

Project Software-ontwikkeling en Professionele Vaardigheden

Opgave

Universiteit Hasselt

Academiejaar 2014-2015

1. Doelstellingen

1.1 Studiegids

De meeste informaticaprojecten in de praktijk zijn veel te groot of moeten op te korte termijn opgeleverd worden om door een enkele persoon te laten uitvoeren. Teamwerk, met duidelijke afspraken (bijvoorbeeld interfaces) tussen teamleden, is noodzakelijk om een grote taak binnen gestelde termijn tot een goed einde te brengen. Meestal zijn de opdrachtgevers zelf geen informatici en zijn ze niet vertrouwd hoe problemen ondubbelzinnig gespecificeerd worden in de informatica. De informaticus zal derhalve op basis van gesprek met de opdrachtgever een analyse moeten uitvoeren van het probleem, waarop ontwerp en planning volgt. Vermits informatica-ontwikkeling een dure zaak is, en vaak opvolging achteraf nodig is (bijvoorbeeld nieuwe mogelijkheden), is rapportering en documentatie van groot belang.

In dit project zullen naast rapportering en documentering ook verscheidene andere professionele vaardigheden ingeoeft worden zoals presenteren, teamwork en time management. Bovendien worden real-life opdrachten gegeven (waar praktisch mogelijk ook met externe opdrachtgever), zodat dit project ook bijdraagt tot de stakeholder awareness van de deelnemende studenten.

Met dit project leert de student dat informatica in de praktijk meer is dan het toepassen van de juiste algoritmen en datastructuren en het schrijven van goede code. De student zal voor een eerste keer ook met alle bovenstaande aspecten in aanraking komen, weliswaar in een gecontroleerde omgeving.

1.2 Uitwerking doelstellingen

De doelstellingen uit de studiegids worden als volgt in praktijk omgezet:

Werken in teamverband

Het project wordt gemaakt in groepjes van twee personen. De samenstelling van de groepjes wordt door de studenten onderling bepaald, alsook het onderwerp (te kiezen uit een lijst van voorstellen). Indien het studentenaantal oneven is, kan hiervan afgeweken worden.

Maken van een analyse

Na het samenstellen van de groepjes, het kiezen van een onderwerp, en een eerste afspraak met de begeleiders en opdrachtgevers, wordt het eerste trimester van het project besteed aan het maken van een analyse. Het doel van het analyseverslag is om vóór er een regel broncode geschreven wordt, een goed beeld te vormen over de belangrijkste/meest complexe onderdelen van het project, waaronder datastructuren en algoritmen, OO-ontwerp, een gedetailleerde planning en een taakverdeling. Bovendien krijgen de begeleiders en opdrachtgever zo al vroeg een duidelijk beeld van de geplande werkwijze, zodat er indien nodig tijdig kan bijgestuurd worden.

Implementatie

De gekozen opdracht wordt geïmplementeerd in een object-georiënteerde programmeertaal naar keuze, tenzij de opdracht anders vereist. Bij de beoordeling van de gemaakte applicatie komen de kwaliteit van de interne structuur en de achterliggende algoritmes op de eerste plaats.

Schrijven van verslagen en handleiding

De uiteindelijke resultaten worden ook schriftelijk gerapporteerd in een eindverslag waarin de interne structuur/werking en een bondige handleiding staan.

Het resultaat wordt ten slotte in een openbare projectvoorstelling verdedigd.

2. Planning

De belangrijkste data in verband met het project zijn weergegeven in onderstaande tabel. Gedurende de looptijd van het vak kunnen wijzigingen doorgevoerd worden, maar deze zullen steeds tijdig worden aangekondigd.

Deadlines	Studenten	Begeleiding
09/01 17:00	<ul style="list-style-type: none">• samenstelling van de projectgroepen• indienen top-drie onderwerpen• schrijven van motivatie voor alle 3 de keuzes	<ul style="list-style-type: none">• richtlijnen structuur analyseverslag• toekenning onderwerpen: 14/1
15/01 - 28/01	<ul style="list-style-type: none">• afspraken met de opdrachtgevers en begeleiders	
01/03 - 23:59	<ul style="list-style-type: none">• inleveren analyseverslag	<ul style="list-style-type: none">• richtlijnen broncode/eindverslag
26/02	<ul style="list-style-type: none">• HC over presentatie	
05/03	<ul style="list-style-type: none">• presentatie analyse	<ul style="list-style-type: none">• richtlijnen presentatie
30/04	<ul style="list-style-type: none">• tussentijdse rapportering met opdrachtgevers	
04/06 - 23:59	<ul style="list-style-type: none">• inleveren eindverslag en project	
09/06	<ul style="list-style-type: none">• eindpresentatie	<ul style="list-style-type: none">• richtlijnen presentatie

3. Puntenverdeling

Er wordt een lijst met onderwerpen van verschillende moeilijkheidsgraad opgegeven vanuit de informaticastaf. De score houdt uiteraard ook rekening met de complexiteit van de opgave en de kwaliteit van de uitwerking. Elk project kan ook opgewaardeerd worden door een aantal extra's.

Het eindresultaat wordt bepaald op basis van:

- Ontwerp/analyse: kwaliteit van het onderzoek/analysewerk.
- Implementatie: kwaliteit van de broncode en applicatie. Zie ook de richtlijnen voor code en OO ontwerp uit de vakken OGP I en OGP II.
- Verslag: inhoud en vorm van analyse- en eindverslag.
- Presentatie: inhoud en vorm van de presentatie.

4. Extra info

4.1 Begeleiding

De coördinerend verantwoordelijke van dit vak is Wim Lamotte.
De co-titularis is Frank Van Reeth.

De begeleiders zijn:

- Bram Bonné: bram.bonne@uhasselt.be
- Jonny Daenen: jonny.daenen@uhasselt.be
- Steven Maesen: steven.maesen@uhasselt.be
- Kris Gabriëls: kris.gabriels@uhasselt.be
- Jelle Hellings: jelle.hellings@uhasselt.be
- Arno Barzan: arno.barzan@uhasselt.be
- Eva Geurts: eva.geurts@uhasselt.be

4.2 Keuze onderwerp

Gebruik BlackBoard om je groep samen te stellen en drie onderwerpen te kiezen.

Voorzie ook een eigen interpretatie en een ruw plan van aanpak voor elk onderwerp. Hierin geef je je eigen kijk op het onderwerp weer, en verduidelijk je hoe je het project wil aanpakken. Op die manier kan je de opdrachtgever overtuigen om jullie groepje te selecteren om dit onderwerp te realiseren. Gebruik de eerste dag om het nodige opzoekingswerk te verrichten en te weten te komen wat er zoal bestaat met betrekking tot het onderwerp. Voorzie voor elk onderwerp ongeveer 1 bladzijde tekst. Vergeet je bronvermeldingen niet.

Het onderwijsteam zal daarna een lijst publiceren waarin elke groep een onderwerp en begeleider toegewezen krijgt, rekening houdend met de kwaliteit van de ingediende voorstellen.

4.3 Communicatie

Tijdens de dagen waarop het project geprogrammeerd staat kan een afspraak gemaakt worden met de begeleiders. Eventuele vragen kunnen dan gesteld worden. Ook op andere momenten kan je vragen stellen via mail aan de begeleider van je onderwerp en bijkomende afspraken maken met je begeleider. Alle mededelingen in verband met het project gebeuren door middel van e-mail en BlackBoard. Kijk beide dus regelmatig na.

Overleg met de opdrachtgevers zal voornamelijk gebeuren tijdens de overlegperiode bij het begin van het project, de presentatie van de analyse, de tussentijdse rapportering, en de eindpresentatie. Indien er onduidelijkheden of vragen zijn, kan er echter steeds via e-mail gecommuniceerd worden met de begeleider. Indien er met de opdrachtgever contact opgenomen wordt, dient dit steeds te gebeuren in samenspraak met de begeleider.

4.4 Source Control Management

Het is verplicht de SCM server te gebruiken gedurende het volledige project. Standaard wordt er een SVN repository voorzien, een git/mercurial repository is te verkrijgen op aanvraag. We verwachten op regelmatige basis 'commits', waarbij het belangrijk is dat iedereen zijn of haar bijdragen commit via de eigen account. Zo kunnen we niet enkel de vooruitgang van de groep opvolgen, maar ook de bijdragen van iedere individuele student. Dit kan van belang zijn bij een eventuele onevenredige participatie in de groep.

Hieronder enkele richtlijnen over het gebruik van het versiecontrolesysteem voor het project:

- Commit alleen wanneer je code compileert en werkt! Jullie projectpartner en het onderwijsteam moeten in staat zijn om eender welke willekeurige commit te downloaden en deze zonder problemen te compileren en runnen.
- Beperk de hoeveelheid aangepaste code per commit tot één logische 'change set'. Dat wil zeggen dat je alles wat in een commit gewijzigd is moet kunnen samenvatten in één zin. Indien dit niet lukt kan je waarschijnlijk beter de verschillende stukken van je code apart committen.
- Commit vaak. Elke dag dat je aan je project gewerkt hebt moet er ten minste één commit gebeurd zijn. Dit is het absolute minimum: normaal gezien zal je, gebaseerd op het vorige puntje, voor elke paar regels code een commit uitvoeren.

- Zorg altijd voor een duidelijke commit message (ook bij een eenvoudige bugfix: welke bug lost je commit op?).
- Hou je code continu up to date met de code van je projectpartner. Zorg er tenminste voor dat je de laatste versie van de code van de server haalt voor je begint met programmeren, en update deze zo vaak mogelijk om merge conflicten te vermijden.

4.5 Verslag

De verslagen worden ingeleverd via BlackBoard. Zowel inhoud als vorm (structurering en presentatie van de tekst) zijn belangrijk: hierover krijgen jullie nog aparte instructies. Op beide aspecten wordt er feedback gegeven. In beide gevallen geldt dat de gebruikte taal in het verslag duidelijk en terzake moet zijn.

4.6 Eindresultaat

De finale versie van de applicatie dient *gezip*t te worden alvorens het in te leveren via BlackBoard. *Lever geen tijdelijke bestanden, gecompileerde objecten en dergelijke mee in.* Overleg met je begeleider over welke datasets/libraries/externe bestanden/... mee ingestuurd moeten worden. De juiste folder lay-out wordt nog meegedeeld in een apart documentje.

Tenslotte voorzien we op BlackBoard een "project gallery wiki" waarin elk project een pagina krijgt. De projectgroepen zijn zelf verantwoordelijk voor de invulling van deze pagina. Voorzie hierop minstens een korte beschrijving en een aantal afbeeldingen van de applicatie in werking. Zorg ervoor dat deze representatief zijn en dat ze elk aspect van de applicatie goed in de verf zetten.

Let erop dat deadlines **zeer strikt** worden geïnterpreteerd!