

Een kwaliteit-gedreven aanpak voor architectuurreview

Danny Greefhorst, Mark van Elswijk

Dit position-paper presenteert een praktische aanpak voor architectuurreview die is ontwikkeld bij het Software Engineering Research Centre. Centraal in deze aanpak staat het redeneren over en vanuit softwarekwaliteit.

Inleiding

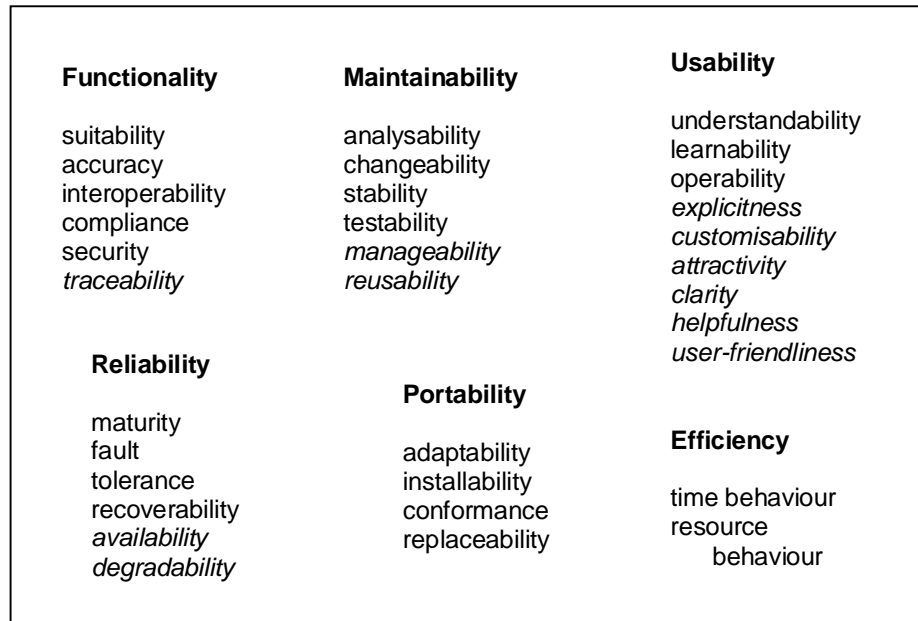
Het Software Engineering Research Centre (SERC) helpt bedrijven bij het verbeteren van de kwaliteit van software. In die rol is SERC ook vaak betrokken bij het beoordelen van software-producten, waarbij software-architectuur een belangrijke rol speelt. Op basis van ervaringen bij dergelijke projecten is er dan ook een eigen aanpak voor het reviewen van software-architecturen ontwikkeld. In deze aanpak staat software kwaliteit centraal en worden workshops, interviews en productreviews gebruikt voor de beoordeling van deze kwaliteit. Belangrijk hulpmiddel bij deze productreviews is een vragenlijst die de verschillende aspecten van het te beoordelen systeem belicht. Dit artikel zal de uitgangspunten van de aanpak nader toelichten.

De kwaliteit van een software-product is een relatief begrip dat tastbaar dient te worden gemaakt door aan te geven welke eigenschappen dat product zou moeten hebben. Het QUINT model [Quint96], dat gebaseerd is op het ISO 9126 model en mede door SERC is opgesteld, biedt een raamwerk van dergelijke eigenschappen, inclusief indicatoren en meetsvoorschriften. Indicatoren geven hierbij een indicatie van de mate waarin een eigenschap vertegenwoordigd is in een product. Er zijn in totaal 32 eigenschappen in het QUINT model gedefinieerd, die zijn gegroepeerd onder zes hoofdeigenschappen, te weten: functionality, reliability, useability, efficiency, maintainability, en portability (zie ook figuur 1).

Er zijn vele definities van software-architectuur in omloop, maar je zou eenvoudig gezegd kunnen stellen dat software-architectuur gaat over de belangrijkste keuzen die in een systeem dienen te worden gemaakt. Hierbij speelt de onderverdeling van het systeem in onderdelen, hun relaties en verantwoordelijkheden een belangrijke rol. Belangrijk is om op te merken dat er veel verschillende soorten onderdelen in een software-architectuur te onderkennen zijn, afhankelijk van de wijze waarop je naar het systeem kijkt. Voorbeelden

van dergelijke views op software-architectuur zijn een conceptuele view die redeneert over domein-specifieke concepten, een execution view die gaat over runtime onderdelen en een code view die gaat over broncode elementen.

Het is de verantwoordelijkheid van de software-architectuur om richting en (gedeeltelijke) invulling te geven aan de kwaliteitseisen die gesteld worden aan een systeem.



Figuur 1: Kwaliteitsattributen in het Quint2/Extended ISO 9126 model.

De aanpak

De specifieke aanpak van een architectuurreview is natuurlijk erg afhankelijk van de situatie. Een belangrijke factor hierbij is het doel van de architectuurreview. In onze ervaring kan de nadruk liggen op

- het bepalen van de kwaliteitseisen die aan een architectuur in ontwikkeling worden gesteld
- het bepalen van de kwaliteit van een architectuur en bijbehorende artefacten
- het bepalen van de impact van wijzigingen op een architectuur.

Om deze doelen te verwezenlijken kunnen een aantal middelen worden ingezet zoals workshops, interviews, productstudies en experimenten.

Een workshop heeft typisch tot doel de (kwaliteits)eisen die aan het systeem worden gesteld in kaart te brengen, en dient bijgewoond te worden door alle bij het systeem betrokken

partijen/stakeholders. Het QUINT model kan als kader dienen voor het structureren van de discussie over kwaliteitseisen. Een workshop kan ook worden gebruikt om toekomstige wijzigingsscenario's te bepalen. De uitwerking van deze scenario's kan buiten de workshop plaatsvinden.

Interviews worden gebruikt om het beeld van de verschillende partijen van de architectuur te bepalen, en om de issues omtrent de architectuur in kaart te brengen. Het voordeel van interviews is dat betrokkenen sneller geneigd zijn hun opinie met betrekking tot het systeem te geven. De in de volgende paragraaf beschreven vragenlijst kan gebruikt worden als basis voor het interview en een eventuele enquête vooraf.

Het doel van productstudies is om de kwaliteit van de architectuur en bijbehorende documenten en producten te bepalen. Een belangrijk onderdeel van deze studies is de vragenlijst waarbij naast architectuurdocumentatie ook andere documenten en producten worden beoordeeld.

Experimenten en prototypes kunnen de haalbaarheid van oplossingsrichtingen toetsen. Ook kunnen experimenten en prototypes worden gebruikt om belangrijke verbeteringen in bestaande knelpunten te demonstreren.

Vragenlijst

Een belangrijk onderdeel van onze aanpak voor architectuurreview is een vragenlijst. Deze vragenlijst dekt naast de architectuur ook andere artefacten in het ontwikkeltraject zoals het programma van eisen, het ontwerp en de code. Het programma van eisen is van belang omdat deze de richting voor het systeem en de onderliggende architectuur bepaalt. Het is verder onzinnig over de kwaliteit van een architectuur te redeneren als de kwaliteitseisen zelf niet goed gedefinieerd zijn. Ontwerp en code zijn ook aandachtspunten in de vragenlijst omdat enerzijds de scheidslijn tussen architectuur en ontwerp niet duidelijk te trekken is, en omdat anderzijds een architectuurreview vaak plaatsvindt in de context van een algehele product-review.

De vragenlijst is grofweg onderverdeeld in vragen per type artefact. Zo zijn er dus vragen specifiek voor eisen, architectuur, ontwerp en code. Daarnaast zijn er een aantal algemene vragen gedefinieerd die algemene aandachtspunten zoals volledigheid, consistentie, begrijpelijkheid en traceerbaarheid verwoorden.

Het belangrijkste onderdeel van de vragenlijst is het deel dat betrekking heeft op de architectuur. De focus ligt hierbij op een beschrijving van die architectuur die de reviewer beschikbaar heeft, wat dus niet noodzakelijk overeenkomt met de architectuur van het systeem zelf. Naast algemene vragen over architectuur zijn er twee belangrijke groepen met vragen te onderkennen. Aan de ene kant zijn er vragen gedefinieerd die specifiek gaan over de mate waarin de architectuur tegemoet komt aan de kwaliteitseisen. Deze vragen zijn dan ook gecategoriseerd volgens de hoofdeigenschappen van het QUINT model. Zo is er bijvoorbeeld onder maintainability de vraag: 'is functionaliteit in het systeem gedupliceerd

op meerdere plaatsen?'. De andere belangrijke categorie van vragen heeft betrekking op specifieke views op de software-architectuur. Het doel van deze vragen is te bepalen of belangrijke informatie voor deze views wel aanwezig is in de architectuurbeschrijving.

Conclusies

De door SERC voorgestelde aanpak wordt gestuurd door de kwaliteitseisen die aan een systeem worden gesteld. Gebruikmakend van instrumenten als workshops, interviews, productstudies en experimenten wordt hierbij bepaald of de voor het systeem beschreven architectuur tegemoet komt aan de vooraf opgestelde kwaliteitseisen. Een belangrijk middel hierbij is een vragenlijst, die naast de architectuur ook aandacht besteedt aan andere relevante software-ontwikkelartefacten.

Literatuur

- [Bass98] L. Bass, P. Clements, and R. Kazman, "Software Architecture in Practice", Published by Addison-Wesley in the SEI Series, 1998.
- [Quint96] B. van Zeist, et.al., "Kwaliteit van software producten, praktijkervaring met een kwaliteitsmodel", Kluwer Bedrijfsinformatie, 1996.