

**FRAMSIDA**

# ABSTRACT & SUMMERING

## Abstract

- What the project does / is about
- Motivation
- Method and what was achieved
- Summary of main results
- Summary of main conclusions

**SHOULD summarise the whole report**

**- WRITTEN LAST**

**Give the reader enough information to decide whether to continue or not!**

# Innehållsförteckning

2.1 Gamification	7
<b>2.2 Topologi</b>	<b>11</b>
<b>2.3 Programspråk</b>	<b>12</b>
2.3.1 Python	12
<b>2.3.2 Javascript</b>	<b>12</b>
2.3.3 jQuery	13
<b>2.3.4 HTML</b>	<b>14</b>
<b>2.3.5 CSS</b>	<b>14</b>
<b>2.4 Ramverk:</b>	<b>15</b>
<b>2.4.1 Django</b>	<b>15</b>
<b>2.4.2 Django REST framework</b>	<b>15</b>
<b>2.5 Plugins:</b>	<b>16</b>
2.5.1 Bootstrap	16
2.5.2 Quill	16
2.5.3 Sweet Alert 2	16
2.5.4 jQuery Upload File	16

# KAPITEL 1 - Introduktion

## Inledning

*“Gamification, eller spelifiering som det också kallas, är att applicera samma sorts spelmekanik som speldesigners gör fast i en annan miljö än spel. Gamification är alltså inte att skapa spel utan en teknik som används för att motivera, engagera och förstärka positiva beteenden” [1].*

Detta arbete redogör hur man på ett företag genom gamification kan göra inläring intressantare och mer lekfullt. Arbetet går alltså ut på att utveckla en inlärningsportal till Telias kontor i Karlstad, där man genom en webbapplikation skall kunna använda sig av diverse olika moduler för inläring. Exempel på detta är olika quiz och onlinekurser. För att lyckas med detta måste vi utveckla en webbsida och ett API. För att sätta en extra prägel av Gamification på programmet skall poängtavlor skapas.

(istället för att skicka ut enskilda kurser skall man kunna ha dem samlade på ett stället)

(genom att ha en plattform för alla möjliga kurser/quiz kan man implementera ett poängsystem där man kan samla poäng genom att göra kurser/quiz)

(genom att samla poäng kan man kanske göra hela proceduren roligare/intressantare)

(användning av DB osv)

## Syfte

Inläring kan vara tråkigt, vilket kan leda till att anställd personal på ett företag inte väljer att lära sig bakgrunden med det arbete som faktiskt utgörs. I detta fall har Telia

svårt att få ut information till alla anställda, då det är vanligt med skiftarbete, vilket gör att arbetstiderna varierar kraftigt.

Syftet med detta projekt är att göra inläring mer intressant. Med hjälp av detta projekt skall all information vara lättåtkomlig, så att all personal får tillgång till samma information trots de varierande arbetstimmarna. Användningen av detta skall då hjälpa de anställda att få bättre förståelse för vad företaget gör, samt hur det fungerar. Detta kan även hjälpa cheferna att få en överblick på om de anställda förstår sig på vad dom bör göra.

För att göra detta möjligt skall man på webbsidan kunna ta kurser och quiz för att samla poäng, man skall även som administratör kunna lägga till nytt content. För detta måste vi även utveckla en sida där en administratör skall kunna bygga upp kurser och quiz (SÅ SOM HAN SJÄLV VILL).

## **Project Goal**

Målet med projektet är att utveckla ett så (färdigt/implementerbart) program som möjligt.

Där skall man ha:

Möjlighet att registrera användare som under tid kommer samla poäng genom att avklara kurser/quiz.

Möjlighet för en användare att ansöka om att göra om en kurs.

Möjlighet för en admin att skapa nya kurser/quiz.

Det skall vara lätt att skapa nya kurser/quiz

## **Summary of the context - why do this?**

Gamification är en intressant infallsvinkel på ett vanligt förekommande problem och kan implementeras inom många olika områden. Då det är väldigt brett användningsområde kan man använda sig av många olika tekniker/metoder. Lyckas man hitta en lösning på att göra inläring roligare kan det användas inom många olika områden/företag.

## **Dissertation layout**

### **Ch 1 - Introduction**

- This project.... (opening)
- Rational behind the project
- Project goal - I will do X
- A summary of the context - why do this?
- Dissertation layout

**Comments:**

- CH1s are often too short!
- Read some examples and decide
- What is the project?
- Why is it worth doing?
- Possible expected results?  
(these can be compared in the evaluation

# KAPITEL 2 - Bakgrund

**Agenda:**

**Gamification:**

**Programspråk:**

- Python
- Javascript
- JQuery
- Html
- Css

**Ramverk**

- Django
- Django rest framework
- Git

**Plugins:**

- Bootstrap
- Quill
- Sweet Alert 2
- jQuery fileupload.

**Assume the reader is a student with as much experience as you had before starting this project.**

**Ch 2 - Background**

- The “instant expert” chapter :)
- What do you (the writer) assume the reader does/ does not know?
- What does the reader need to know?
  - Definitions, terminology, topicality
- A reasonable set of references to pursue the topic further
- Existing systems ( Evaluation! )

Detta kapitel beskriver hur programmet är strukturerat, samt hur kommunikationen mellan de olika modulerna går till. Kapitlet innehåller även information om de verktyg som används under utvecklingen, samt en del information om dem.

## 2.1 Gamification

Detta avsnitt går igenom vad Gamification är samt hur det i detta projekt används. I avsnitt 2.1.1 förklaras vad gamification är samt tanken bakom det. I avsnitt 2.1.2 ges exempel på hur gamification kan användas, samt exempel på vart det redan finns. I avsnitt 2.1.3 kommer exempel på hur vi i detta projekt använt oss av Gamification i detta projekt.

### 2.1.1 Vad är gamification

Tanken bakom Gamification är att fånga användarens intresse, och genom ett fångat intresse få användaren att vilja fortsätta eller senare återvända. Det kan också vara tänkt att göra diverse områden roligare.

Momentet Gamification kan gå ut på att lägga till några ytterligare moment till redan existerande ideer. Exempel på detta kan vara så simpelt som Facebook[2] och dess "likes". Principen där är att man kan ladda upp en bild eller skriva en text som visas för vänner man på plattformen har. Gamification-momentet där är att andra användare på Facebook kan välja att trycka "like" eller kommentera en sådan bild eller text. Vissa kan se detta som ett tävlingsmoment där dem vill sträva för att få fler likes, andra kanske får ut en känsla av spänning i väntan på likes, medan vissa kanske inte bryr sig om det. Vad det dock gör är att lägga till ett extra moment på en sådan simpel sak som att lägga upp en bild, och genom det skapa intresse.

Metoden bakom Gamification har funnits väldigt länge, avklarade uppgifter resulterar i belöningar för att göra någonting mer intressant. Med dagens teknologi



har det dock kommit nya idéer om hur man kan implementera Gamification, samt vart det kan implementeras.

Det som är spännande med Gamification är att det kan implementeras inom många olika områden, det går även alltid att utöka och ändra efter behov.

Det finns även många olika sätt att implementera Gamification på, beroende på vilken effekt man vill åt kan vissa metoder fungera olika. Mer om detta tas upp i avsnitt 2.1.2.

## 2.1.2 Hur kan Gamification användas?

Gamification är fortfarande ett relativt nytt begrepp, men det är ett begrepp som många kanske hört talas om. Även om man ej hört talas om begreppet så är det säkerligen någonting näst intill alla personer stöter på dagligen.

Några exempel på detta är:

- “Gilla-markeringar” på Facebook, Instagram
- Kundklubbar (samla poäng för att få rabatter och individuella rabatter på produkter beroende på tidigare köp)
- Poängbaserade system

Många har insett att det kan vara lönsamt att implementera Gamification-lösningar på diverse olika plan, det finns därför numera många företag som specialiserar sig på Gamification. På grund av att Gamification är ganska nytt begrepp finns det många olika definitioner och många olika åsikter på vad det är, även fast det de senare åren gått mot ett mer samlat begrepp.

Exempel:

“Gamification, eller spelifiering som det ibland kallas, är att använda de mekanismer och aktivitetsflöden som finns i ett dataspel och överföra dem in i

ett annat digitalt system, eller att bygga en digital produkt kring något verkligt.”[3]

Denna definitionen kommer från ett företag vilket jobbar med att implementera motivationsteori och Gamification till organisationer för att motivera kunder och anställda, man kan då förstå att deras definition är som den lyder, att ta mekanismer och aktivitetsflöden från spelutveckling och överföra dem in i andra digitala system. Problemet med denna beskrivning är att vi inte anser att det behöver vara ett digitalt system man implementerar Gamification, utan det finns många andra metoder att använda sig av detta.

Även fast Gamification är ett relativt nytt begrepp kan man argumentera för att det är en metod som används många år. Ett begrepp inom Gamification är att man efter någon uppgift skall bli belönad, antingen med poäng eller genom troféer. Tanken med det är att användaren skall vilja komma tillbaka för att antingen få mer poäng eller samla fler troféer. En parallell man kan dra till den metoden är när man som ung kunde samla simmärken, genom att klara av diverse utmaningar blev man belönad med märken. Det var kanske inte nödvändigtvis var någonting man var tvungen att göra, men vissa valde att gå för märkena och på så vis samla troféer.

Andra paralleller man kan argumentera att dem finns är inom skolan, mer specifikt betygssystem och belöningar för “bästa student”. Betygssystemet krävs det att man inom ett ämne presterar bra, antingen genom att studera eller kunna ämnet bra sedan innan. Om en student presterar bra får denne ett högre betyg, vilket i detta fall kan ses som belöningen. En mer lämplig beskrivning på Gamification enligt författarna är då:

*“Gamification, eller spelifiering som det också kallas, är att applicera samma sorts spelmekanik som speldesigners gör fast i en annan miljö än spel. Gamification är alltså inte att skapa spel utan en teknik som används för att motivera, engagera och förstärka positiva beteenden” [1].*

### 2.1.3 Existerande system

Ett system som blev uppmärksammat och om man får säga, ganska lyckat, är MySkiStar[4] vilket är en lojalitetsplattform utvecklad av Ninetech[5]. MySkiStar tog idén att tillämpa Gamification på skidåkning, och löste detta genom att med hjälp av liftkort läsa ut data om de olika användarna. Användare kunde då samla poäng för diverse olika aktiviteter som flest åk och flest fallmeter, användarna kunde även genom plattformen läsa ut egen åkdata och se topplistor över alla användares statistik. Ninetech lyckades med hjälp av MySkiStar lägga till någonting extra, samt unikt till aktiviteten skidåkning. För den tävlingsinriktade öppnade detta för att tävla mot- samt jämföra sig med andra användare. För användaren som inte ville tävla mot andra kunde man genom sin egna statistik få ett mått på användarens egna data, och med hjälp av den sätta upp mål för nästkommande dagar eller dylikt.

Lojalitetsplattformar är en metod vilket företag använder flitigt. Lojalitetsplattformar går ut på att samla data om medlemmarna, och med hjälp av datan skräddarsy erbjudanden eller rabatter för medlemmarna. Det kan även gå ut på att medlemmarna samlar poäng genom inhandlade varor, och efter det beroende på hur mycket användaren samlat erbjuda diverse rabatter. Som namnet lyder är tanken med detta att genom ett medlemskap försöka göra användare mer lojala.

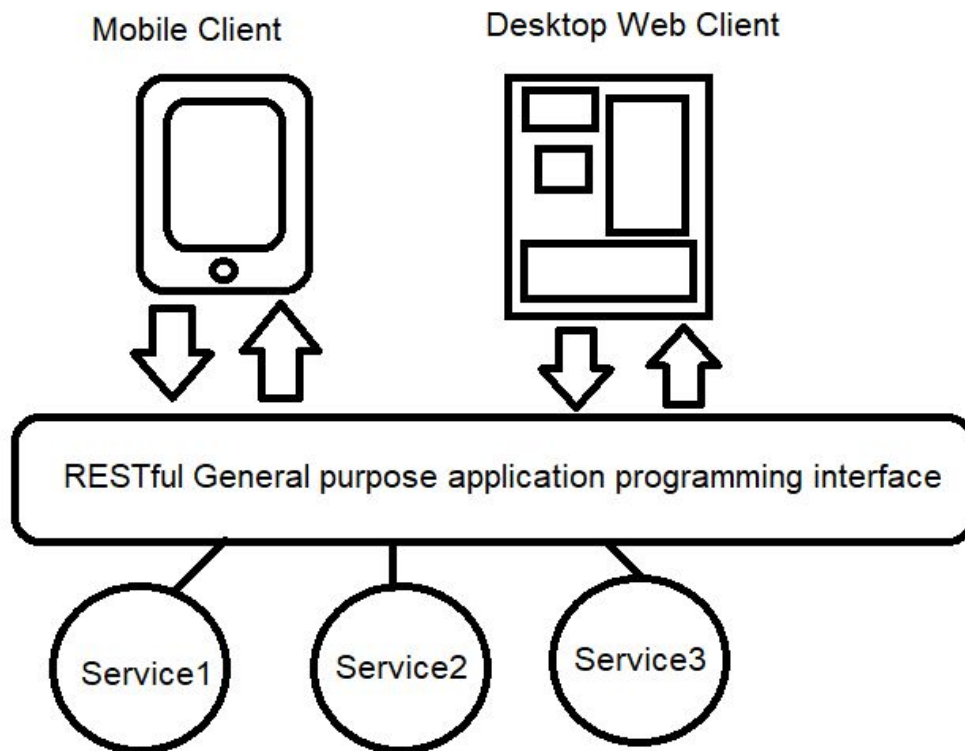
Ännu en implementation av Gamification som fick mycket uppmärksamhet är Pokemon GO[], vilket gick ut på att samla Pokemons(troféer). Man lyckades med detta med hjälp av mobiltelefonen och dess GPS-egenskaper.....

Vad Pokemon GO lyckades med var att göra "vandring" intressantare, och ge användare en anledning att gå ut och röra på sig.

Kahoot[6] är en plattform som implementerar Gamification för inläring. Kahoot är en spelbaserad plattform vilket riktar in sig mot skolor, studenter och företag. Man kan på plattformen skapa egna spel eller välja redan existerande spel. Kahoot har ett

smidigt system för att skapa olika quiz där man själv kan ställa in de olika frågorna, lägga till bilder samt ställa in om frågorna skall vara tidsbaserade eller ej med mera.

## 2.2 Topologi



I webblösningar används ofta tre stycken olika gränssnitt i utvecklingen. Det finns en klient, en server och ett API(Application Programming Interface). Klienten hämtar ner HTML, JavaScript och CSS kod från en server. Sedan kommunicerar webbklienten med API:et om att hämta och lagra specifik data. Ett API tar emot och skickar HTTP meddelanden med data, vilken data som ska skickas och vad som ska finnas tillgängligt kan bestämmas av de som bygger API:et. Anledningen till att man vill ha denna lösning istället för att prata direkt med servern är att ifall man vill koppla på olika sorters klienter kan man återanvända API:et och information som innan endast finns i webblösningen nu finns tillgänglig för t.ex. en mobilapp eller ett OS program. Det ökar även prestandan på webblösningen då servern endast har jobbet av att

skicka ut kod som klienten får köra och APllet endast behöver ta emot och skicka data.

## 2.3 Programspråk

I avsnitt 2.2 går vi igenom programspråken vi använt oss av samt hur vardera språk ser ut och fungerar.

### 2.3.1 Python

Python är ett högnivåspråk som har ett dynamiskt typsystem, dynamisk minnesallokering och objektorientering vilket gör det väldigt flexibelt. Det är i grunden ett skriptspråk vilket gör det smidigt för mindre lösningar, men det har även support för större lösningar via ramverk t.ex. Django. Python använder inga måsvingar utan baseras helt på indentering och har stor fokus på att det ska vara enkel och lättläslig kod. Python är ett av de mest populära språken just nu enligt Geeksforgeeks vilket också är ett bevis på dess användbarhet och skalbarhet. Den nyaste versionen är Python 3.7.1.

```
def article_view(request,id):
    if not request.user.is_authenticated:
        return display404(request)
    if request.method == "GET":
        template = loader.get_template('News/article.html')
        context = {'User' : request.user, 'ID' : id}
        return HttpResponse(template.render(context, request))
```

### 2.3.2 Javascript

Javascript är också ett högnivåspråk vilket kan karaktäriseras som dynamiskt, svagt typat och hanterar funktioner som första-klass-objekt. Detta betyder att ett objekt kan lagra data, exekvera funktioner, returneras etc. Javascript är en av de 3 kärnspråk som bygger upp internet tillsammans med HTML och CSS. Javascript tillåter hemsidor att bli mer interaktiva, vilket är något de flesta hemsidor använder idag. Javascript blir medskickat av servern och exekveras på klientens enhet men skapar

ändå känslan av att det är hemsidan som gör saker. Javascript påstås vara det mest populära programspråket i världen enligt Geeksforgeeks så även om det ibland kan anses vara osäkert så används det nästan överallt i världen idag.

```
var prevScrollpos = window.pageYOffset;
window.onscroll = function() {
var currentScrollPos = window.pageYOffset;
  if (prevScrollpos > currentScrollPos) {
    document.getElementById("navbar").style.top = "0";
  } else {
    document.getElementById("navbar").style.top = "-60px";
  }
  prevScrollpos = currentScrollPos;
}
```

### 2.3.3 jQuery

jQuery är ett JS(Javascript) bibliotek som jobbar mot att genom användningen av klassisk JS få ner mängden kod och på så sätt göra det till ännu mer av ett högnivåspråk. En av det mest användbara funktionerna med jQuery är att man kan använda sig av något som kallas AJAX (Asynchronous JavaScript and XML). AJAX låter klienten att själv kontakta antingen en webbserver eller ett API(Application Programming interface) för att utbyta information dynamiskt medans enheten fortfarande är på sidan. jQuery används av nästan 35% av de 1 000 000 mest besökta hemsidorna vilket gör jQuery till de populäraste javascript biblioteket.

```
$.ajax({
  type: "GET",
  url: "http://127.0.0.1:7000/v1/news/" + "{{ID}}" + '/',
  dataType: "json",
  success: function(data) {
    // alert(JSON.stringify(data));
    $('#id_headertext').text(data.title.toString());
    $('#id_shorttext').text(data.shortDescription.toString());
    $('#id_date').text("Created on: " + data.date.toString());
    articleDescription.container.innerHTML = data.description;
    // ...
  }
});
```

## 2.3.4 HTML

HTML (HyperText Markup Language) är ett märkspråk som används vid utveckling av webbsidor. Ett märkspråk är ett format för olika dokument där man kan definiera olika element som sedan visas. HTML förklarar och definierar alltså vad en webbsida har för innehåll genom HTML element. HTML element kan alltså vara allt från en klass(<div>), bild(<img>),hyperlänk(<a href="">) etc. Tillsammans med CSS och JavaScript så bildar de 3 språken det vi idag kallas webben. All HTML-kod ligger i en eller flera HTML-filer och det är dessa filer som skickas till en klient efter klienten har gjort förfrågan att besöka en specifik hemsida via sin webbläsare. På klientens enhet exekveras HTML-filerna och klienten kan beskåda hemsidan, detta ger klienten känslan av att vara "inne" på den önskade hemsidan.

```
<!-- Main -->
<section id="main">
  <!-- Thumbnails -->
  <section class="thumbnails">
    <div>
      <a href="{% url 'news' %}">
        
        <h3>News And Articles</h3>
      </a>
    </div>
```

## 2.3.5 CSS

CSS (Cascading Style Sheets) är ett språk som bestämmer hur ett dokument ska presenteras genom att sätta olika egenskaper. t.ex. som i bilden nedan sätter vi bland annat egenskaperna *position*, *width*, *border-color* etc. HTML och CSS går väldigt bra ihop eftersom HTML beskriver vad en webbsida innehåller medans CSS ändrar hur webbsidans innehåll ser ut. En välutvecklad webbsida har inslag av båda språken.

```
.navbar-expand-lg{
  background-color: transparent;
  border-color: rgba(10, 9, 9, 0.8);
  position: fixed;
  top: 0;
  width: 100%;
  padding: 0px 0px;
  text-decoration: None;
  z-index: 999;
}
```

## 2.4 Ramverk:

I avsnitt 2.3 går vi igenom de ramverk som vi använt för att underlätta skapande processen av vår webbservice.

### 2.4.1 Django

Django är ett open-source högnivå Python ramverk som tar bort mycket av de onödigt krångliga stegen med webbutveckling men samtidigt ger utvecklaren möjlighet att själv designa sin webb backend. Django är byggt på MVC (model, view, controller) modellen där relationsdatabasen som bifogas är "Model", webmallsystemet fungerar som "View" och URL-kontrollern som "Controller".

Ramverket kommer med saker som inloggning, sessions, admin sida, mappstruktur och automatisk synk av objekt med databasen. Den nyaste versionen är Django 2.1.5.

### 2.4.2 Django REST framework

Är ett ramverk för att bygga ett API(Application Programming Interface) inom Django ramverket som följer REST standarden. REST(Representational State Transfer) innebär att API:et tar emot HTTP requests som GET,PUT,POST och DELETE och behandlar datan korrekt. Ett API innebär att en komponent har ett par funktioner som kan nås av andra enheter. Detta behöver inte vara alla funktioner utan kan endast vara de funktioner som ägarna vill ska vara tillgängligt för världen.



## 2.5 Plugins:

I avsnitt 2.5 går vi igenom de plugins som vi använt för att underlätta skapande processen av vår webbservice.

### 2.5.1 Bootstrap

Bootstrap är ett open-source, frontend web-ramverk som fokuserar på att göra utveckling av informativa sidor enklare. Bootstrap innehåller HTML och CSS-baserade designmallar för olika interface-komponenter som exempelvis knappar, layouts.

### 2.5.2 Quill

Quill är en open source plugin skriven i Javascript som tillåter läsning och skrivning i rich text boxes. Det finns alltså support för punktlister, bold, kursiv och understruken text. Det har även support för att lägga in tweets, videos och bilder. [why quill]

### 2.5.3 Sweet Alert 2

Sweet Alert 2 är en open source plugin skriven i javascript som tillåter skapandet av enkla och snygga notifikationer. SA2 har support för i princip alla moderna webbläsare och får användas av vem som helst, hur som helst. [sweet alert 2]

### 2.5.4 jQuery Upload File

jQuery Upload File är en open source plugin skriven i jQuery som förenklar filuppladdning hos klientsidan. Det har support för drag&drop, enkel fil, flera filer, filsäkerhet, sekventiell uppladdning etc. Det fungerar även för alla server plattformar såsom PHP, Python, Ruby, Java, Node.js, Go etc.

# Prototype/Experiment

## Ch 3 - Prototype / Experiment

- What did you do?
- Give enough information to allow someone else to continue the project
- Stages of the project
  - Time, structure wise - organisation!
- Choices of A over B - why?
- Summary - reader cross checks

I detta projekt använder vi oss av AJAX för att anropa API:et direkt från klienten för att hämta specifik data från databasen. Man kan göra detta genom att definiera metoder i webbservern också, men vi valde denna lösningen för att endast använda webbservern för att hämta definierade templates för hur de olika sidorna skall se ut,

och istället hämta specifik data med hjälp av AJAX. På grund av detta får vi mindre belastning på webbservern, vilket gör webbsidan mer effektiv.

## 2.1.4 Gamification i detta projekt

Detta projekt använder sig av Gamification för att försöka göra inläring roligare. Detta görs främst genom att lägga till ett tävlingsmoment likt det på MySkiStar. Anställda på Telia skall kunna samla poäng genom att göra diverse test relaterade till nyhetsartiklar eller kurser vilket en administratör lägger ut på webbsidan. Med hjälp av dessa poäng skapas på webbsidan olika poängtavlor för användarna. Månadens bästa, totalpoängledaren osv. Det är sedan tänkt att dessa poängtavlor skall visas upp på diverse skärmar runt kontoret.

För att göra testmomentet mer intressant implementerades diverse olika funktioner för att testa de anställda på, metoderna som implementerades i detta projekt kan liknas med metoder som Kahoot använder sig av, detta för att likt Kahoot försöka göra inlärandet intressantare.

Detta gjordes för att de anställda skall få en samlad plattform för inläring, där inläringen sker på ett mer lekfullt sätt. Detta hjälper även företagets chefer att få en överblick på kunskapsnivån av de anställda. Med hjälp av detta kan då slutsatser tas över vad de anställda kan, samt vad de anställda kan behöva vidareutbildas på.

# Kapitel 4 - Results & Evaluation

## Ch 4 - Results & Evaluation

- Comparison against what was planned
- Comparison against existing systems?
- Issues & problems
- Warnings / advice for future work

The second time round (hindsight!)

+ve / - ve

# Kapitel 5 - Conclusions

## Ch 5 - Conclusions

- Project evaluation
- project vs prototype/experiment
- what did you learn?
- What went well?
- What went less well?
- With hindsight?
- Future development?

## Comments:

- Sometimes tends to be a summary rather than project evaluation!
- Take up non-computing aspects
  - Writing a specification
  - Working in a company
  - Communication: formal / informal
- More personal comments?

# References & Appendices

## References

- [1] <https://www.inspirationcompany.se/gamification>
- [1] <https://www.python.org/about/>
- [3] <https://www.motification.se/gamification/>
- [4] <https://www.ninetech.com/kundcase/skistar-myskistar>
- [5] <https://www.ninetech.com/>
- [6] <https://kahoot.com/>

## Appendices

### lite om gamification (definition av gamification)

#### Definitioner

<https://www.growthengineering.co.uk/definition-of-gamification/>

#### Varför använda?

<https://www.sogeti.se/losningar/digitala-kanaler/gamification/>

#### lojalitetsplattform

<https://www.ninetech.com/kundcase/skistar-myskistar>