Jos Warmer

Voor de zomervakantie was ik als jurylid betrokken bij de RAD Race. Eigenlijk een woord dat niet meer helemaal de lading dekt. RAD suggereert een beetje dat er vooral snel, maar niet altijd netjes gewerkt wordt. Dat geldt voor veel van de in de RAD Race ontwikkelde software al lang niet meer.

Van U/FP naar FP/U

Het eerste aantal teams in de Rad Race bereikten een productiviteit van ongeveer zeven *functiepunten per uur*. Als je nagaat dat in projecten tussen de vier en acht *uren per functiepunt* gerekend wordt dan blijkt dat er een verschil is van minimaal een factor dertig!

Natuurlijk zullen de critici zeggen: "Met twee mensen snel iets in elkaar zetten is heel anders dan een *echt* project met veel mensen doen". Klopt, maar dat kan nooit een factor dertig verschil verklaren.

Neem een project dat geschat is op 560 functiepunten. Als project zou er zo een 4 * 560 = 2240 uur ofwel 56 weken mee gemoeid zijn. De klant wil graag over een week of zes klaar zijn, dus dat wordt een project van 56 / 6 = 9.33 fulltime mensen. Voor zo'n kleine applicatie is dat een enorm team. Bovendien moeten we hier projectleiding à 15% bij optellen dus dat komt ver boven de tien mensen. Stel dat een team van twee mensen met een productiviteit als in de Rad Race aan de gang gaat. Dat kunnen deze mensen met zijn tweeën in een week klaar zijn. Dat is niet alleen ruim op tijd, maar omdat het team lekker klein blijft is een groot project met alle potentiële problemen als overhead of miscommunicatie niet aan de orde.

Ofwel, bij een structureel hogere productiviteit worden projecten lang niet zo snel groot en zullen de voordelen van kleine teams in veel meer gevallen automatisch gevoeld worden. Wij maken zelf projecten groot door onze lage productiviteit.

Een tweede aspect dat opvalt aan de RAD Race is dat de mensen een hoog niveau hebben, ze begrijpen waar ze mee bezig zijn en kennen de door hen gebruikte tools van binnen en buiten. Mijn conclusie hieruit is dat het loont om te zorgen voor goede opleiding van ontwikkelaars. Zowel in conceptuele zin, als in het begrijpen en bedienen van de tools.

Iets dat niet zo direct opvalt, is het feit dat er in de RAD Race geen verschil in competenties gemaakt wordt. De ontwikkelaars praten zowel met de klant als de gebruiker, nemen zelfstandig technische beslissingen en ontwerpen het uiterlijk van de applicatie. Geen aparte user interface experts, aparte requirements analisten, aparte ontwerpers en aparte bouwers. Nee, alle mensen beheersen alle competenties. Dit is een aspect dat in de verschillende agile methoden ook sterk gepropageerd wordt. Als een klein team alle aspecten van systeemontwikkeling beheerst kunnen ze sneller knopen doorhakken, beter communiceren en zal een veel groter begrip van de applicatie als geheel bijdragen tot goede beslissingen en een goede applicatie.

Een laatste aspect dat me opvalt in de RAD Race, is dat er veelvuldig gewerkt wordt met allerlei vormen van generatietools. Je kunt natuurlijk niet alles genereren, maar het blijkt dat veel van de code die geschreven wordt toch telkens weer meer van hetzelfde is. Dat vraagt natuurlijk om een slimme benadering waarbij je als ontwikkelaar al dit soort standaardpatronen niet meer zelf hoeft uit te schrijven. De echt unieke code schrijf je er vervolgens met de hand bij.

De omstandigheden bij de RAD Race zijn natuurlijk niet hetzelfde als bij een 'echt' project. Ik denk echter dat een verschil van een factor 30 niet alleen hieruit te verklaren is.

Ik stel voor dat we met de grote IT-bedrijven als doelstelling nemen om in de toekomst niet meer over *uren per functiepunt*, maar over *functiepunten per uur* te spreken. Dat betekent dat we slechts een factor 4 hoeven te verbeteren en dat de deelnemers aan de RAD Race nog steeds zeven keer zo productief zijn. Dat is toch niet teveel gevraagd?

Jos Warmer
Partner Ordina SI&D.
E-mail: jos.warmer@ordina.nl.