

# Projectopdracht “Software Engineering Lab 1”

## Inleiding

Het vak “Software Engineering Lab 1” wil jullie de principes voor het ontwikkelen van een software-applicatie bijbrengen, van het formuleren van functionele en niet-functionele vereisten tot het opleveren van geteste en gevalideerde code. . We richten ons hierbij in de eerste plaats op mobiele toepassingen. De projectopdracht voor dit vak bestaat dan ook uit het ontwikkelen van een mobiele app op het Android platform in groepjes van 4 studenten. We hadden ervoor kunnen kiezen om alle groepjes eenzelfde, vooraf goed omlijnde opdracht toe te vertrouwen, maar we hebben dit bewust NIET gedaan. De voornaamste redenen hiervoor zijn:

- Het uitdenken van een interessante toepassing, en het achterhalen van de nodige functionaliteit van deze toepassing is een van de kerntaken van software engineering. Het in kaart brengen van behoeften en deze omzetten naar een prototype is dan ook een belangrijke competentie voor dit vak.
- Onderzoek heeft aangetoond dat enthousiasme en inzet voor een project waarvan studenten ownership nemen (omdat ze het zelf uitgedacht hebben) vele malen hoger liggen dan voor een project waarvan onderwerp en krijtlijnen vastliggen.

De opdracht voor het project is dus vrij generiek: “Bedenk een originele en nuttige mobiele toepassing, en maak hiervoor een implementatie op het Android platform”.

## Projectverloop

Om dit project tot een goed einde te brengen, voorzien we 6 stappen, als tussentijdse opdrachten. Het doel is hierbij steeds om het eindproduct zo kwalitatief mogelijk te krijgen.

### Stap 1: Pitch

Een “pitch” is een korte (10 minuten) presentatie, waarin je kernachtig een publiek van je project wil overtuigen. Vragen die je hierbij poogt te beantwoorden zijn:

- Welke behoefte vult de app in?
- Wat is het typisch gebruik van de app en waarom is dit nuttig (b.v. tijdswinst, aangeboden functie bestaat (nog) niet, of bestaat slechts in beperkte mate)?
- Waarom zou je een vriend aanraden de app te installeren?

**Voorbereiding** : in een hoorcollege [13 februari] krijg je richtlijnen i.v.m. het houden van een goede pitch.

### Thuiswerk :

- Rond eerste projectideeën overleggen met je studiegenoten
- Op basis van interesse groepjes van 4 studenten vormen
  - Je registreert je voor een groepje op Minerva, VOOR 16 februari 2018

- Projectidee verder verfijnen
- Presentatie van 10 minuten maken, per groep van 4 studenten vorm je 2 subgroepjes, die elk een eigen pitch maken.

### Evaluatie

- Pitch-presentatie per 2 studenten op **27 februari**.
  - o Jullie verdelen zich in 2x2. Elk groepje van 2 studenten geeft een aparte pitch.
- Je zal hierbij op de volgende manier geëvalueerd worden:
  - o Heb je rekening gehouden met de richtlijnen voorgesteld in het hoorcollege?
  - o Is de app op een duidelijke en overtuigende manier voorgesteld?
  - o Is de functionaliteit van de app duidelijk?
  - o Kan de app binnen de opgegeven periode en binnen de studiepunten van dit vak ontwikkeld worden?

### Hoe kies ik een app ?

- Kies een onderwerp waarvoor je enthousiast bent (je zal vrij veel tijd aan het ontwikkelen van de app besteden)
- Wees realistisch in je ambities: het is moeilijk in te schatten hoeveel tijd een project zal kosten. Onderschattingen komen veel meer voor dan overschattingen! Je zal tot een werkend product moeten komen in 2 iteraties van telkens 3-4 weken. De laatste lesweken worden er geen hoorcolleges of werksessies voorzien, zodat je deze tijd ook kan spenderen aan de ontwikkelingen.
- Stel jezelf de vraag: waarom is dit een mobiele toepassing? Indien de functionaliteit even goed via een desktop kan aangeboden worden, is het twijfelachtig of het om een interessante mobiele app gaat.
- Het project wordt publiek gepresenteerd (indien je een idee/concept geheim wenst te houden, kies je dit dus beter niet voor dit project)
- Je mag geen project kiezen waarvoor je betaald wordt.
- Je mag uiteraard bibliotheken/raamwerken gebruiken, maar je mag niet vertrekken van een codebase van iemand anders.

### Stap 2: Workshop HCI

Efficiënte en intuïtieve interactie met de gebruikers van de mobiele app is essentieel om je app een succes te maken. Hoe de interactie tussen mens en computer moet gestructureerd worden is een wetenschap op zich.

Omdat die interactie zo belangrijk is om een succesvolle app te bouwen, starten we het project met het ontwerpen van de verschillende schermen van de app en de navigatie tussen die schermen.

**Voorbereiding:** in een gastcollege [**23 februari**] worden de basisregels m.b.t. HCI uitgelegd, in het bijzonder ook rekening houdend met de Android context. Ook wordt ingegaan op een tool om snelle prototypes te bouwen, aan de hand waarvan het mogelijk is om interfaces op hun gebruiksvriendelijkheid te evalueren.

**Thuiswerk:**

- Maak user stories voor je app (zeer korte beschrijving van **wat** elke soort **gebruiker** met de app kan doen om een welbepaald **doel** te bereiken)
- Vul het dashboard sjabloon voor je app in (hierin leg je vast hoe je zal testen)
- Bouw een prototype van de verschillende schermen van je app (tool naar keuze, Marvel wordt aangeraden)
- Maak een release backlog: bepaal welke user stories in de (enige) release van je app zullen steken.

**Evaluatie:** in een workshop [13 maart] kan je je prototype via een aantal tools evalueren, en indien nodig bijsturen. De criteria waarop je wordt geëvalueerd zijn:

- o Is er voldoende nagedacht over de verschillende gebruikers en hun behoeften?
- o Zijn de concepten van het hoorcollege correct toegepast?
- o Zijn de user stories voldoende uitgewerkt?
- o Is de functionaliteit van de app duidelijk uit de user stories?

### Stap 3: Sprint 1

Om je app nu te coderen, voorzien we 1 release waarbinnen 2 sprints gepland worden. In deze eerste sprint probeer je de belangrijkste user stories te realiseren, maar hou je ook rekening met de beschikbare tijd.

**Voorbereiding:** Sprint backlog plannen [werksessie 20 maart]

- Selecteren van de user stories die in deze sprint zullen gerealiseerd worden
- Verfijnen van de user stories, zodat ze voldoende detail hebben om het programmeren aan te vatten
- Inschatten van de tijd (werkuren) die nodig is om elke user story te realiseren

**Thuiswerk:** Realiseren sprint backlog

- Overleg (liefst dagelijks – SCRUM meeting) binnen je team hoe de planning voor de nabije toekomst eruit ziet (wie doet wat ?)
- Hou de vordering van de sprint in een burn-down chart bij
- Hou je code actueel op een github repository
- Overleg met je mentor (fysiek of elektronisch) 1x per week

## Evaluatie [werksessie 27 april]

### Code evaluatie:

- *Indien je in je code 2 of meer functies opmerkt die identiek zijn, op enkele details na, maak dan een algemenere functie.* Maak eventueel gebruik van hogere orde abstracties. In projecten komen we regelmatig het zogenaamde ``knip-en-plak-hergebruik" tegen. Knip-en-plak-hergebruik is een verbloemde term voor codeduplicatie, en moet ten allen prijze vermeden worden.
- Gebruik duidelijke naamgevingen in je code, die de bedoeling van de variabele of functie ook voor anderen duidelijk maakt.
- *Zorg dat er nergens in je code vastgebakken waarden voorkomen.* Vastgebakken getallen komen meestal voor als boven- of ondergrenzen (bijvoorbeeld het maximum aantal spelers). Je kan dit simpelweg vermijden door de hard gecodeerde waarden als constanten te definiëren. Zorg er wel voor dat je code nergens steunt op een specifieke waarde van een constante.
- *Vermijd het gebruik van globale variabelen.* In het algemeen maken globale variabelen je code onduidelijk en onleesbaar. Het gebruik van globale variabelen midden in je code zorgt er bovendien voor dat je code niet te hergebruiken valt. Als je toch een globale variabele gebruikt in een functie, dan is het een goed idee deze mee te geven als parameter. Zo steunt de betreffende functie niet op het bestaan van deze globale, en kan aldus hergebruikt worden.
- *Hou je procedures kort.* Procedures die in hun geheel op het scherm passen zijn makkelijker te begrijpen. Op papier zouden procedures van een A4 lang uitzonderlijk moeten zijn, langer is zeker uit den boze.
- *Bij hergebruikte code dient steeds verwezen te worden naar de bron.*
- *Documenteer je code zodat ook anderen deze kunnen begrijpen.* Niet gedocumenteerde code is per definitie slechte code.
- *Maak maximaal gebruik van de ingebouwde functionaliteit van Android.* Zoek eerst grondig naar een mogelijkheid in de API vooraleer je zelf een feature probeert te implementeren. Let ook op met oudere code-voorbeelden op StackOverflow of in blogs: de Android API is met elke release sterk uitgebreid waardoor sommige zaken misschien efficiënter of eenvoudiger kunnen geïmplementeerd worden!

### Bespreek:

- de feedbackrapporten die je intussen ontving (zie stap 4).
- wat goed/fout liep in de groep

Neem dit mee als input voor planning sprint 2 (onmiddellijk aansluitend)

## Stap 4: Feedback geven

Feedback geven en krijgen is belangrijk om je app bij te sturen. Bovendien kan het nakijken van andermans code ook leerrijk zijn. Daarom wordt van jou verwacht om een app van een andere groep te evalueren op verschillende criteria.

**Voorbereiding:** Hoorcollege van 17 april

**Thuiswerk:** Evaluatie van de app + neerschrijven van beknopte evaluatie

**Evaluatie:** Rapporten worden geëvalueerd door begeleiders van het luik “Communicatie” en worden als input gebruikt voor sprint 2.

## Stap 5: Sprint 2

Hier volgen we hetzelfde proces als voor sprint 1, waarbij nu een aantal bijkomende user stories gerealiseerd worden, en waarbij ook de feedback op sprint 1 in rekening gebracht wordt.

**Voorbereiding:** planning sprint 2 [4 mei]

**Thuiswerk:** coderen van user stories sprint 2

**Evaluatie:** postersessie [15 mei]

## Stap 6: Postersessie

We sluiten het project af met een postersessie, waarbij het de bedoeling is je app in de kijker te zetten en ook een demo te geven aan je mede-studenten de projectbegeleiders.

**Voorbereiding:** hoorcollege waarin de principes van een goede poster worden toegelicht[24 april]

**Thuiswerk:** realisatie poster + maken van demo-scenario

**Evaluatie:** mondelinge presentatie van de poster en demo [15 mei]

Je wordt gequoteerd op de volgende criteria:

- ☐ poster
- ☐ demo en toelichting
- ☐ idee/toegevoegde waarde van de app
- ☐ technische uitwerking (rekening houdende met de beschikbare tijd voor dit vak)

## Deadlines

15 februari	Samenstelling groepen op Minerva
27 februari	Pitch geven
10 maart	User stories + release backlog op github repo gefinaliseerd
13 maart	Prototype gefinaliseerd voor HCI workshop
20 april	Release sprint 1 op github
26 april	Deadline feedback rapport
12 mei	Release sprint 2 op github
15 mei	Demo- en posterdag