Door geen uitspraken te doen over bijvoorbeeld cultuur, kennis en kunde van ontwikkelaars, motivatie, of stijl van leidinggeven wordt het ook lastig na te gaan welk effect een ingreep in het proces heeft op de kwaliteit van het product (wordt het product beter of niet). De klant koopt immers een product en niet een proces.



Figuur 1: Software Quality Model

Aan de hand van praktijkervaringen binnen Info Support is er een Software Quality Model (SQM) ontwikkeld (figuur 1). De kracht van dit model is de integrale benadering van een aantal (minder bekende) factoren. Deze factoren zijn ingedeeld in vijf verschillende aandachtsgebieden die nauw met elkaar samenhangen en voortdurend met elkaar interacteren.

Deze aandachtsgebieden zijn: medewerker, team, proces, product en softwarefabriek. Met dit SQM-model krijgt een ICT-organisatie inzicht in de belangrijkste factoren die de kwaliteit en de daaruit voortvloeiende onderhoudskosten bepalen (applicatiebeheer).

Naast deze kwaliteitsbeïnvloedende factoren is er een context waarbinnen softwareontwikkeling plaatsvindt die bepalend is voor de softwarekwaliteit. Hierbij onderscheiden we twee categorieën van factoren, namelijk organisatie- en omgevingsafhankelijke factoren.

¹ Het geheel van eigenschappen en kenmerken van een product of dienst dat van belang is voor het voldoen aan vastgestelde of *vanzelfsprekende* behoeften (ISO 8402).

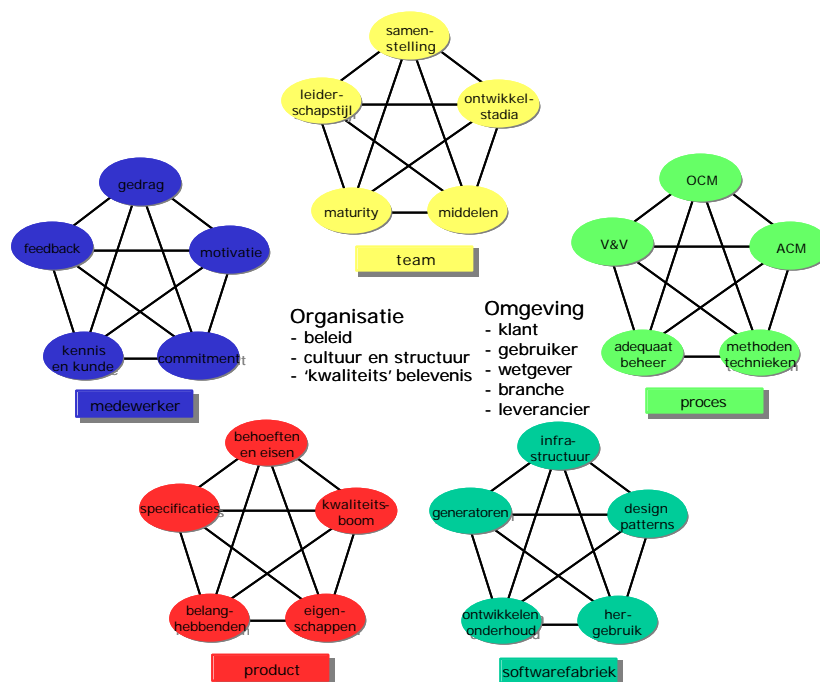
Het SQM-Model

Context

De context waarbinnen softwareontwikkeling plaatsvindt is verdeeld in twee categorieën; organisatie- en omgevingsafhankelijke factoren. Bij organisatieafhankelijke factoren gaat het om de invloed van bijvoorbeeld beleid, cultuur en structuur. Het betreft ook de mate waarin kwaliteitsbeleving in een organisatie is verankerd. Daarnaast is er de contextcategorie omgeving. Tot deze categorie kan de invloed van bijvoorbeeld klant, gebruiker, wetgever en branche worden gerekend.

Medewerker

Er kan nog zoveel geïnvesteerd worden in verbetering van processen, methoden en technieken. De investeringen leveren weinig op als de medewerkers die in deze verbeterde processen en met deze nieuwste methoden en technieken en tools moeten werken, niet in staat zijn adequaat hiermee om te gaan of niet bereid zijn hiermee te werken. Gebrek aan commitment, motivatie, feedback, slechte of verwaarloosde opleidingsprogramma's en onvoldoende mature gedrag zijn allemaal factoren die ervoor kunnen zorgen dat de kwaliteit niet gehaald wordt.



Figuur 2: Software Quality Model – overzicht van de belangrijkste factoren

Team

Bij het ontwikkelen van software streven we naar de 'beste' teams die in staat zijn 'de beste' software te maken. Veel problemen bij softwareontwikkeling zijn te wijten aan slecht functionerende teams. Dit is de reden dat het belang van teamwork en de inrichting en samenstelling van teams als een van de belangrijkste kwaliteitsbepalende factoren wordt onderkend. Een belangrijk onderdeel hierbij is dat de levenscyclus van een team gekoppeld wordt aan de verschillende fasen in de projectlevenscyclus. Het teamgedrag ofwel 'teammaturity' kan de kwaliteit van het softwareproces naar een hoger niveau brengen. Dit gedrag wordt gekenmerkt door: pro-actief, focus op oplossingen, enzovoort.

Leidinggevendens trachten medewerkers te motiveren om te veranderen door een positieve, leuke en uitdagende werkomgeving te scheppen en hen de vrijheid en de middelen ter beschikking te stellen die ze nodig hebben om goede software af te leveren. Een optimale werkomgeving met de modernste apparatuur en tools is voor softwareontwikkelaars een belangrijke motivatie om de werkzaamheden uit te voeren. Het besparen hierop is dus niet kwaliteitsbevorderend voor de software temeer deze uitgaven vaak een fractie zijn van de totale projectkosten.

Proces

Een eenzijdige aandacht voor uitsluitend ontwikkeling, zoals in de traditionele projectbeheersing gebruikelijk is, blijkt niet meer te voldoen. Projectbeheersing moet zich uitstrekken over de totale levenscyclus van de software, dat wil zeggen over ontwikkeling, gebruik en beheer. Projectbeheersing wordt aldus productbeheersing. Nog een stap verder gaat de beheersing van een heel scala van producten ('multiproductbeheersing'). Daarbij speelt hergebruik een cruciale rol. Bij de overgang van 'ontwikkelbeheersing' naar 'productbeheersing' spelen twee organisatie-aspecten een rol. Het eerste aspect richt zich op het onderkennen van een aparte functie binnen de organisatie, verantwoordelijk voor de evolutie van een softwareproduct over de totale levenscyclus, de product-manager. Het tweede aspect heeft te maken met de wijze waarop die functie moet worden georganiseerd.

Wanneer er wordt gesproken over de eisen die aan de informatievoorziening worden gesteld, daarbij lettend op de behoeften, wensen en eisen van de gebruikers en de eisen vanuit het bedrijfsproces of vanuit de bedrijfsvoering gesteld dan wordt uitsluitend de organisatie als uitgangspunt genomen. Echter, het beheer van de informatievoorziening moet niet alleen vanuit het perspectief van de bestaande organisatie worden gevoerd, maar ook vanuit de optiek van beheer en de invloed en de mogelijkheden van nieuwe technologieën op en voor de organisatie. Deze 'blik naar twee kanten' is een kenmerk van adequaat beheer van de informatievoorziening en is essentieel in de hedendaagse informatievoorziening.

Gesteld kan worden dat hierbij validatie en verificatie zeer noodzakelijk zijn om software met een redelijke tot goede 'aantoonbare' kwaliteit af te leveren. Daarom dienen altijd verschillende soorten validatie- en verificatie-activiteiten, bijvoorbeeld inspecties, walkthroughs en testen te worden gecombineerd.

Softwarefabriek

De gigantische toename van de omvang en complexiteit van software heeft tot gevolg dat ontwikkelteams steeds groter worden met als gevolg dat de communicatie binnen en coördinatie tussen die teams alsmaar moeilijker wordt. Om te kunnen omgaan met het fenomeen van grotere en complexere software hebben veel ontwikkelorganisaties naar oplossingen gezocht waarvan de softwarefabriek er een is. Een organisatie die zowel ontwikkeling als onderhoud en hergebruik beheerst, wordt getypeerd als een softwarefabriek. Een nobel streven, waarbij 'hergebruik' het sleutelwoord is. Belangrijke basiselementen binnen de softwarefabriek zijn infrastructuur, modelondersteunende generatoren en templates voor ontwerp en analyse.

Product

Kwaliteitszorg begint in een zo'n vroeg mogelijk stadium. Dat betekent dus vanaf de eerste ontwikkelfase (planning). Hierdoor worden vertragingen in een later stadium voorkomen. Er moeten goede specificaties worden geformuleerd. Goede specificaties zijn: ondubbelzinnig, volledig, verifieerbaar, consistent, wijzigbaar, traceerbaar en bruikbaar.

Verschillende auteurs hebben in de afgelopen decennia gestreefd naar een zo goed mogelijke beschrijving van software(product)kwaliteit. De overeenkomsten en verschillen van de verschillende opvattingen hebben de roep om een eenduidig en operationeel begrippenstelsel voor software-kwaliteit versterkt. Alleen standaardisatie lijkt tegemoet te kunnen komen aan die behoefte. Binnen de ISO/IEC 9126 standaard wordt een onderscheid gemaakt tussen interne softwarekwaliteit, externe softwarekwaliteit en softwarekwaliteit-in-gebruik. Met name met het recentelijk ontwikkelde begrip kwaliteit-in-gebruik wordt getracht nadrukkelijker aan te sluiten bij de dagelijkse praktijk van de klant.

Uitgaande van het onderscheid tussen kwaliteitsbehoeften, kwaliteitseisen en producteigenschappen kunnen er twee trajecten worden onderkend. Tijdens het eerste traject worden activiteiten verricht om kwaliteitseisen te kunnen specificeren (het specificatietraject). Tijdens het tweede traject worden activiteiten verricht om kwaliteitseisen te realiseren (het realisatietraject).

Voor meer informatie kan contact opgenomen worden met:

ing Richard Sweer MBA RI
richards@infosupport.com
mobiel: +31 (0) 6 53217193

Info Support b.v.
Einsteinstraat 22¹
3902 HN Veenendaal
tel: +31 (0) 318 55 20 20
www.infosupport.com