Hoe verder?

Na het lezen van het adviesdocument heeft u waarschijnlijk de smaak te pakken. Naast de technische infrastructuur is er nog een heel gebied waarop een bedrijf moet letten om goed te kunnen samenwerken en het werk op een professioneel niveau te houden. In dit artikel komen een aantal onderwerpen aan bod om hierbij te helpen.

1 Afspraken voor samenwerking

Als er met veel mensen aan dezelfde projecten gewerkt gaat worden krijgt men te maken met de persoonlijkheden van medewerkers. Iedereen heeft namelijk zijn eigen manier van werken. Als u in de broncode kijkt van een bepaald project waaraan meerdere mensen hebben gewerkt en dit ziet er uit als een mengelmoesje, dan is het van belang om goede afspraken te maken om dit te kunnen voorkomen.

1.1 Codestyle

Zorg er voor dat iedereen op dezelfde manier programmeert. Iedereen moet bijvoorbeeld dezelfde manier van indentatie gebruiken, dus maak een keuze tussen tabs of spaties, en hoeveel. Ook naamgeving van variabelen en functies is belangrijk, dit gebeurd het liefste via een vast schema zodat je nooit hoeft te raden waar een bepaalde variabele voor staat.

Een voorbeeld kan genomen worden aan de styleguide van Python[1] om een idee te krijgen hoe zoiets er uit zou kunnen zien.

1.2 Framework

Om het samenwerken en de consistentie te verbeteren, kan gebruik gemaakt worden van een framework. Hierin wordt basisfunctionaliteit uit handen genomen zodat niet iedereen bijvoorbeeld zijn eigen functies hoeft te schrijven, er kan gebruik gemaakt worden van de ingebouwde functionaliteit van het framework. Ook nodigt een framework uit om op een bepaalde manier te werken, als iedereen dit volgt zal alles ook een stuk overzichtelijker worden.

Hiernaast baseren de meeste frameworks zich op de MVC structuur, wat staat voor Model, View, Controller. Dit zorgt voor een scheiding tussen de business logica en user interface van een website. Zo wordt het bijvoorbeeld gemakkelijker om het de layout van een site aan te passen zonder daarbij andere gedeeltes aan te tasten.

2 Ontwikkelmodel

De meeste bedrijven maken gebruik van het zogenaamde *waterfall* model, hierbij doorloopt een project verschillende fases. Dit gaat van Conception, Initiation, Analysis, Design, Construction, Testing naar Maintenance. Een vrij lineair proces dus, waarbij elke fase een deliverable kan hebben als bijvoorbeeld ontwerpspecificaties en designstudies. Het waterfall model maakt het moeilijk een project op te delen, er moeten vrij vroeg zaken vastgelegd worden waardoor het moeilijk wordt om in te spelen op veranderingen in eisen. Dit alles maakt het ongeschikt voor een project waar de eisen niet helemaal duidelijk zijn of waarbij deze waarschijnlijk gaan veranderen in de loop van het project.

Een bedrijf kan ook gebruik maken van een *agile* ontwikkelmethode. Deze heeft als doel om elke paar weken compleet afgewerkte en geteste features op te leveren. De nadruk ligt op het zo vroeg mogelijk bereiken van *business value* door te beginnen aan het kleinste functionele deel van een project. Zo kan er dus al snel iets online komen voor een klant. Door snel en continu te leveren maak je je klant blij en heb je ook een goed instrument om de voortgang te meten. Ten slotte kan je snel reageren op wijzigingen doordat je jezelf niet vastlegt op een vooraf gesteld plan.

De keuze voor een plan gedreven project of eentje die gebaseerd is op een Agile methode hangt af van de omstandigheden. Niet elk project of bedrijf leent zich hiervoor maar in kleine dynamische bedrijven kan deze manier van werken van erg geschikt zijn. Barry Boehm en Richard Turner[2] suggereren dat deze keuze gebaseerd moet worden op de volgende omstandigheden:

Agile:

- Low criticality
- Senior developers
- Requirements change often
- Small number of developers
- Culture that drives on chaos

Plan-driven:

- High criticality
- Junior developers

- Requirements don't change too often
- Large number of developers
- Culture that demands order

Referenties

[1] G. van Rossum, http://www.python.org/dev/peps/pep-0008/.

[2] B. W. Boehm and R. Turner, *Balancing Agility and Discipline*, 3 ed. (Addison-Wesley, 2005).