



Sander Hoogendoorn
 blog.sanderhoogendoorn.org

Dat is schrikken! Als gevolg van de immer toenemende vraag naar software en het dalen van het aantal studenten die software development willen leren, moet de productiviteit van ontwikkelaars stijgen. Om precies te zijn: honderd keer. Althans, dat voorspelt Gartner.

Honderd keer. Dat is niet mis. Nu houd ik me al zo'n twintig jaar bezig met het vernieuwen van software development, maar ik heb werkelijk geen flauw idee hoe we dit moeten gaan waarmaken. .

Someday maybe

Het toepassen van bijvoorbeeld agile software development, model driven development, gestandaardiseerde architecturen, frameworks, design patterns, het stimuleren van hergebruik en ga zo maar door helpt ons best een eindje op weg. Maar daarentegen is de complexiteit van de technologie in die twintig jaar ook schrikbarend gestegen

Kraken en piepen

Want zeg nou zelf. Toen we vijftien jaar geleden client-server applicaties ontwikkelden met PowerBuilder haalden we een productiviteit van ongeveer twee uur per functiepunt. Die productiviteit halen we in ons huidige tijdperk slechts heel af en toe. Met kraken en piepen. Niet dat vroeger alles beter was, maar wel makkelijker – het zijn net belastingen. Waar we met PowerBuilder altijd voor Windows ontwikkelden, schrijven we nu voor het web, *en* voor het web 2.0, *en* voor mobiele devices. En waar vroeger 'gewoon' een database onder onze applicaties lag, is dat nu een enorme variëteit aan bronnen, van SAP tot web services, een enterprise service bus, en zelfs zo nu en dan een verdwaalde database. Die dan nog wel stamt uit het client-server tijdperk. Alle goede bedoelingen ten spijt gaan we die honderd keer productiviteitsverbetering voorlopig niet halen uit het verbeteren van de tools die we inzetten.

Dus moeten we alle middelen inzetten om de productiviteit in projecten te verbeteren. Eerst onze persoonlijke productiviteit eens bekijken. Net als u word ik dagelijks bedolven onder grote hoeveelheden mail, en komen er tig keer per dag mensen mijn kamer binnenlopen die iets willen – van kennismaken tot teksten voor offertes of schattingen van nieuwe projecten. Allemaal nuttig natuurlijk, maar ook allemaal aanleiding om het werk dat ik eigenlijk moet doen, zoals

het schrijven van deze column, het maken van slides voor een komend seminar of zelfs het schrijven van code, uit te stellen.

Stuff

Totdat mijn collega Robert mij een boek in handen duwt. 'Getting things done' heet het. Erg Amerikaans. Gouden letters en auteur David Allen zuur glimlachend strak in pak op de kaft. Dokter Phil. Maar dan voor persoonlijke productiviteit. De filosofie is eenvoudig. Maak je hoofd leeg, zodat je dat kunt gebruiken voor werk. In mijn geval niet eenvoudig; ik probeer minstens vijftig balletjes tegelijk in de lucht te houden. Speciaal voor mij biedt David Allen een overzichtelijk werkschema. Alles dat binnenkomt is 'stuff'. Het probleem daarmee is dat *stuff* zich in je hoofd nestelt om er ooit iets mee te doen. Zet daarom alle *stuff* om in acties, of als er geen acties aan verbonden zijn, ruim het op. Je gooit het weg, slaat het op ter referentie of benoem het als 'Someday maybe', wat inhoudt dat ik er ooit iets mee doe, maar vooral niet *nu*. Als mijn code nu niet klopt vertaal ik het in: Refactor Customer Service en koppel deze actie aan een project. Nu is het benoemd en het kan weg. Als een actie binnen twee minuten is uit te voeren, doe het dan direct, schrijft Allen. In andere gevallen plan je het in – in je agenda, niet in je achterhoofd – of delegeer je het. Simpel toch?

Maar het werkt. En dus plan ik sinds kort tijd in mijn agenda om acties uit te voeren, in plaats van me te laten leiden door alles dat binnendruppelt via mijn Outlook-infuus. Het grappige van 'Getting things done' is dat dit allemaal – inclusief werkschema – in de eerste vijftig pagina's van het boek staat. Herkenbaar en soms pijnlijk. En de rest van de tweehonderdvijftig pagina's die het boek telt? Someday Maybe. «