

toolview

Elke ingenieur gebruikt tools en iedereen heeft zijn eigen voorkeur; vaak gebaseerd op wat er toevallig als eerste beschikbaar was. Zo ook de ICT-ingenieur. Ook hij gebruikt tools, software-tools wel te verstaan. Daar zijn er oneindig veel van, waardoor kiezen erg moeilijk is.

Het leren gebruiken van een tool kost tijd. Maar als je de naam en het doel van het tool kent, loont het gebruik al vaak de moeite. Het inleren en inzetten kost vaak minder tijd dan aanmodderen zonder een goed tool.

Veel handige tools worden echter niet gebruikt, eenvoudigweg omdat ze niet bekend zijn. Daarom wenden we deze rubriek aan om degelijke handige tools voor te stellen. Van daar het volgende verzoek:

Langzamerhand begint mijn voorraad tools op te raken! Ken jij 'handige tools' die je met collega's wilt delen? Mail mij, de naam en een URL is al voldoende. Desnoods neem ik contact met je op voor meer informatie. Ik zal het schrijfwerk doen en er een vermelding van maken in deze rubriek.

DOXYGEN

door & more

Doxygen is een opensource tool van Nederlandse komaf. Het is ontwikkeld in Eindhoven, door een student aan de TUE. En hoewel de URL www.doxygen.org bestaat, kom je automatisch uit op de site van de 'computer-hobbyclub' van die universiteit uit.

Er zijn nog veel meer documentatie-generatoren in omloop. In de jaren 90 was het een populair tool om te maken. Sommige ondersteunen slechts 1 taal; denk aan JavaDoc. Het voordeel van Doxygen is dat het heel compleet is doordat steeds meer goede ideeën uit andere tools er in opgenomen zijn. Daardoor gaan steeds minder mensen een nieuw tool maken; ze breiden liever Doxygen uit.

Voor meer info, kijk op:
<http://www.stack.nl/~dimitri/doxygen/>

Elke programmeur is ook schrijver. Althans de echte programmeur, niet de 'klick-and-go' hobbyist. Een echte programmeur schrijft, bij voorkeur pagina's vol code. Het schrijven van documentatie is echter minder populair. Maar dat laatste kunnen we oplossen ...

De code, de code, ...

Programmeurs houden over het algemeen niet van documenteren; ze schrijven liever code. En bij onduidelijkheden kijken ze liever in de code, dan in een document. Dit verschijnsel is al oud; waarschijnlijk even oud als het programmeren zelf. Ik ken geen onderzoek naar dit verschijnsel. Het ligt niet aan het schrijven zelf lijkt mij, want dat doen programmeurs frequent. Noch aan een afkeer van boeken, want die lezen ze veel. Wellicht komt het doordat er een andere filesoort of editor gebruikt moet worden. Maar dan is er goed nieuws: Doxygen! Dit is een tool om documentatie te genereren, uit sourcefiles!

En de code

Doxygen gaat uit van de code. Het kan de meest gebruikte talen, zoals C, C++, C# en zelfs Objective-C en IDL lezen en analyseren. Hierdoor worden alle functies, objecten en andere structuren dus automatisch opgenomen in de documentatie. Natuurlijk is er meer nodig voor goede documentatie, namelijk tekst. Beschrijvende tekst die uitlegt wat functies doen en waarom keuzes gemaakt zijn. Als de programmeur die teksten plaats in commentaar, dan zal Doxygen die gebruiken. Waarbij een 'tag' gebruikt wordt om het soort beschrijving aan te geven. Het resultaat kan een goed leesbaar boek worden (kies als formaat: MS-Word of postscript) en/of een stel handige hypertext-pagina's (keuze uit html of hyper-pdf). Maar ook andere formaten, zoals LaTeX of manual/help-pagina's zijn mogelijk.

Make doc

Het excuus 'de documentatie is verouderd' om in de code te kijken, is bij gebruik van Doxygen niet meer geldig. Er wordt immers uitgegaan van dezelfde bron(code). Hierdoor is het heel eenvoudig om de documentatie actueel te houden: Het genereren van de documentatie is te automatiseren, bijvoorbeeld in de Makefile. Samen met versiebeheer, maakt dit principe het zelfs mogelijk om correcte documentatie te gebruiken voor elke release.

Toch schrijven

Om Doxygen te gebruiken, moet je een uitgebreide configuratiefile schrijven. Dat kan afschrikwekkend werken, maar met een beetje google'n valt het reuze mee. En als één programmeur er mee start, kunnen de anderen volgen. Het echte werk is toch het documenteren zelf. Want al gebruiken we die fijne editor, al schrijven we (slechts) codefiles en genereren we de documentatie, het schrijven van zinvol commentaar blijft schrijfwerk. Maar goed, we zijn 'schrijvers'. Althans, de 'tikers' onder ons.