

Kanjers

16 september 2008

Games volgen elkaar in snel tempo op. De ontwikkelingen moeten dus altijd snel, snel, snel. Liever gisteren dan morgen. Tijd om na te denken is er nauwelijks. De toekomst? Die begon gisteren. En toch leren we steeds beter om na te denken over morgen, over volgende week, over hoe we volgend jaar spellen willen maken. Want één ding is zeker: ooit komt er een volgend platform. We willen immers nog sneller en efficiënter spellen maken en dan is een geweldig ontwikkel- en softwareplatform essentieel. We moeten dus leren om daarover na te denken.

Onlangs las ik de managementboeken 'Good to great' en 'Build to last' van Jim Collins over hoe je een geweldig bedrijf maakt. Ze zijn inspirerend en zetten aan tot nadenken. Zo is goed een belemmering om geweldig te worden. Eigenlijk zijn ze ook triviaal. Als het bedrijf eenmaal geweldig is, lijkt het zó gemakkelijk te gaan. Dat is ook in de sport. De Olympische Spelen lijken moeiteloos te verlopen voor de echte kanjers.

Momenteel werk ik aan het platform dat we gaan gebruiken tussen 2010 en 2020. Maar zorg ik ervoor dat het platform dan een kanjer is? Wáárom zou het dan geweldig zijn om dáár spellen op te maken? 'Als het moeiteloos gaat', hoor ik jullie denken. Ja, maar hoe bereik ik dat? Goede techniek is nodig, maar niet voldoende. Goede mensen zijn essentieel. Het zijn topsporters die de techniek optimaal weten te gebruiken. Dat lees je ook in die managementboeken. Je moet de goede mensen hebben. Ook daar denk ik over na. Hoe krijg ik mensen die nadenken over hoe ze die geweldige techniek willen gebruiken? Voor de programmeurs onder ons, dat is metadenken. Voor anderen is 't filosofie: denken over denken. Of quatsch, flauwekul en onzin, als je een heel nuchtere technicus bent.

Ik denk nu na over embedded computers die we over een decade gaan gebruiken, maar kan een architect van nu wel tien jaar vooruitkijken? Als iemand tien jaar geleden de huidige embedded-trends zoals Linux, iPhone en gigageheugens zou hebben voorspeld, zouden we dat niet als onzin hebben afgedaan?

Trouwens, wat is een architect? Is dat iemand die naar de toekomst kijkt of is dat iemand die geschoold is in het verleden? De typische architect is wat ouder, heeft waarschijnlijk leren programmeren in Basic of Pascal, in een tijd dat de Commodore C64 een topcomputer was. Wel 64 kbyte Ram! Is zo weinig geheugen eigenlijk nog te koop? Toen hij ging studeren, was C++ misschien net uitgevonden, begon de industrie na te denken over OO en wist iedereen het zeker: 640 kbyte is voor altijd voldoende. Moet je die, oudere, architect eigenlijk niet continu wantrouwen? Wat weet hij nu van moderne talen als Java, Python of XSLT? Laat staan van de laatste ontwikkelingen op het gebied van compilers of programmeerconcepten? Weet hij wel dat www.wikipedia.nl al 82 concepten kent?

Hoe voorkom ik dat ik een platform bedenk dat tien jaar geleden geweldig zou zijn? Hoe voorkom ik dat ik modernismen verwerp omdat ze vroeger nog niet bestonden? Of is het nadenken over die valkuil al voldoende om hem te vermijden? Architectuur mag nooit slechts over techniek gaan. Net als in de sport zijn technologische innovaties niet voldoende. Trainen blijft belangrijk. Een complete architectuur gaat ook over mensen, over werkwijzen en hoe de organisatie kan groeien. Vrij naar Jim Collins: leg het fundament om te groeien, zoek of train de juiste mensen en richt je in op verbetering. Wie continu verbetert, wordt ooit geweldig. Als je dat beter doet dan alle anderen, word je een kanjer.

Albert Mietus

[Terug naar overzicht](#)



Albert Mietus is R&D-architect bij JVH Gaming.