|  |
| --- |
| Högskolan Dalarna |
| Unit testing of project from GIK299 |
| GMI24H |

|  |
| --- |
| Mikael Olsson och Nathalie Wongkam  2022-05-01 |

# General walkthrough of process

In this section we will try to describe our process and how we, the purpose is to clarify before we provide the required STLC documentation. We decided to use AGILE. More on that later.

When writing the user stories, we tried to think of the basics first and then as we go through the cycles and iterate through we put harsher and harsher requirements and get more specific. The user stories are supposed to supply us with the requirements but user requirements don’t necessarily have to cover every single fine detail. Some tests are too specific or detailed to be put into a real user story. These are things I consider to be a programmer story. Some things a user would never think of or notice.

# User stories

When creating the user stories, we took the approach that we will implement tests and when we iterate, we will create harsher and harsher requirements based on the user stories.  
  
We initially began with the user story:   
I want a program that is able to create blog posts, creating a post should be user friendly and it should always be clear in which step of the process I am in creating the post. I also want to be able to see my old posts and the old posts should be saved to a file. I would also like the ability to search for a post.   
  
This story gave us the requirements.

* Create a blog post.
* Clarity when using
* List all blog posts.
* Search for blog posts.
* Save blog posts to file.

At this point we would then implement the tests and make sure everything works, the next step would be to iterate on this process and do it again.  
User story:

When I use the program I would never want to see it crash, if something comes up I would like to be notified. When the blogposts are saved it should be saved as json format so that I can use the information for other stuff, speaking of what format it should save it in, I would like it to always save to the same file and if the file doesn’t exist then it should create a new file. This program should be able to used by everyone and that means that everything should be clear and the UI should look nice so that it is easy to tell what is going on.  
  
This story gave us the requirements.

* Handles exceptions
* Converts input to json
* If file doesn’t exist it should create the file
* If the files exists it should read from it
* Nice and well-functioning UI

Now we will iterate again with another user story.

User story:

I would like it to be more clear when using the search function that I did not find a post.

# Test strategy

Diskutera hur och vad du kommit fram till.

Exempel: Genom att använda…

# Test plan

Vill du kommentera något i laborationen? Var den bra, för svår, för enkel och så vidare så skriv gärna med det också.

# Test cases

Diskutera hur och vad du kommit fram till.

Exempel: Genom att använda…

# Test report

Diskutera hur och vad du kommit fram till.

Exempel: Genom att använda…

# Diskussion

Diskutera hur och vad du kommit fram till.

Exempel: Genom att använda…