

# Planering för Projekt OH-HAIKU

*Software Engineering Project - DAT255*

Linus Oleander  
Jesper Josefsson  
Victor Karlsson  
Jesper Gunnerling  
Eric Bengtsson

# Introduktion och syfte med projektet

Denna planeringsrapport avser projektet OH-HAIKU vars syfte är att utveckla en Android-applikation för att kontrollera om en text är skriven enligt diktformen Haiku.

## Behov

Twitter-användare har efterfrågat en applikation som avgör om en text är skriven enligt diktformen Haiku. Handledaren för projektet har utvecklat en applikation innehållande en enkel algoritm för att kontrollera detta, vilken har laddats ner tillräckligt många gånger för att påvisa att ett behov finns.

## Haiku

Haiku är en sorts diktform som härstammar från Japan. Enligt Svenska Haiku Sällskapet kan den exakta definitionen av en haiku variera.

Den mest naiva definitionen av en västerländsk haiku innebär tre rader text, med stavelserna fördelade enligt 5-7-5. Japanska haikun skrivs emellertid på en enda lodrät rad och man räknar inte stavelser, utan teckenljud. Japanska teckenljud är ofta kortare än västerländska stavelser vilket leder till att västerländska haikun ofta låter längre än japanska.

I detta projekt kommer vi att inledningsvis använda den naiva definitionen av västerländsk haiku.

Vidare kriterier, enligt Svenska Haiku Sällskapet:

- Den innehåller en årstidsangivelse
- Den är uppdelad i två delar för överraskningseffekt
- Den är inte sentimental
- Den för samman naturbetraktelser med livets villkor
- Den ska vara fåordig och undvika upprepningar
- Den ska helst inte innehålla personliga pronomina
- Den ska undvika alltför starka adjektiv

I mån av möjlighet kan något av dessa krav införlivas i applikationen.

## Kravspecifikation

### Grundfunktioner:

- Kunna identifiera och räkna stavelser i en text
- Kunna avgöra om texten uppfyller kraven för en haiku
- Kunna beräkna någon form av poäng för hur haikun, kanske på basis av hur nära den är en "korrekt" haiku
- Kunna berätta vilken rad som innehåller för många eller för få stavelser
- Kunna tweeta texten med haikupoängen
- Kunna skriva ut Certified Haiku på Twitter i fallet att texten är en fullständig Haiku
- Ett smidigt och trevligt GUI

### Eventuella funktioner:

- Kunna kontrollera om en användares sms är en Haiku
- Kunna analysera användare på Twitter. Hur många procent av tweetsen är Haikus? Hur Haiku är en användare?
- Införliva "mjuka" krav i poängalgoritmen:
  - Räkna personliga pronomina

- Leta ord som har med årstider att göra
- Leta ord som har med naturen att göra
- Leta efter upprepningar

## Projektorganisation och tillvägagångssätt

Vi planerar att undersöka möjligheten att använda Scrum inom projektet. Scrum är en metod för systemutveckling. Arbetet skall enligt denna metod färdigställas i funktionella faser där överlämningar mellan faserna skall ske tydligt. I projektet kommer således vissa arbetsroller att förekomma, tillsammans med obligatoriska möten och dokument. Arbetsrollerna består av en product owner, en scrum master och ett team. Product owner fungerar som kravställare, scrum master fungerar som brobyggare i teamet och försöker säkerställa och coacha för att teamet skall efterleva processen. Teamet är helt enkelt utvecklingsgruppen.

Product owner för teamet skall ansvara för product backlog vilket kan ses som en samling av prioriterade behov. Scrum master för teamet fastställs senare. Han skall ansvara för sprint backlog vilket innebär de behov som skall tas tillvara och implementeras i projektet. Varje sprint kommer att vara under allt från 1 vecka till 3 veckor. Teamet är uppdelat i olika ansvarsområden. Att ha ansvar för ett område innebär krav på avrapportering och uppdatering på veckobasis till teamet genom scrum master.

Arbetet kommer att stödjas av daily scrum, sprint review, sprint retrospective samt sprint planning. Daily scrum kommer att innehålla statusuppdateringar från teamet inom varje sprint och sköts via mail. Sprint review handlar om att köra en genomgång av varje sprint, och kunna uppvisa ny funktionalitet genom att köra produkten. Review-delen kommer främst att diskuteras under mötena som sker på veckobasis. En kort genomgång av vilka sprinters projektet kommer att innehålla sker under rubriken Projektets milstolpar och sprinters. Retrospective-delen handlar om att lära sig av föregående sprinters. Dessa diskussioner skall hållas under veckomötena. En mer grundlig Sprint-planning kommer att göras den 26:e mars.

## Ansvarsområden

Inom teamet kommer följande ansvarsområden att gälla:

Roll	Ansvarig
Project management	Jesper G, Eric
Rapportansvarig	Jesper G, Eric
Requirements	Samtliga
Testansvarig	Linus
GUI-Ansvarig	Jesper J
Algoritmansvarig	Victor

Rollerna innebär inte att personerna i fråga utför allt arbete på det området. Istället innebär rollen ett ansvar för att planera och delegera arbetsuppgifter på ett sätt som säkerställer att arbetet genomförs.

## Övriga utvecklingstekniker

Ambitionen är att använda BDD - behaviour driven development - så långt det går. Detta innebär att vi skriver tester som definierar programvarans egenskaper före vi påbörjar det "riktiga" programmeringsarbetet. Därefter skrivs kod med syftet att få testerna att gå igenom. På detta sätt testas kodbasen i takt med att den färdigställs.

## Licens

Då kodbasen är tänkt att publiceras för allmänheten så är valet av licens viktigt. Målet är att välja en lättförståelig licens som ger utvecklare, i de här fallet oss, credit för nerlagt arbete.

I nuläget planerar vi att skriva det mesta av kodbasen själva. Vi behöver alltså just nu inte beakta licenser för eventuella externa bibliotek. Det kan dock komma att ändras.

Vi valde tillslut att granska tre licenser; GPL V3, MIT och Apache.

På vår kravspecifikation för bra licens står två saker;

- Enkel att sätta sig in i, vi är trots allt inga jurister
- Ger oss som utvecklare s.k credit för nerlagt jobb. Kodbasen bör te.x tillhandaha en LICENSE-fil med våra namn i

Vi kände redan från början att GPL-licensen var lite för extrem. Att tvinga användaren att publicera tilliggande kod kan ställa till problem (för tredje part). Vilket kan resultera i att användaren väljer en annan kodbas.

Kvar har vi MIT- och Apache-licensen. Dessa två licenser är relativt liberala och tvingar inte användaren till några extrema anpassningar.

Valet föll slutligen på MIT då licensen i fråga är flera gånger kortare än Apache (~ 160 ord v.s 4 sidor). Den kompakta texten och enkelheten är till stor fördel för oss OSS-utvecklare som inte nödvändigtvis har den juridiska kompetensen som krävs för att förstå innebörden av alla detaljer.

Enligt den researchen som gjordes så täcker Apache-licens fler rättighetspunkter (vad som får och inte får göras). Dessa punkter försäkrar båda parter (utvecklare och tredje part) i en eventuell juridisk tvist, vilket är viktigt om kodbasen ska användas av större organisationer och projekt. Vi är dock 5 studenter och nöjer oss således med:

*- Permission is hereby granted, free of charge [...] to use, copy, modify, merge, publish, distribute, sublicense.*

## Projektets milstolpar och sprinters - preliminärt

Fas*	Sprinter	Syfte	Planerat start- och slutdatum	Tidsåtgång
Intro	S0	<ul style="list-style-type: none"><li>• Fastställa projektets syfte</li><li>• Fastställa projektets metod</li><li>• Planeringsrapport färdigställd</li><li>• Motivering av licens</li><li>• Kravspec påbörjad</li></ul>	2012-03-2012-03-23	7 dagar
XX	S1	<ul style="list-style-type: none"><li>• Fungerande GIT-repo</li></ul>	2012-03-23	4 dagar

		<ul style="list-style-type: none"> <li>· Färdigställa beskrivning av projekt och projektplanering</li> <li>· Kravspec fastställd</li> <li>· Första utkast för Haiku-algoritm</li> </ul>	2012-03-27	
XX	S2	<ul style="list-style-type: none"> <li>· Enkel Haiku-algoritm som skrivs i Java med Ja/Nej-resultat-funktion</li> </ul>	2012-03-27 2012-03-30	5 dagar
XX	S3	<ul style="list-style-type: none"> <li>· Total beskrivning av aktiviteter i varje sprinter fastställda à sprint-planing</li> <li>· Testfall och testplan klar</li> <li>· Fungerande Haiku-algoritm</li> </ul>	2012-03-30 2012-04-06	8 dagar
XX	S4	<ul style="list-style-type: none"> <li>· Möjlighet att tweeta från appen</li> <li>· I mån om tid: Haiku-statistik på Twitter för att jämföra användare, samt kontroll av Haiku-sms</li> </ul>	2012-04-06 2012-04-13	7 dagar
XX	S5	<ul style="list-style-type: none"> <li>· Testfall för större delen av systemet (cov:+50%)</li> <li>· Inga fler features, endast buggfix</li> </ul>	2012-04-13 2012-04-20	7 dagar
XX	S6	<ul style="list-style-type: none"> <li>· Källkodspaket, installerbart paket (apk),</li> <li>· Användarmanual, utvecklarmanual</li> <li>· Beskrivning av 3PP</li> </ul>	2012-04-20 2012-04-27	7 dagar
XX	S7	<ul style="list-style-type: none"> <li>· Projektinlämning / Release?</li> </ul>	Läsvecka 8	

\*Fasnamn är ännu inte fastställda.