ICA

INFORMATICA
COMMUNICATIE
ACADEMIE

project guide

develop a web application

year: 2020 - 2021
opleiding: HBO-ICT
authors: Lars Tijsma
Sander Leer
Robert Holwerda
Bart van der Wal
version: 1.0

HAN

image source: http://www.evolutionoftheweb.com/

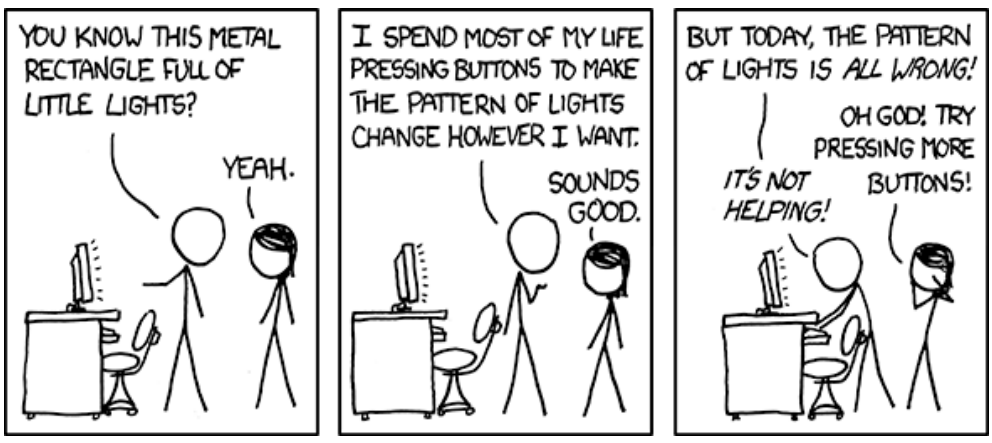
table of contents

1. introduction

The second half of the DWA semester is a large project assignment. You will participate in a team, and work towards the design and implementation of an interesting web application.
This Project Guide describes how this project is organized, and what we expect from you, and what you can expect from us in return.
We wish you a fun and productive time.

We may encounter compelling reasons to change aspects of the project in mid-flight. When that happens, we will communicate those changes clearly via e-mail. We will also update this guide. You can always find the latest version of this Project Guide on GitHub.

We are very interested in any feedback from students about the project. Please make your opinions known to any of the teachers who have been assigned to your group. You can also turn to them for clarification if anything in this document is unclear.
For any remarks, suggestions or complaints about this Guide in particular, please contact the maintainer of this document, Lars Tijsma (lars.tijsma@han.nl).



comic source: http://xkcd.com/722/

2. overview of the project

your team

By the time your project starts you will already have been assigned to a group. This happens in the course-phase before the actual project starts. Teams usually consist of 4 to 6 people. You will be assigned to a group by the teachers.

the assignment

Assignments are selected by the teacher team, who are open for proposals from students or companies (see appendix B for conditions on proposed project assignments).

The assignment is always about the design and programming of a web application using a modern client-side framework, a NoSQL database, Node.JS and the WebSocket protocol.

method

The team will use the Scrum method to manage its activities and communication. You will receive instructions on Scrum in several workshops. Scrum is an iterative, and agile, method. *Iterative* means that the team divides its time into short periods (of two weeks, in our case), called *iterations* or *sprints*. At the end of each sprint the team, and other stakeholders will review the progress that was made. At the start of the next sprint, new decisions can be made about what the team should do in the coming sprint, based on a to-do list, and on the previous review. An important aspect of these sprints is that for each feature you create, the design, programming *and* testing of the code for that feature is part of the same sprint.
Scrum is one of several members of the *Agile* family of software engineering methods. Being *agile* means multiple things. Agile methods 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.
This is taken from http://www.agilemanifesto.org/. There are also twelve *Principles of Agile Software*: http://www.agilemanifesto.org/principles.html.

In the DWA project you will see this agile mindset in the following ways:

- Documentation requirements are relatively light. See the section on the Software Guidebook.
- There is no preconceived set of requirements for each assignment. Sprint planning meetings are used to make agreements between the product owner (a teacher) and the team. This happens multiple times during the course of the project.
- You are expected to play an active role in shaping the direction of the project and the shape of the end result.

coaching and guidance

As a team, you will see three teachers who will support you, require things from you and grade your results:

- One teacher is your **product owner**. You will meet him at the start of each sprints, and at the end of each sprint. The product owner is the one with whom you decide what to design and build. He is also the one who decides whether your sprint-results are OK.
  The product owner is a role that is part of the Scrum method.
- Another teacher is your **project coach** ("procesbegeleider" in Dutch). With him you have e weekly discussion about the *process* your team is in. The project coach is also the first teacher to approach for technical support or if the team needs facilities that haven't been provided yet.
- The third teacher is your **professions skills coach.** Like the project coach, he or she will support the process, and help grade you on those aspects of the project.

deliverables

The most obvious deliverable is your product: the web application you wrote. You will use a Git repository to maintain your code base, and **the entire repo** is the deliverable for the product.
The second deliverable is the Software Guidebook, which contains all the information needed by other software engineers, outside your team, who

may have to maintain or extend your product.
Then there are some documents that you use to “control” the process: The most important one is the project plan. You will also maintain a product backlog. This is basically the to-do list for your project (the name comes from the Scrum method). The team will, of course, make notes of meetings.
All the deliverables described are documents that you create and maintain as a team. There are also some documents that every team member creates by him- or herself: Every day the team members will maintain the log (logboek). Every sprint you will write a short report about the functioning of your team called the Retrospective. Every sprint you will plan a short personal research and write a short report about the result of this research called the Personal Research (persoonlijke onderzoeken). You’ll publish your individual documentation in your personal repo.

workshops

The DWA project is not just a place where you demonstrate the skills and knowledge from the courses. There is new stuff to be learned. This will take the form of a set of workshops about the following subjects:

- User Stories & Scrum
- Skills
- Automated testing
- Technical documentation (Software Guidebook)

All workshops will be scheduled in OW 1-9 and 2-1.

grades

There are three moments where you will receive grades that contribute to your passing (or failing) the project.
After your team has handed in the final version of the **project plan**, there will be a short verbal examination in which each team member must show both his or her personal contribution, as well as his or her knowledge and involvement with all the other parts of the document.
Secondly, there is a **mid-term assessment** halfway through the project. Your product owner and another teacher will grade the work you’ve done up to that point. They will also give feedback on how to improve your results in the weeks ahead.
Then there is the **final assessment** at the end of the project. Your work will be graded by all teachers who are involved with your group.
Both assessments take into account all work you do on the project:

- the project plan;
- application itself;
- the way the team uses the Scrum method;
- the software guidebook;
- the use of automated testing
- how professional you've operated with regard to communication, cooperation and project skills.

Both the midterm assessment and the final assessment start with a group-assessment in which the team shows its work to the teachers. But there is also an individual aspect: You may receive a lower or higher grade than the grade for the whole team, based on:

- the effort put into the Result-oriented Planning and the Personal Retrospectives;
- your individual contribution to the team products;
- your conduct during the assessments;

3. competencies

At ICA, what we want you to learn is defined in terms of competencies: things you should be able to do. One aspect of the project is that it is not only an opportunity to acquire the skills involved, but also to demonstrate your competency.
These are the competencies for the DWA project. Currently they are still in Dutch. A next version of this guide will contain an English translation. Each competency is described by a list of things you do to demonstrate the competency.

C1. De student maakt op basis van een klantvraag, een product backlog, en onderhoudt die tijdens het project.

- Vertaalt de informele specificatie in een set user-stories, en leidt daar back-log items uit af.
- Voorziet ieder backlog-item van een doordachte “definition of done”.
- Brengt eigen ideeën en overwegingen in, in de gesprekken met de product-owner over de back-log.

C2. De student onderzoekt nieuwe ontwikkelingen m.b.t. (web)technieken en/of methoden op hun toepasbaarheid in het project.

- Draagt bij aan een onderzoek waarvan de vraag relevant is in de context van een organisatie of een beroepstaak.
- Doorloopt de onderzoekscyclus en gebruikt geschikte vormen van onderzoek.
- Schrijft mee aan een researchadviesrapport dat voldoet aan de ICA controlekaart.
- Maakt inschattingen over de kwaliteit van kennis en gekozen pragmatiek.

C3. De student maakt architectuur beslissingen voor de verspreiding van business-logic over server, clients en 3rd-party services, en een ontwerp maken voor de resulterende API's.

- Ontwikkelt een Technisch Ontwerp waarin zichtbaar is wat de verantwoordelijkheden zijn van de verschillende hoofdcomponenten van het systeem.
- Ontwerpt en documenteert een (realtime) webbased API waaruit helder wordt hoe client en serversoftware samenwerken.

C4. De student realiseert een prototype van een realtime web-applicatie.

- Maakt een gebruik van beschikbare moderne ontwikkeltools voor webdevelopment.
- Houdt zich aan code-conventies en standaarden zoals die in het team zijn afgesproken.
- Schijft code die robuust is tegen fout-condities en illegale invoer.
- Schrijft leesbare en onderhoudbare code.

C5. De student test de werking van het systeem op een gestructureerde en geautomatiseerde manier.

- Gebruikt de aangeleerde tools voor geautomatiseerd testen.
- Ontwikkelt geautomatiseerde tests tijdens het ontwikkelen, niet achteraf.

C6. De student documenteert de technische aspecten van de applicatie ten behoeve van opvolgers.

- Beschrijft de overwegingen en oplossingen die achter zijn/haar code en ontwerpen schuil gaan.
- Verplaatst zich in de informatiebehoefte van opvolgers.
- Behandelt, samen met groepsleden, alle aspecten die relevant zijn voor opvolgers.
- Schrijft conform de ICA controlekaart.

C7. De student opereert op professionele wijze qua communicatie, samenwerking en projectmatig werk.

- Kan een doel- en doelgroepgerichte tekst schrijven, die voldoet aan de ICA contrôlekaart.
- Presenteert de inhoud op een voor het publiek heldere en logische wijze en voldoet aan de checklist presenteren.
- Zet gespreksvaardigheden effectief in om het doel van het gesprek te bereiken.
- Schept aan het begin van een project aantoonbaar en actief voorwaarden voor een optimale interactie in het team.
- Kan de samenwerking in een groep analyseren, bijsturen en zijn interventies evalueren om een groepsopdracht tot een goed einde te brengen.
- Kan een project aan de hand van gegeven eisen of methode adequaat inrichten.
- Stuurt actief en aantoonbaar het verloop van het project (voortgang en evaluatie) door gebruik te maken van de beheersfactoren.
- Laat zijn ontwikkeling zien door middel van kritische evaluatie van en reflectie op eigen handelen.

4. timelines

To get a more exact sense of the structure of the entire project, it helps to start with the structure of an individual sprint.

sprint timeline

Here is what a typical two-week sprint will look like:

start of sprint:	**Sprint Planning Meeting** with product owner
every day in each sprint:	update your **Personal Log**
halfway during week 1:	meeting with the **project coach**
	meeting with the **skills coach**
halfway during week 2:	meeting with the **project coach**
	meeting with the **skills coach**
a day before meeting the coach:	publish your **retrospective** on your personal repo
	publish your **log** on your team repo
a day before meeting the skills coach:	publish your **personal research** on your personal repo
end of the sprint:	**Sprint Review Meeting** with the product owner

The exact moments of these meetings will not be the same for every team. It depends on the schedules of your teachers. For example: it is most obvious to have each sprint start on a Monday, and and on a Friday. But it may be better to move the sprints a day, and have them start on Tuesdays and end on Mondays.

5. rules

There are some rules.

- You work 40 hours every week.
- You work, with your team, in MS Teams.
- When there is a compelling reason why you can't be with your team, you will notify both the team and your project coach.
- If you have a pressing reason why you can't be in Teams for a particular day, discuss this with your coach in advance.
- If you fall ill, notify your coach as soon as possible.
- You will respect and abide by agreements made with the team, and with the teachers.

- The team uses GitHub for source code management.
- The team will make daily back-ups for all documents that are not committed to GitHub.
- The team will make sure a written record exists of agreements and to-do items that are not suitable for the product backlog.

Your project coach will notify you if you're breaking these rules to often. He may give you a final warning. If, after receiving a final warning, the offending behavior is seen again, you will be fired from the project. In that case you will receive no credits ("studiepunten" in Dutch) for the project.