

Projecthandleiding 'Containing' (project 2a)



NHL Hogeschool Leeuwarden
Afdeling Engineering
Opleiding Informatica
2e jaar

Wouter van der Ploeg
David Schweizer
2015

Inhoud

Inleiding.....	3
Inleiding project.....	3
Doel van het project.....	3
Indeling teamleden.....	3
Begeleiders (tutoren).....	4
Experts.....	4
Begin van het project.....	4
Technieken die in het Project worden toegepast.....	4
Tijdschema.....	5
Producten.....	6
Faciliteiten.....	6
Extra Informatie.....	6
Assessment.....	6
Toetsing en beoordeling.....	7
Tijdpad.....	8
Bijlage 1.....	9
Algemene Eisen en criteria:.....	9
Bijlage 2.....	10
Beoordelingscriteria voor UML:.....	10
Bijlage 3.....	11
Testen/testverslag.....	11

Inleiding

In deze handleiding wordt een aantal zaken rondom het eerste 2^e jaars project beschreven. Het project loopt in de 2^e periode van het studiejaar 2015/2016. In dit project komen de vakken die gegeven worden in de eerste periode van het 2e jaar samen.

Inleiding project

Project 2a begint met een opdracht van een opdrachtgever met een reëel probleem. Het probleem of uitdaging (net wat je wilt) ligt op de vlak van het simuleren/animeren van logistieke problemen met de computer. **De opdracht wordt gegeven in het bijgevoegde document 'Opdrachtdocument Containing 2015'.**

In het begin zal je niet direct alles duidelijk zijn, de opdrachtgever spreekt een ander jargon dan jij. (In bedrijfsprojecten is dat meer regel dan uitzondering)

Voor onduidelijkheden en lastige zaken hebben we dan ook experts per vakgebied in het leven geroepen, die tijdens het project benaderbaar zijn.

Kortom het is een project dat uit het leven gegrepen is, waarbij niet alles op voorhand duidelijk is.

Voor alle duidelijkheid: Wat je moet doen staat beschreven in deze projecthandleiding. Overkoepelende eis is dat je de opdracht zoals in het opdrachtdocument beschreven staat zo goed mogelijk uitvoert. Waar er onduidelijkheden in dat document zijn is het aan jou om daar opheldering voor te krijgen.

Doel van het project

Programmeerervaring opdoen in een realistisch project.
Dat wil zeggen:

- in teamverband kunnen programmeren.
- verantwoordelijkheid nemen en dragen voor het afleveren van een werkend en overdraagbaar product. (inclusief testen en documentatie)
- verantwoordelijkheid nemen en dragen voor het organiseren (rolverdeling, mijlpalen, planning, testen etc.) van het project.

Indeling teamleden

De teams bestaan uit 5 of 6 studenten. Dat betekent dat je goede afspraken moet maken over de taakverdeling en samenwerking, maar ook dat het mogelijk is om een groot project tot een goed einde te brengen. De groepen worden samengesteld door de begeleidende docenten. Per groep is er 1 docent als tutor (begeleider) beschikbaar. Deze tutor hoeft niet persé inhoudelijk deskundig te zijn.

Begeleiders (tutoren)

Groep	Tutor	email tutor
1	Aswin van Woudenberg	aswin.van.woudenberg@nhl.nl
2	David Schweizer	david.schweizer@nhl.nl
3	Fiona Sariedine	f.sariedine@nhl.nl
4	Wouter van der Ploeg	w.van.der.ploeg@nhl.nl
5	Albert Sikkema	sikkema@nhl.nl
6	Joris Lops	joris.lops@nhl.nl
7	Jop Wielens	Jop.Wielens@nhl.nl
8	Nynke van der Wijk	nynke.van.der.Wijk@nhl.nl
9	Jos Foppele	j.foppele@nhl.nl
10	Greta Schaafsma	g.schaafsma@nhl.nl
11	Jan van der Wal	j.van.der.wal@nhl.nl

Experts

De Experts zijn op afroep beschikbaar, dit houdt niet in dat de experts direct naar je toekomen (bij elk probleempje).

De bedoeling is dat het team de vragen/problemen eerst verzamelt, dit onder woorden brengt en voorlegt aan de experts. Die zullen daar dan op korte termijn op reageren.

Vakgebied	Expert	Email
OOP	Dick Bruin	dick.bruin@nhl.nl
	Albert Sikkema	sikkema@nhl.nl
Graphics	Dick Bruin	dick.bruin@nhl.nl
UML	Wouter van der Ploeg	ploegw@nhl.nl
Testen	Fiona Sariedine	f.sariedine@nhl.nl
Wiskunde	Jos Foppele	foppele@nhl.nl
Scrum	David Schweizer	david.schweizer@nhl.nl
Projectmanagement	tutor	zie lijst tutoren

Begin van het project

Na de start moet de opdracht duidelijk zijn voor het projectteam. Daarna moet in het begin van de 2^e periode een realistisch Testplan, (scrum-)Product-Backlog en 1^e sprint-backlog gemaakt worden.

Technieken die in het Project worden toegepast.

In het project moeten de volgende technieken worden toegepast:

- OO programmeren in Java en C++ (OOP)
- Ontwerpen met UML
- Grafische animatie met behulp van JMonkey in 3D.
- Testen volgens specificatie
- Wiskundige bewerkingen (bv Dijkstra-algoritme)
- Scrum, proces, samenwerking (projectmanagement)
- XML

Tijdschema

Er is **geen** uitloop mogelijk, vóór de extra week van periode 2 moet het af zijn!

In de eerste week van periode 2 ben je verplicht een goed tijdplan op te stellen (je begeleider gaat dat controleren). Daarnaast ben je verplicht ten minste één keer per week met je begeleider te vergaderen over de voortgang. Verder staat het je vrij om met de groep te bepalen hoe je er voor gaat zorgen dat het eindproduct aan de eisen van de opdrachtgever voldoet. Maar houdt wel rekening met de mijlpalen die in het Tijdspad staan vermeld.

Producten

- Een kort en bondig eindrapport met daarin:
 - o een procesverslag
 - o verantwoording van de gemaakte keuzes
 - o een geactualiseerde UML analyse
 - o de gebruikte wiskunde
 - o jullie evaluatie van het project
- executables / jar en de source-code met documentatie.
- Een mondelinge presentatie ***in het Engels*** en een demonstratie voor de opdrachtgever en de andere teams bij de afsluiting.
- Testverslag
Er zijn 3 testen verplicht:
 - 1) Het testen van de kortste route. De test moet ook aantonen dat het programma fool-proof is
 - 2) Het testen op XML. De test moet ook aantonen dat het programma op foutieve, of corrupte XML-files niet stuk loopt
 - 3) Het testen van de verbinding tussen de verschillende programma's. De test moet ook aantonen dat ook bij een haperende, wegvallende, storende verbinding het programma niet stuk loopt.

Deze formuleringen zijn nog vrij algemeen. Geef in je testverslag duidelijk aan hoe je de drie bovenstaande testen concreet hebt uitgevoerd.

Zie bijlage 3 voor een template.

Faciliteiten

De werkzaamheden worden zoveel mogelijk in de studielandschappen uitgevoerd. Voor de afronding met presentatie gebruiken we een zaal met PC en beamer.

Extra Informatie

Extra informatie over het project wordt via Blackboard <https://elo.nhl.nl> beschikbaar gemaakt, onder de naam 'Project Containing TECH_E_05_1040'.

Assessment

In week 4 van de periode moet je een individueel assessment doen. Daarin toetsen we of jouw individuele bijdrage aan het programmeren en de andere werkzaamheden voldoende is. We vragen op het assessment uitleg van de code die jezelf gemaakt hebt en naar ten minste één ander onderdeel. ***Dit assessment moet voldoende zijn om verder aan het project deel te mogen nemen (en dus om studiepunten te halen)!***

Toetsing en beoordeling

De beoordelingscriteria zijn als volgt gegroepeerd in een aantal categorieën.

	Te behalen	Min. ¹⁾	Toegekend
Objectgeöriënteerd Programmeren (oordeel expert) <ul style="list-style-type: none"> • Codering • Documentatie • Testen 	20	11	
Analyse (oordeel expert) <ul style="list-style-type: none"> • Klassendiagrammen (controller + simulator) – klassen/attributen/methodes/toelichting/associaties • Sequence diagram • Use-Casediagrammen • State Transition Diagram • samenhang KD, SD's, UCD en STD's 	20	11	
Uitslag tussentijds-assessment (oordeel expert)	20	11	
Projectmanagement + Groep (oordeel tutor) <ul style="list-style-type: none"> • Product backlog • Sprint backlogs • Sprint documentatie • Eindrapport • Onderling samenwerken • Daily scrum • Presentatie • Evaluatie leerdoelen 	15	8	
Wiskunde (oordeel expert)	15	8	
Tevredenheid Opdrachtgever (expert) <ul style="list-style-type: none"> • Demonstratie eindproduct • Help-functie 	10	7	
Totaal aantal punten	100	56	

¹⁾ Om de studiepunten voor het project te verkrijgen moet **voor elke categorie minimaal het minimum aantal** punten behaald zijn.

Tijdpad**Periode 2**

datum	week	dag tijd	wie	wat
	-2		Stud – individueel	<i>Beginnen met Analyse (SON)</i>
28-10	-2	Wo 8.30u	Studenten+ beg	<i>Aftrap</i>
2-11	0			<i>Tentamenweek</i>
9-11	1	Ma 13.00u Di-Wo	Studenten+ beg. Stud – individueel	centrale bijeenkomst <i>Assessment SON volgens rooster</i>
16-11	2	Ma 13.00u	Studenten+ beg.	centrale bijeenkomst
		Ma 17.00u	Studenten	inleveren gezamenlijke analyse, planning.
23-11	3	Ma 13.00u	Studenten+ beg.	centrale bijeenkomst
30-11	4	Ma 13.00u	Studenten+ beg.	centrale bijeenkomst
		nnb	Studenten+ beg.	TestSessie 1 <i>a.d.h.v. tijdig ingeleverd testverslag¹</i>
7-12	5	Ma 13.00u	Studenten+ beg.	centrale bijeenkomst
		nnb	Stud – individueel	assessments project <i>volgens rooster</i>
14-12	6	Ma	Studenten+ beg.	centrale bijeenkomst
		nnb	Studenten+ beg.	TestSessie 2 <i>a.d.h.v. tijdig ingeleverd testverslag</i>
21-12	7/8			<i>Kerstvakantie</i>
4-1	9			
11-1	10	Do	Studenten+ beg.	Afsluiting: Presentaties in het Engels. Demonstraties + Inleveren producten

 1

Eén dag vóór de testsessie moet het testverslag ingeleverd zijn.

Bijlage 1

Algemene Eisen en criteria:

- Het programma voldoet zo goed mogelijk aan het opdrachtdocument.
- Het programma moet in Java en C++ geprogrammeerd worden.
- Commentaar per class/ goede naamgeving/goed inspringen, etc.
- Juist gebruik constanten (final of static final)
- Vang alle Exceptions netjes af
- Verdeel zo mogelijk in logische packages
- Gebruik van JMonkey om 3D te realiseren.
- Testen volgens specificatie
- Documentatie van de eigen gemaakte classes.
(Zodanig dat een volgende groep verder kan met de gemaakte classes)
- De gemaakte classes zijn in overstemming met de gedocumenteerde UML.
- Gemaakte Testprogramma's/Testdata moeten gedocumenteerd en bij het project blijven.
- Een gebruikersvriendelijke installatie van de eigen applicatie
- Bedieningsgemak: het werken met het programma moet voldoende gemakkelijk zijn voor de gebruiker.

Bijlage 2 Beoordelingscriteria UML (zie ook het SON-dictaat: “Het controleren van een UML-analyse”)

<i>syntaxcriteria UCD</i>	<i>inhoudelijke criteria UCD</i>
a) is input terug te vinden?	a) uitleggen?
b) actieve actor bij elke use-case?	b) #UC ≤ 7?
c) alle user-stories?	c) story-koppeling?
d) use-case goed geformuleerd?	d) include / extend ?
e) actoren buiten systeem?	e) opsplitsing?
f) is output terug te vinden?	f) interactie actors?

<i>syntaxcriteria KD</i>	<i>inhoudelijke criteria KD</i>
a) actor → klasse?	a) logische eenheden?
b) actieve klassen?	b) te groot?
c) geen ongedekte use-cases?	c) data-klassen nodig?
d) story → methode?	d) database?
e) ongedekte acties?	e) in welke klasse gegevens?
f) associatie → aanroep?	f) uitleg collega?
g) aanroep → associatie?	g) nuttige generalisatie?
h) naam associatie?	h) weerspiegeling probleem?
i) multipliciteiten?	i) goede code?

<i>syntaxcriteria SD</i>
a) begint SD met actor?
b) acties use-case gedekt door methoden?
c) komen alle objecten uit het KD?
d) beschikt object over benodigde data?
e) zijn alle actoren aanwezig in SD?
f) komen scenario's terug in SD?
g) leesbaar op A4?

<i>syntaxcriteria STD</i>	<i>inhoudelijke criteria STD</i>
a) is er een begintoestand?	a) werkelijk bestaande toestanden?
b) namen toestanden goed geformuleerd?	b) toestanden complementair?
c) geen zwarte gaten?	c) object of real-time STD?
d) geen supernova's?	d) nieuwe methoden?
e) gebeurtenis of conditie per transitie?	e) elke methode toegewezen aan klasse?
f) kun je uitgaande transities onderscheiden?	f) leveren condities nieuwe attributen op?
g) testbare condities?	
h) zijn de triggers werkelijke gebeurtenissen?	
i) dekt het SD alle gebeurtenissen?	

Bijlage 3

Testen/testverslag

De volgende onderwerpen moeten minimaal in het testverslag terugkomen:

- Naam groep/leden
- Datum / tijd van het testwerk en het testverslag
- Hoe hebben jullie de testen uitgevoerd?
- Welke testmethodes/technieken hebben jullie toegepast?
- Voor het testverslag de volgende tabel met testpunten opnemen:

<u>Punten</u>	<u>Status</u> (Open/Close)	<u>Omschrijving</u> (bij Open probleem) (bij Close eventueel opmerking)	<u>Datum</u> (datum van Close of wanneer dit punt in de planning staat)	<u>Prioriteit</u> (hoog, middel, laag)