

# Management en Architectuur in embedded systems ontwikkeling

*Bij de ontwikkeling van een complex product zijn zowel het project management als een goede systeem architectuur van belang. Welke van de twee is nu het belangrijkste? Zijn er veel verschillen of juist overeenkomsten? Hoe zit het met verantwoordelijkheden en processen? En wat is het belang van communicatie?*

Task Switch, Ger Schoeber

## Organisatie

Aan ieder project ligt een projectorganisatie ten grondslag. Zijn de projecten klein dan zal ook het organigram eenvoudig zijn. Of zoals Hans Sassenburg onlangs zei op het Bits&Chips event -Making software work-: “the cheapest way to make software, is having one engineer” [Bits&Chips]. De communicatie en overlegstructuur voor een eenmansproject is uitiem eenvoudig en efficiënt. Groeit het aantal engineers in een project dan neemt uiteraard de behoefte aan organisatie en communicatie toe. Dat dit ook een ander uiterste kent hebben we allemaal wel eens gelezen in het boek “The Mythical Man-Month” [Brooks]. Met uit dit boek het beroemde statement: “Adding manpower to a late software project makes it later”.

De laatste jaren zien we vaak het patroon dat een project gestuurd of geleid wordt door een duo: een projectleider samen met een architect. Kan een projectleider het niet meer alleen af? Wie is de echte eindverantwoordelijke? Wat is de toegevoegde waarde van een architect?

## Projectleider

Bij de start van een project wordt een projectleider aangesteld als verantwoordelijke voor de realisatie. De realisatie is afgekaderd door een viertal grenzen: functionaliteit, moment van oplevering, kosten en kwaliteit. Het is de verantwoordelijkheid van de projectleider om binnen deze gestelde grenzen het project te realiseren. Accepteert de gevraagde projectleider de opdracht, dan verplicht hij of zij zich tot het realiseren ervan conform de gestelde eisen. Ervaart de projectleider de risico's voor realisatie als te groot, onacceptabel of onverantwoord, dan ligt het ook in zijn verantwoordelijkheid de opdracht als zodanig niet te accepteren. Is de projectleider in staat deze afweging voldoende gefundeerd te maken?

## Architect

In een dergelijke initiële fase zien we reeds een architect zijn intrede doen. Zijn technische expertise is zeer welkom om een gedegen afweging van de risico's te kunnen maken. Wat betekent het om een digitale TV te bouwen? Hoe complex zijn de interdisciplinaire relaties in een digitale kopieermachine? Welke nieuwe afhankelijkheden komen naar voren bij de integratie van camera, bluetooth, wifi, agenda en mobiele telefoon? Hoe zit het met de veiligheid in een auto bij de enorme toename van infotainment en communicatie op weg naar drive-by-wire? De grote groei van embedded computing power, de onderlinge integratie ervan plus de ontsluiting richting internet maakt het schatten ervan tot een Betuwelijn- of HSL-ervaring.

Hier zien we al in de beginfase van een project de belangrijke inbreng van de kennis van de architect ter ondersteuning van de projectleider. Prof. dr. Daan Rijsenbrij zei onlangs bij zijn aanvaarding van het hoogleraarschap aan de Universiteit van Nijmegen, dat de architect “zijn gewicht in goud” meer dan waard is in de digitale wereld [Automatisering Gids].

## Verantwoordelijkheden

Hoe zit het nu met verantwoordelijkheden? Zoals gezegd aanvaard de projectleider de eindverantwoordelijkheid bij de acceptatie van een project. Wat zijn dan de verantwoordelijkheden van de architect? We zouden kunnen zeggen dat de projectleider de organisatorische verantwoordelijkheid draagt en de architect de technische verantwoordelijkheid. Organisatorische verantwoordelijkheid betreft budget, mensen, middelen en time-to-market. De technische verantwoordelijkheid behelst functionaliteit en kwaliteit van het te leveren product. Slechts

zelden wordt echter een product in isolatie gebouwd. Ik bedoel hiermee dat een product zeer vaak een onderdeel van een lijn van producten is. En hier treedt een spanningsveld op. Keuzes en beslissingen van technische aard (lees 'architectuur') hebben een belangrijke invloed niet alleen voor het huidige product/project, maar ook voor de volgende producten die in dezelfde familielijn zitten.

Bijvoorbeeld de ontwikkeling van een low-cost DVD recorder. Aangezien de opvolger conform de marktontwikkeling minimaal al weer USB (on-the-go) connectivity zal moeten bieden kan dit een belangrijk aandachtspunt zijn bij het ontwerp van de low-cost versie waarvoor deze functionaliteit niet gevraagd is. Bedrijfseconomisch gezien is het een goede beslissing om de USB functionaliteit nu al in de architectuur van de controller chip te integreren. Organisatorisch gezien betekent het wel extra werk, tijd en geld. En daar had de projectleider niet op gerekend.

Waar de verantwoordelijkheid van een projectleider helder afgebakend is door de grenzen van functionaliteit, tijd, geld en kwaliteit, komt deze hier in botsing met de verantwoordelijkheid van de architect. Zijn beslissingen overschrijden de grenzen van het huidige project. Het betreft het eeuwige gevecht tussen korte en lange termijn. Conform de woorden van Stephen Covey: "Mensen zijn vaak zo druk bezig met kreupelhout te kappen dat ze zich niet eens afvragen of ze wel in de goede jungle zitten" [Covey].

### **Processen en methodes**

Naast de inrichting van de organisatie worden ook de nodige processen opgetuigd voor de uitvoering van een project. Processen helpen om de uitvoering van een project te beheersen. Aan de management kant wordt al snel gevraagd voor het inzetten van een Prince2 certified project manager. Prince2 (Projects in controlled environments) [Bentley], is een methode voor algemeen projectmanagement voor allerlei soorten van projecten. De methode biedt geen expliciete ondersteuning voor systeem- of software ontwikkeling. Het projectmanagementproces wordt volgens Prince2 onderverdeeld in acht hoofdprocessen, verdeeld over management fasen en technische fasen. De stuurgroep en projectmanager voeren controle over de managementfasen. In de technische fasen worden duidelijke deelproducten gedefinieerd en gerealiseerd.

Vanuit een technisch perspectief neemt RUP (Rational Unified Process) [IBM] een grote vlucht. RUP richt zich met name op de systeem- en softwareontwikkeling. De insteek bij RUP is om alle risicovolle zaken op het technische en functionele vlak zo snel mogelijk te realiseren. Op zich kan het zelfs zinvol zijn beide methodes geïntegreerd toe te passen. Hierbij kan Prince2 gebruikt worden als raamwerk en RUP voor de verdere invulling er van [Helm, Harbrecht].

Het aardige van Prince2 is dat de nadruk ligt op bedrijfseconomische overwegingen (kosten versus baten). Dit sluit extra goed aan bij de economische situatie waarin een groot aantal bedrijven sinds een aantal jaren verzeild zijn geraakt. De iteratieve aanpak die RUP voorschrijft ondersteund in die zin het op tijd kunnen ingrijpen. Dit helpt het voorkomen van verrassingen achteraf.

Ook het werk van de architect kan grote baat hebben bij de inzet van een raamwerk of methode. Vanuit de technische productverantwoordelijkheid (functionaliteit en kwaliteit) is het voor een architect enorm belangrijk het probleem van zijn klant volledig te begrijpen. Wanneer de klant wordt geraadpleegd voor requirements, dan is het onvoldoende om puur de gevraagde requirements over te nemen en daar het product op te baseren. Het is de taak van de architect te begrijpen waarom de klant de betreffende requirements heeft. Het is de kunst om in de schoenen van de klant te gaan staan en vanuit zijn gezichtsveld het probleem te aanschouwen. Dit is de enige manier om, *niet* het product te maken waar de klant om vraagt, maar *wel* het

product te realiseren waar de klant behoefte aan heeft. Een aardig raamwerk hiervoor is het CAFCR model [Muller]. CAFCR bekijkt de architectuur vanuit een 5-tal gezichtspunten die hier kort worden toegelicht.

Customer objectives. Wat is belangrijk voor de klant? Wees bewust dat “klant” hier een zeer breed begrip is. Afhankelijk van het te realiseren product of systeem is er een hele reeks van belanghebbenden die op een bepaalde manier als klant gezien kunnen worden. Dit loopt uiteen van eindgebruiker, dealer, inkoop, verkoop, productie, onderhoud, etc.

Application. Hoe is de toepassing van het product, hoe wordt het product door de klant gebruikt? Deze eerste twee views zijn zeer sterk gefocuseerd op de klant. Met andere woorden: wat heeft de klant nodig en vooral ook: waarom.

Functional. Dit beschouwd de functies en de kwaliteit die het product moet gaan bieden. Denk hierbij aan zowel de functionele als ook de niet-functionele requirements of de product specificatie. Ook wel de “wat” van het product genoemd.

Conceptual. Dit gaat in op hoe het systeem werkt. De technische architectuur en haar concepten.

Realisation. Hoe is beschikbare technologie ingezet voor de realisatie van het product. Deze laatste twee views richten zich sterk op de “hoe” van het product oftewel het ontwerp.

De architect heeft de belangrijke verantwoordelijkheid om tot een realisatie van een product te komen wat uiteindelijk de behoeftes van de klant zo goed mogelijk afdekt. Dit vereist een zeer goed inlevingsvermogen om de behoeftes te kunnen begrijpen (let wel: begrijpen gaat verder dan aanhoren en opnemen). De praktische toepassing van het CAFCR raamwerk is niet een top-down of bottom-up benadering, maar juist een combinatie. De customer kant (CAF-CR) geven een behoefte of rechtvaardiging van de onderwerpen aan de CAFCR zijde. De technische kant (CAFCR) maken de onderwerpen aan de CAFCR zijde juist mogelijk. Door iteratie worden alle gezichtspunten zo helder en compleet mogelijk belicht. Iteratie waarborgt hierbij een goede samenhang.

Het resultaat zou het ontwerp kunnen zijn van een product wat kan wedijveren met de concurrentie qua time-to-market en bill-of-material (business perspectief retailer), wat eenvoudig en intuïtief in haar gebruik is (eindgebruiker of operator perspectief), wat goed te ontwikkelen, te fabriceren en te testen is (productie perspectief) en met voldoende technische innovatie voor toekomstige opvolgers in de lijn van producten (business perspectief eigen organisatie).

Als we de verschillende modellen en processen naast elkaar zetten dan zien we interessante aanknopingspunten tussen de management modellen en het besproken architectuur raamwerk. Zo heeft de iteratieve aanpak van het CAFCR raamwerk raakvlakken met RUP. Dit betreft met name de inception, elaboration en construction fases van RUP. Daarnaast vormt het expliciete begrip van de klant zijde (customer en business objectives) een aardige aansluiting bij Prince2 (de business case en feasibility study). Hieruit blijkt dat de verschillende behoeftes, modellen en processen van enerzijds de projectleider en anderzijds de architect prima met elkaar te verenigen zijn. Het is de kunst om de onderwerpen zodanig te combineren dat ze elkaar ondersteunen en versterken.

## **Communicatie**

Wat bij Prince2 expliciet buiten de deur wordt gehouden zijn de sociale en communicatieve vaardigheden. Dit maakt het tot een wat mechanische methode. Hierin onderscheid CAFCR zich in positieve zin. Er wordt expliciet aandacht besteed aan een zo goed mogelijke communicatie tussen de verschillende belanghebbenden. Het middel wat hiervoor aangedragen wordt is “story telling”. Het nodigt uit om de product definitie zo snel mogelijk te koppelen met het concrete en feitelijke domein.

Zoals eerder gezegd, blinken kleine projecten uit in efficiency door de korte communicatielijnen. Dit betekent dat juist in de hedendaagse, complexe en grote projecten meer dan expliciete aandacht voor onderlinge communicatie en afstemming nodig is. Dit betreft communicatie en rapportage door de projectleider richting zijn management en stuurgroep en communicatie over business, markt en status richting de engineers. Dit geldt even zo zeer voor de architect. Veel communicatie en afstemming met het brede scala aan belanghebbenden, maar ook het uitdragen van visie en richting van de technische oplossing richting de engineers. En hierin wordt het verschil gemaakt naar een geslaagd projectresultaat: de optimale aandacht voor communicatie binnen en buiten het project door zowel de projectleider als de architect.

Ger Schoeber is consultant en eigenaar van Task Switch. Task Switch levert diensten op gebied van coaching, management, architectuur, workshops en assessments ([www.task-switch.nl](http://www.task-switch.nl)).

## Referenties

- [Bits&Chips] Hans Sassenburg – “Software Releasing: Politics or Numbers”, [http://www.bits-chips.nl/events.asp?mode=&ac=view&evt\\_id=23&tab=2](http://www.bits-chips.nl/events.asp?mode=&ac=view&evt_id=23&tab=2).
- [Brooks] Frederik Brooks – The Mythical Man-Month, Addison-Wesley, 1975.
- [Automatisering Gids] “Goede IT-architect is goud waard”, 1 oktober 2004.
- [Covey] Stephen R. Covey – “De Zeven Eigenschappen van Effectief Leiderschap”, Business Bibliotheek, 1993.
- [Bentley] Colin Bentley – “Prince2, A practical handbook”, 1997, Butterworth-Heinemann.
- [IBM] Rational Unified Process, <http://www-306.ibm.com/software/awdtools/rup/index.html>.
- [Helm, Harbrecht] Gerard van der Helm en Dik Harbrecht – “Prince2 RUP”, Automatisering Gids, 1 oktober 2004.
- [Muller] Gerrit Muller – CAFCR: A Multi-view Method for Embedded Systems Architecting, 2004.