

# AGILE WEB APPLICATION PROGRAMMING

Data & Informatie (201300180)

Projecthandleiding 2014

**Djoerd Hiemstra**

(Versie 0.9 – 18 februari 2014)

(Versie 1.0 – 11 april 2014)

Versie 1.2 – 22 april 2014

## Voorwoord

Veel van de meest succesvolle nieuwe bedrijven van de afgelopen 15 jaren, zoals Google, Facebook, Spotify, Twitter, begonnen met een geweldig idee, kennis van webtechnologie, en een klein, behendig (agile) team van gemotiveerde softwareontwikkelaars, vaak maar enkele personen. In de module Data & Informatie leer je een webapplicatie voor op de desktop of mobiele telefoon (of beide) te ontwerpen en implementeren in een team van 5 personen, met de nadruk op requirements, webtechnologie, databasetechnologie, semi-gestructureerde data, en het samenwerken door middel van agile softwareontwikkeling.

*Maak een epische web app!*

Enschede, april 2014

Djoerd Hiemstra, Luís Ferreira Pires, Klaas Sikkel, Maurice van Keulen

## Inhoudsopgave

1 Inleiding.....	3
2 Randvoorwaarden en beoordeling.....	4
3 Werkwijze en gereedschappen.....	5
4 Planning en deliverables.....	6
5 Science Challenges.....	10
6 Conclusie.....	12

## 1 Inleiding

In de module Data & Informatie leer en oefen je het ontwerpen, implementeren en testen van complexe softwaresystemen, die bestaan uit meerdere lagen, gebruik makend van standaard-gereedschappen zoals databasemanagementsystemen, CSS- en scriptingraamwerken, en web services. Hierbij spelen data een belangrijke rol: zowel klassiek gestructureerde gegevens in databases, alsook semi-gestructureerde data en het verkrijgen van gegevens uit sociale media en andere web resources.

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

*Manifesto for Agile Software Development (<http://agilemanifesto.org>)*

In de module wordt een groot project uitgevoerd volgens “agile” methodieken. Dat betekent onder andere dat ervaring wordt opgedaan in de omgang met stakeholders (de “klant”), dat teams zelfsturend zijn (er is geen manager die zegt wat je moet doen), dat teams multi-disciplinair zijn, en dat er elke 2 weken een nieuw of verbeterd product wordt opgeleverd (er wordt gewerkt in korte sprints van 2 weken). De scrum werkwijze wordt uitgelegd in het tweede college op dinsdag 10.45 uur. Je zult onder andere de volgende begrippen moeten kennen en kunnen hanteren: *product backlog, sprint, sprint backlog, user stories, sprint planning meeting, sprint review meeting, sprint retrospective, task board, daily standup, scrum master, development team, product owner, burn down chart*. Zie: [http://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development)) en paragrafen 4.2, 4.3, 4.9, 4.10, 4.13, 4.19, en 4.20 van deze handleiding. IBM biedt een gratis inleiding in Agile aan via: <http://ibmsoftwaredelivery.sourceforge.net/agile-for-dummies-an-ibm-limited-edition-e-book/>

Groepen kunnen een eigen case kiezen (zie paragraaf 1.1), of als ze ambitieus zijn, een van de *echte* cases (zie paragraaf 1.2). Groepen moeten de keuze van hun case **uiterlijk 25 april** overleggen aan een van de scrummentoren. Een van je scrummentoren (de docent) zal optreden als *product owner* en spreken namens de klant. Als je een echte case doet dan is er beperkt de mogelijkheid om (via de product owner) de opdrachtgever om meer informatie te vragen.

### 1.1 Een eigen case

Groepen zijn vrij om webapplicatie te maken voor een eigen case. Maak een webapplicatie voor je flat, je sportvereniging, de theatersportgroep, het bedrijf van je oom, de kerk, of welke andere organisatie dan ook. De webapplicatie moet aan een aantal technische randvoorwaarden voldoen, zie hoofdstuk 2 en hoofdstuk 4.

### 1.2 Een echte case

Doe mee aan Science Challenges en kies een van de cases die aangeleverd zijn door echte klanten. Het meedoen aan Science Challenges betekent dat je je inspant voor een echte klant, dat je extra hulp krijgt en meer kans hebt samen te werken (onder andere woensdagmiddag om 15.45 uur), en dat je een bruikbaar product aflevert voor de klant. De groep die het beste product oplevert ontvangt een prijs. Het meedoen aan Science Challenges brengt geen extra risico mee dat je de module niet haalt, integendeel, het verhoogt de kans om goed te scoren voor het project. Zie hoofdstuk 5.

## 2 Randvoorwaarden en beoordeling

Het project wordt beoordeeld aan de hand van de volgende 24 randvoorwaarden. Deze randvoorwaarden worden toegelicht in hoofdstuk 4.

I never teach my pupils. I only attempt to provide the conditions in which they can learn.

*Albert Einstein (<http://www.einsteinssecret.net/about-the-secret/>)*

1. Correct gebruik van git en git branching.
2. Product backlog en sprint backlog op Trello (user stories + tasks).
3. Scrum 'daily' standup meetings op iedere woensdag-, donderdag- en vrijdagochtend.
4. Er is een case/project gekozen.
5. Er is een correct UML use case diagram voor het systeem.
6. Er is een correct UML klassendiagram voor het systeem.
7. Er is een correct SQL Database ontwerp.
8. Er is een mockup van de applicatie met behulp van een CSS framework.
9. Er zijn 5 presentaties/demonstraties gedaan (de sprint review meetings).
10. Er zijn 3 sprint evaluatie- en planning sessies gedaan.
11. Code compileert met behulp van Maven naar een Java Web Archive.
12. De code wordt getest met unit tests.
13. Het team gebruikt test-driven development .
14. Code maakt correct gebruik van Java servlets.
15. Code maakt gebruik van RESTful services + XML of JSON.
16. Code gebruikt JDBC.
17. Code gebruikt een framework voor HTML, CSS en Javascript.
18. Het systeem heeft login faciliteiten.
19. Team work: iedereen draagt bij aan Git, Trello, presentaties, etc.
20. Werkend systeem: het systeem draait op de centrale Tomcat server.
21. Een rapport is opgesteld over Ethische aspecten en richtlijnen.
22. Analyse van de security.
23. Concurrency: correct gebruik van database transacties.
24. Bonus (optioneel): Doe mee met Science Challenges, zie hoofdstuk 5

### 3 Werkwijze en gereedschappen

Bij grote softwareprojecten is het onmogelijk (of in elk geval te duur) om alles zelf te bouwen. Je zult gebruik maken van bestaande softwarebibliotheken, bestaande softwaregereedschappen, bestaande softwareraamwerken en bestaande applicaties. De keuze van de werkwijze en gereedschappen hangt af van de omgeving waarin je werkt, en is soms arbitrair. Het agile manifeste zegt dat individuen en interacties boven werkwijze en gereedschappen gaan. Veel werkwijzen en gereedschappen leiden tot goede software en de precieze keuze voor een werkwijze is onbelangrijk. Echter, samenwerken in een team is alleen mogelijk als de teamleden afspreken om *dezelfde* werkwijze te volgen en (grotendeels) dezelfde gereedschappen te gebruiken. Iedereen in het team gebruikt deze gereedschappen, zelfs als je persoonlijke voorkeur ergens anders ligt, zoals mooi verwoord door Guido van Rossum, de geestelijke vader van Python.

You can make me use Eclipse but you can't make me like it.

Guido van Rossum (<https://twitter.com/gvanrossum/status/424595747397332992>)

Succesvolle grote softwarebedrijven dwingen een standaard werkwijze vaak af voor een *groep* van projecten, zodat medewerkers gemakkelijk kunnen inwerken in een nieuw project. Elk project ziet er tot op zekere hoogte gelijk uit: broncode staat op een vaste plek, unit tests staan op een vaste plek; het compileren van software volgt vaste fases (compile, test, package, verify, install), documentatie ziet er hetzelfde uit, etc. Zonder een standaard werkwijze en vaste gereedschappen kost het meer tijd om je in te werken in nieuwe projecten, en niet onbelangrijk, zonder standaard werkwijze kost het de docenten van Data & Informatie meer tijd om de projectresultaten te beoordelen. Ieder project *moet* gebruik maken van de volgende gereedschappen:

- Trello (voor het plannen van werk)
- VisualParadigm (voor het maken van ontwerpmodellen)
- Eclipse (voor softwareontwikkeling)
- Git en github (voor het samenwerken aan code)
- Maven (voor het compileren, testen en bouwen van de webapplicatie)
- Java en Java servlets / JSP (voor het maken van webapplicaties)
- Jersey (voor RESTful webservices)
- JUnit (voor unit tests)
- PostgreSQL (als database management system)
- Apache Tomcat (web server)

My goals were different and I wanted to make an *opinionated* piece of software and I preferred the notion of convention over configuration. I wanted a project's infrastructure to look the same and work the same so I continued to pursue my own model for a project (...) I wanted to save people time by being able to find things in the same place.

Jason van Zyl (<http://maven.apache.org/background/history-of-maven.html>)

Sommige gereedschappen, zoals Maven, komen met een rigide standaard werkwijze waarvan je alleen moet afwijken als er *zwaarwegende* redenen zijn om je eigen werkwijze te volgen. Dit wordt vaak aangeduid met “convention over configuration”. Veel softwareraamwerken dwingen ontwikkelaars in zo'n rigide werkwijze. Onder een raamwerk (Engels: frameworks) verstaan we een *herbruikbaar softwarecomponent*. Hoe breder je een raamwerk wilt toepassen, hoe meer configuratie nodig is, en desto minder tijdwinst en nut het raamwerk heeft bij het softwareproject. We raden groepen aan om aanvullende afspraken te maken over werkwijzes en gereedschappen, bijvoorbeeld over het gebruik van een raamwerk voor HTML5, CSS, en JavaScript.

## 4 Planning en deliverables

In dit hoofdstuk worden de randvoorwaarden uit hoofdstuk 2 verder toegelicht en expliciet gemaakt. Voor het realiseren van de eerste paar randvoorwaarden (in de eerste sprint) liggen de deadlines vast. Daarna krijgt het team meer vrijheid om in eigen tempo te werken.

I love deadlines. I like the whooshing sound they make as they fly by.

*Douglas Adams*

### 4.1 Git en github

Zorg dat je lid bent van een team van 5 leden (teamindeling vindt plaats op 22 april 2014, zie de informatie op Blackboard). Ieder lid zorgt dat het een gratis account heeft op github.com. Vraag een van je scrum mentoren naar je groepnummer, en vraag om de accounts van elk teamlid toe te voegen aan je Github team (bijvoorbeeld: als je groep 1 bent, dan is je teamnaam “di01”, zie: <http://github.com/utwente-di/>). Voeg een file **README.md** toe.

*Belangrijk:* Zorg dat de HEAD altijd compileert naar een werkend product; zorg dat je branches maakt voor nieuwe features; zorg dat unit tests klaar zijn (en gecommitt) voordat de nieuwe functionaliteit geïmplementeerd wordt.

*Tip:* Zie <http://git-scm.com/book/en/Getting-Started>

*Deadline:* 25 april, iedereen lid van github.

### 4.2 Sprint backlog op Trello / planning meeting

Ieder lid zorgt dat het een gratis account heeft op trello.com. Maak een project aan op Trello, voeg taken toe als 'user stories' en maak een sprint backlog. Zorg dat elke taak een geschatte tijdsbesteding heeft. Zorg dat je beide scrum mentoren ook lid zijn van het Trelloproject. Voeg de URL van het Trello bord (bijvoorbeeld <https://trello.com/b/0tcjABX8/foobarretje> als je team 'foobarretje' heet) toe aan de file **README.md** in de github repository van het team.

*Tip 1:* Houdt planning meetings op 24 april, 7 mei, 21 mei, 11 juni en 25 juni.

*Tip 2:* Vraag een van je scrummentoren om een pakje *planningspoker* kaarten.

*Deadline:* 25 april: Sprint backlog op Trello + iedereen lid van Trello.

### 4.3 Scrum standup meetings

Elke woensdag-, donderdag-, en vrijdagochtend vindt tussen 8.45 uur en 9.45 uur een scrum standup meeting plaats van maximaal 10 minuten. Elk team heeft 2 scrummentoren, en waar mogelijk is altijd een van de mentoren aanwezig bij de scrum standup meeting. Tijdens de meeting blijven de teamleden staan (vandaar 'standup' meeting) en beantwoordt elk teamlid de volgende drie vragen:

1. Wat heb je bereikt?
2. Wat ga je vandaag doen?
3. Wat belemmert je voortgang? (problemen)

De vergadering mag maximaal 10 minuten duren dus problemen worden buiten de vergadering verder besproken. Teamleden helpen elkaar (na de scrum standup meeting) bij het oplossen van obstakels die de voortgang belemmeren. Elk team heeft op woensdag, donderdag en vrijdag de gehele morgen van 8:45 uur tot 12:30 uur een eigen projectruimte. Het precieze tijdstip wanneer één van je scrummentoren aanwezig is wordt bekend gemaakt via Blackboard.

#### 4.4 Choose a case/project

Schrijf een document van 1 A4 met waarin je de case beschrijft die je gekozen hebt. Het document mag gebruik maken van de teksten in paragraaf 1.1, 1.2, etc. Converteer je document naar **.pdf** formaat, noem het **project.pdf** en zet dit in de github repository onder de map '**design**'.

*Deadline: 25 april.*

#### 4.5 UML use case diagram

Maak met behulp van VisualParadigm een UML use case diagram voor de webapplicatie die je gaat ontwikkelen. Exporteer je VisualParadigm project met het diagram als **.vpp** file *en* als **.xmi** file, en zet dit in de github repository van het team onder de map '**design**'.

*Deadline: 1 mei.*

#### 4.6 UML klassendiagram

Maak met behulp van VisualParadigm een UML klassendiagram voor de webapplicatie die je gaat ontwikkelen. Exporteer je project met het diagram als **.vpp** file *en* als **.xmi** file, en zet dit in je centrale git repository onder de map '**design**'.

*Belangrijk:* We gebruiken het UML klassendiagram voor andere doeleinden dan in module 2. In module 2 was het klassendiagram een model van het systeem zelf, sterk gericht op Java, met veel details (methoden, constructors, abstracte klassen). Voor het project van module 4 moet het klassendiagram een model zijn van (de informatie die opgeslagen wordt over) de omgeving van het systeem, sterk gericht op het databaseontwerp (onafhankelijk van Java), met weinig detail.

*Deadline: 1 mei.*

#### 4.7 SQL Database ontwerp

Maak een bestand '**schema.sql**' met daarin het Database ontwerp dat afgeleid is uit het klassendiagram van paragraaf 4.6. Zorg dat het schema de primaire sleutels (primary keys) en vreemde sleutels (foreign keys) correct definieert. Zet het schema in de github repository van het team onder de map '**design**'.

*Deadline: 1 mei.*

#### 4.8 De mockup

Maak met behulp van een CSS framework een mockup van je applicatie. Je mockup bestaat uit een of meer HTML pagina's (die gebruik maken van de stylesheets van het framework). Maak een enkel archiefbestand van je mockup als **mockup.tar.gz**, **mockup.zip**, of **mockup.war** bestand (Zie paragraaf 4.9 voor **.war**, gebruik in dit geval nog geen Java code of JSP), en zet het in de github repository van het team onder de map '**design**'. Het is ook toegestaan om een mockup te maken met behulp van een teken- of designprogramma zoals bijvoorbeeld Sketch, of met pen en papier. Maak in dat geval een bestand **mockup.pdf**.

*Tip 1:* Zie [http://en.wikipedia.org/wiki/Mockup#Software\\_Engineering](http://en.wikipedia.org/wiki/Mockup#Software_Engineering)

*Tip 2:* Zie voor lijst van frameworks : [http://en.wikipedia.org/wiki/CSS\\_frameworks](http://en.wikipedia.org/wiki/CSS_frameworks)

*Tip 3:* Sketch: <http://www.bohemiancoding.com/sketch/>

*Tip 4:* Pen: <http://en.wikipedia.org/wiki/Pen> en papier <http://en.wikipedia.org/wiki/Paper>

*Tip 5:* Gebruik op 2 mei termen als *HTML5*, *Responsive* en *mobile-first* om 'hip' te klinken.

*Deadline: 1 mei.*

#### 4.9 Sprint review meeting

Er vinden 5 sprint review meetings plaats, om de twee weken op vrijdag om 13.45 uur. Tijdens de sprint review meeting demonstreert het team het werk dat is gedaan tijdens de afgelopen sprint aan de belanghebbenden (stakeholders). Het team legt ook uit welke taken niet gehaald zijn. Zet het materiaal dat je gebruikt voor de presentatie (bijvoorbeeld slides) in de github repository van het team onder de map 'meetings', elk bestand moet de prefix **review** hebben, bijvoorbeeld: **review20140502.odp**.

*Tip:* [http://en.wikipedia.org/wiki/Scrum\\_%28software\\_development%29#End\\_meetings](http://en.wikipedia.org/wiki/Scrum_%28software_development%29#End_meetings)

*Review meetings:* Om 13.45 uur op 2 mei, 16 mei, 6 juni, 20 juni, 4 juli.

#### 4.10 Sprint retrospective

Onder leiding van de scrum master wordt de afgelopen sprint geëvalueerd door de volgende twee vragen te beantwoorden: 1) Wat ging er goed tijdens de sprint? 2) Wat kunnen we verbeteren in de volgende sprint? Zet een verslag van de sprint retrospective in de github repository van het team onder de map 'meetings'.

*Tip:* Houd de sprint retrospectives op 7 mei, 21 mei, 11 juni en 25 juni.

#### 4.11 Compilatie naar WAR met behulp van Maven

Het team volgt de standaard Maven lay-out voor Java projecten, zoals uitgelegd in de eerste practicummiddag op 22 april. Zorg dat je webapplicatie 'out-of-the-box' compileert door alle afhankelijkheden (dependencies) in the Maven **pom.xml** file te zetten. Zet de java code en de Maven pom.xml in de github repository van het team.

#### 4.12 De code wordt getest met unit tests

Zoveel mogelijk code wordt getest met JUnit unit tests. Zorg ervoor dat, voor unit tests die de database gebruiken, deze werken op een lege PostgreSQL database, waarbij de tests zelf test data toevoegen en verwijderen (zie ook de JUnit annotations @BeforeClass en @AfterClass).

*Tip 1:* Zie <http://en.wikipedia.org/wiki/Junit>

*Tip 2:* Er zijn tools voor 'code coverage', bijvoorbeeld EclEmma: <http://www.eclEmma.org/>

#### 4.13 Test-driven development

Unit tests worden gecommited in de github repository voordat de te testen code volledig geïmplementeerd is. Gebruik een nieuwe git branch voor code waarvan de test faalt, zorg dat de JUnit tests altijd goed gaan in de HEAD. Check nooit code in dit niet compileert.

*Tip:* [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)

#### 4.14 Java Servlets

De code maakt correct gebruik van Java Servlets.

*Tip:* Zie [http://en.wikipedia.org/wiki/Java\\_Servlet](http://en.wikipedia.org/wiki/Java_Servlet)

#### 4.15 RESTful web services en XML

De code maakt gebruik van Jersey voor RESTful web services.

*Tip:* Zie <http://en.wikipedia.org/wiki/RESTful>



#### 4.16 PostgreSQL en JDBC

De software met gebruik maken van PostgreSQL door middel van JDBC.

*Tip:* Zie [http://en.wikipedia.org/wiki/Java\\_Database\\_Connectivity](http://en.wikipedia.org/wiki/Java_Database_Connectivity)

#### 4.17 Framework voor HTML, CSS en Javascript

De software maakt gebruik van een vrij te kiezen CSS framework, zie ook paragraaf 4.8.

#### 4.18 Login faciliteiten

De software maakt gebruik van (veilige) login voor (sommige) gebruikers.

#### 4.19 Team work

Elk teamlid draagt bij aan Git, Trello, de presentaties, etc. Er zijn geen teamleden die onevenredig veel doen. Teamleden kunnen per sprint wisselen van taken, zodat iedereen een keer de scrum master kan zijn.

#### 4.20 Werkend systeem

Het systeem draait op de centrale Tomcat 'deployment' server: <http://datainfo.ewi.utwente.nl>. Vraag de docent voor de loginnaam en het wachtwoord van de server. Voeg de url van de applicatie (bijvoorbeeld [http://datainfo.ewi.utwente.nl/di00\\_appje](http://datainfo.ewi.utwente.nl/di00_appje) als je applicatie 'di00\_appje' heet) toe aan de file **README.md** in de github repository van het team.

*Waarschuwing:* Met je groeplogin kun je applicaties van andere groepen beheren, en overschrijven. Het is daarom verplicht om je als prefix van je applicatiennaam 'diXX\_' te nemen, waarbij XX je groepsnummer is. Alle acties op de server worden gelogd.

*Tip:* Ambitieuze groepen die bereid zijn een domeinnaam aan te vragen voor hun applicatie, kunnen het domein laten wijzen naar de centrale Tomcat server. Geef je domeinnaam en de applicatiennaam door aan de docent, zodat we de server kunnen configureren voor de domeinnaam. NB Een eigen domeinnaam levert geen hoger cijfer op.

#### 4.21 Ethische richtlijnen

Het team schrijft een document over de analyse van ethische aspecten. Converteer je document naar **.pdf** formaat, noem het **ehtics.pdf** en zet dit in de github repository onder de map 'design'.

*Meer informatie:* Tijdens de colleges Ethiek op 14 mei, 27 mei en 11 juni.

#### 4.22 Analyse van security

Het team doet een analyse van de security aspecten van het systeem, en zorgt dat de applicatie beschermt is tegen SQL-injectie en cross-site scripting.

*Meer informatie:* Tijdens de security dagen: 16 en 17 juni.

#### 4.23 Concurrency

De webapplicatie maakt correct gebruik van database transacties en isolation levels.

#### 4.24 Science Challenges

Deelname aan Science Challenges is niet verplicht. Zie hoofdstuk 5.

## 5 Science Challenges

Groepen die extra uitdaging willen en die hun kennis nuttig willen aanwenden, kunnen meedoen met Inter-Actief Science Challenges. Science Challenges organiseert een challenge speciaal voor Data & Informatie. Deelnemers van de challenge kiezen één case uit vier cases die aangeleverd zijn door Concordia, Strategie & Beleid van de UT, en Twente Academy. Woordvoerders van deze organisaties zijn (via de scrummentoren) beschikbaar om de cases toe te lichten. Bijeenkomsten van Science Challenges vinden elke woensdagmiddag 8/9 uur plaats. Voor Science Challenges geldt:

- Het meedoen aan Science Challenges is niet verplicht voor het halen van de module;
- De Science Challenges webapplicaties worden voor het halen van de module precies hetzelfde beoordeeld als de eigen case, dat wil zeggen dat de applicatie moet voldoen aan de randvoorwaarden van hoofdstuk 2 en hoofdstuk 4;
- Het meedoen aan Science Challenges levert geen extra risico op dat je de module niet haalt; integendeel, het verhoogt de kans om goed te scoren op het project;
- De Science Challenges webapplicaties worden beoordeeld door een jury van lokale ondernemers en de belanghebbenden. De beste applicatie krijgt een prijs;
- De woensdagmiddagen worden georganiseerd door Inter-Actief. Je kunt hier bijvoorbeeld hulp krijgen van ouderejaars studenten.

Meld je aan op: <http://sciencechallenges.nl>.

### 5.1 Case 1: Concordia Kunstmenu

Concordia laat jong en oud, kenners en leken, vaste gasten en toevallige voorbijgangers op een aangename manier genieten van hedendaagse film, theater en beeldende kunst.

*Willem Jaap Zwart – Directeur Concordia*

Het Kunstmenu is een cultuureducatieprogramma voor het gehele primair onderwijs in Enschede. Alle basisschoolleerlingen kunnen gedurende hun schoolloopbaan structureel ten minste één keer kennismaken met zeven disciplines. Dit zijn: beeldend, drama/theater, multimedia, dans, literatuur/creatief schrijven, muziek en cultureel erfgoed. Concordia is de makelaar en bemiddelaar voor het aanbod en scout professionele cultuuractiviteiten uit het plaatselijke, regionale en landelijke aanbod. De commissie cultuureducatie met vertegenwoordigers uit het culturele en onderwijsveld in Enschede adviseert over de inhoud en vernieuwingen van het Kunstmenu. Vrijwel alle basisscholen en culturele instellingen (Wilminktheater, Nationale Reisopera, TwentseWelle, Tetem, Rijksmuseum Twenthe etc) nemen deel aan het Kunstmenu. Aanvraag van activiteiten vindt online plaats. Concordia honoreert vervolgens de aanvragen en koppelt op die manier vraag en aanbod aan elkaar. Concordia wil in het kunstmenu graag het volgende: 1) Opslag van het aanbod in een database; 2) Gestructureerde zoekmogelijkheden in het aanbod; 3) Aansluiting van elementen uit de doorlopende leerlijnen op het aanbod ; 4) Online aanmelden; 5) Toekennen en plannen. Meer informatie op: <http://sciencechallenges.nl>.

### 5.2 Case 2: Concordia Kunstverhuur

Concordia's kunstverhuur is vergelijkbaar met een bibliotheek. Met een jaarabonnement kun je om het half jaar een nieuw werk uit de collectie huren. Je kunt de huurperiode verlengen en er is geen aankoopverplichting. Zowel bedrijven als particulieren kunnen lid worden. Bij bedrijfslidmaatschappen gaat het veelal om een aantal kunstwerken die door het hele kantoor heen worden opgehangen. Concordia heeft een collectie van ca. 1700 kunstwerken, verdeeld over een collectie die voor particulieren bestemd is en een collectie voor bedrijven. Dit onderscheid is vooral

gemaakt voor bijvoorbeeld series schilderijen. De collectie bestaat voornamelijk uit schilderijen, maar omvat ook beelden. De laatste jaren is de kunstverhuur terug gelopen. Concordia wil de kunstverhuur graag nieuw leven in blazen. Daarbij hebben wij de volgende wensen: 1) uitgebreidere zoek- en reserveringsmogelijkheden ; 2) adviesfunctie ; 3) samenstellen van een eigen online-expositie ; 4) aanspreken van nieuwe doelgroepen ; 5) mogelijkheden tot het gebruik van sociale media vanuit de catalogus. Meer informatie op: <http://sciencechallenges.nl>.

Every great innovation is in my opinion based on two components: a creative part and a practical one. The creative part, mostly based on creating an idea of a service/product is always useless without the practical execution and the acquisition of necessary resources.

*Hadi Ghorashi – Student of the University of Twente*

### 5.3 Case 3: UT Think Tank, Non Stop

Think Tank Non Stop is a web application for the University of Twente that motivates and attracts young creative people to publish their designed, innovative ideas. People that are actually doers (engineering, programming, textile students etc.) can then take up those ideas, rate them and choose the one they themselves consider the most worthy to execute. Most entrepreneurial workshops use the same method: bring executors and creators together. Why shouldn't we, the University of Twente support exactly the same idea here to strengthen our identity as THE entrepreneurial university? The monitoring of the website could be done through a students' rating system, which kicks out the perceived less potential ideas, or ones more depending on the budget, by someone who is indeed qualified to rate those ideas. Furthermore, the creation of a certain standard form that must be filled out (length of text), run through a text mining program before handed in, should be included to avoid the amount of spam within the site. Meer informatie op <http://sciencechallenges.nl>

Ik zou graag elk jaar een dag willen waarop we als UT'ers creativiteit en kennis met elkaar delen En met elkaar nadenken over: hoe zien we de toekomst? Waaraan moeten we onze aandacht en tijd besteden? Welke kansen pakken we? Wat gaan we doen om ook over 20 jaar nog zichtbaar te zijn?

*Susanne Wichman – Strategie & Beleid, Universiteit Twente*

### 5.4 Case 4: Twente Academy Social Network

Twente Academy helpt middelbare scholieren beter te presteren en bereidt ze voor op het student zijn. Inmiddels is het aantal projecten gestegen naar tientallen. Deze worden uitgevoerd door een poule van 200 studenten van de universiteit. Een groep van 35 studenten heeft een coördinerende of administratieve taak waarvoor ze regelmatig op kantoor werken. De rest van de medewerkers werkt vooral tijdens projecten, zoals Science on Tour, de begeleiders van de Eindexamenkampen en de Persoonlijk Assistent Leraren. Doordat veel medewerkers onregelmatig werken en ze weinig contactmomenten hebben is er weinig communicatie onderling. Afstemming gebeurt door middel van mail en suggesties of ideeën vinden vaak niet hun weg naar de coördinator. Daarom wil Twente Academy uitzoeken wat de mogelijkheden zijn voor een social network om de afstemming, ideeën en problemen vaker met elkaar te kunnen bespreken. Ook geeft het begeleiders meer mogelijkheid om invloed uit te oefenen op hoe er gewerkt wordt. Een ander voordeel is dat er veel sprake is van verloop onder de studenten en de kennis continue verdwijnt. Een social network zou het mogelijk maken om deze onderwerpen terug te zoeken.

Met projecten als de Masterclasses, de Eureka!Cup, de Mathrace worden de leerlingen gemotiveerd om verder te kijken dan de lesstof en om zichzelf uit te dagen.

*Nadja van Kessel – Teamleider Twente Academy*

## 6 Conclusie

Als je eenmaal hier bent beland, gefeliciteerd! Je bent *of* klaar om het het project te beginnen, *of* misschien wel helemaal klaar met het project, want wie leest er tegenwoordig nog handleidingen van A tot Z? Mocht je al klaar zijn, dan zul je gemerkt hebben dat sommige onderdelen gemakkelijk waren, maar andere onderdelen enorm frustrerend. Het installeren van een enkel stuk software, bijvoorbeeld, kan je uren bezig houden. Het juiste gebruik van een softwarebibliotheek vereist soms dat je je inleeft in de programmeur van de bibliotheek.

... somewhat unique to programming is that it consists of near constant failure. Unlike learning other skills where one can expect to be reasonably competent after sufficient practice, programming largely consists of constantly failing, trying some things, failing some more, and trying more things until it works. One of the biggest differences between experienced and novice programmers is that experienced programmers know more things to try.

*Alicia Liu, Overcoming the Imposter Syndrome (<https://medium.com/tech-talk/bdae04e46ec5>)*

### 6.1 De mythe van de 'echte' programmeur

Mocht je je zorgen maken over je eigen kunnen, mocht je denken dat je geen 'echte' informaticus of 'echte' programmeur bent, mocht je van mening zijn dat de docenten van deze module je volledig in het diepe hebben gegooid, lees dan Alicia Liu's blog post hierboven. Een goede programmeur heeft een aantal eigenschappen die goed zijn om hier een laatste keer te herhalen: (en we bedoelen niet dat ze vale T-shirts dragen en van Star Trek houden!)

- Wees geduldig en systematisch, probeer verschillende oplossingen één voor één.
- Lees handleidingen, the 'fine' manual: <http://en.wikipedia.org/wiki/RTFM>
- GIYF: Google is your friend! of Stackoverflow, of Duckduckgo (als je licht paranoïde bent). In deze handleiding hebben we gekozen om te verwijzen naar Wikipedia als startpunt voor het vinden van informatie.
- Leg grotere problemen uit aan een van je teamleden, of als je teamleden er niet zijn, aan je rubber eendje: [http://en.wikipedia.org/wiki/Rubber\\_duck\\_debugging](http://en.wikipedia.org/wiki/Rubber_duck_debugging)

Als je met computers werkt, dan moet je...? geduld hebben!

*Luís Ferreira Pires (tegen zijn kinderen)*

### 6.2 Evaluatie van de module

Tenslotte, dit is het eerste jaar dat we deze module draaien. We zijn ambitieus geweest met veel onderwijsvernieuwingen en veel inhoudelijke vernieuwingen. Sommige details zullen we tijdens het draaien van de module nog aanpassen en bijstellen. We krijgen graag feedback van studenten, studentassistenten, en docenten die betrokken zijn bij deze module. De evaluatie van de module wordt gedaan door een evaluatiegroep bestaande uit twee docenten, vier of vijf studenten, en een vertegenwoordiger van de OEC (Onderwijs EvaluatieCommissie). Studenten, studentassistenten, en docenten die iets in willen brengen in de evaluatiegroep kunnen contact opnemen met de OEC-vertegenwoordiger, Eva van den Eijnden.