Project Methodology
Course Assignment Report

Planning **2**

Tools **2**

Communication: Slack 2

Version Control: Github / Gitkraken 2

Planning system: Wrike.com 3

Tasks: Trello.com 3

Gantt chart explanation **4**

Sections 4

Critical Path Timeline: 4

Planning 4

Resource Gathering/Prototype stage 4

Second stage planning 4

Production 4

First presentation/Feedback revisions 5

Testing 5

Bug-fixing 5

Polishing/Launch 5

Github Repo **5**

Planning

For this assignment we were tasked to make a gantt chart for a website design for the company “Lofthus frukt og saft”. I’ve previously worked as a project lead on a couple of projects for a web-design and programming company, but this is my first time using gantt charts, and I must say I only wished the company I worked for then used it.

My initial approach to the task was doing some internal brainstorming, scribbling down some notes as I had ideas on what parts would be needed to successfully complete the project. After some deliberation and some notes, I started adding a time approximation for each task I wrote down, so that I would be more easily able to add this to the gantt chart timeline.

Tools

Being involved in similar projects both on a professional basis, and on a hobby basis, the choice of software to be used for collaboration was pretty easy.

Communication: Slack

For communication between team-members I would choose *Slack.*
This is a program I’ve used several times before, and used properly, it’s not only a great tool for communication, but also a great tool for storing all the information that is communicated throughout the project. The project would get it’s own channel, and with Slacks awesome search function, everything could easily be accessed at a later date if ever needed.
Github would also be linked to the Slack channel, so the channel would be updated with git changes.

Version Control: Github / Gitkraken

For version control the obvious choice is github (even if Microsoft did buy them). Obviously the biggest, most known git service provider, but I’ve never had any troubles with them in the past. Their servers are quick and reliable, although it would not be free seeing as we would obviously need a private repo for the project.
Our team would use the Gitkraken client, a really fast and well designed client, that has a low price for enterprise solutions. I prefer this client over Github’s own desktop client due to the design and simplicity. Everything is easier to understand, even for people not used to version control.

Planning system: Wrike.com

I would use Wrike.com as the solution for our project control and gantt timeline.
It offers a cloud based solution for project planning, with easily available reports, gantt timelines, assignee delegation, tasks, subtasks, projects and so on.
After testing out different project management providers, this is definitely the best one I could find, for a project of any size.

Tasks: Trello.com

While I would use wrike as the main project management software, I would like to use Trello as more of a more detailed, quick way of organizing task.
While Wrike would be used for the “big picture” tasks, all the smaller subtasks involved in a task would be set up in Trello.
This is to make the wrike timeline clean and easy to understand, not cluttering it up with way too many detailed small subtasks such as individual image edits, smaller bugs that needs fixing etc. Having tons of personal experience with Trello from game development, I really do love this service.

Gantt chart explanation

Sections

The first thing I did in my gantt was to make a critical path, highlighting the main timeline, and the tasks all the team members would be involved in.
I then proceeded to make a section for the “entire team”, where all the tasks the entire team would need to be present for are listed.
Following are a section for each part of the team, such as frontend, backend, writer etc.

Critical Path Timeline:

Planning

The plan starts with a meeting with the client and the entire team, for brainstorming, and to hear all the ideas and thoughts the client has about the project. The following day is a meeting with just the team, discussing and detailing how to execute the project, and whatever ideas and thoughts each team-member might have.

Resource Gathering/Prototype stage

Then comes the prototyping stage. This is where the writer and photographer starts getting pictures and relevant info, and where the frontend members start sketching, wireframing and making a illustrator prototype for the client to see. Backend takes care of domain setup and some PHP for the contact form.

Second stage planning

After the prototyping stage comes another meeting with the client to present the prototype and get feedback from the client. The most important thing in this meeting is to make sure all the changes the client wants are noted down, and if not feasible, discussed with the client on a better approach. The following task is another meeting with the team to plan the actual execution of the project production.

Production

The production stage of the timeline is where the majority of the workload happens. The frontend dev team has the biggest chunk here, writing the actual HTML and CSS for the website, and binding all the assets from the writer and photographer together. In the start of this task the writer and photographer makes changes based on the feedback from the prototype presentation to the client. The writer also makes English translations for all content, so that the page can have an option for English speaking visitors.

First presentation/Feedback revisions

Following the production stage, there is a presentation meeting with the client, and the rest of the week goes to make changes based on the feedback from the client. The writer gets the responsibility of making sure that there are no copyright infringements etc at the end of the week.

Testing

After this task comes testing, firstly for mobile, desktop, then towards a gathered target audience. The tests with some select “civilians” are important for gaining information on what could be improved. The users should be from a wide demographic of age and technical prowess.

Bug-fixing

Next comes bug-fixing where all the bugs discovered in testing should be fixed. Any improvement based on testing feedback should be implemented during this time. After this task the website should be a fluid bug-free experience.

Polishing/Launch

The last task before launch is final polishing. This is where the entire team goes through the page, looking for any smaller improvements that can be done. The webpage is tested with load-time tools, to see if load-time can be improved. Image-sizes and compression is checked and fixed if needed. SEO is also improved, making sure all headings, keywords and metadata is optimized.

Once this is done the product is presented to the client and launched.

Github Repo

https://github.com/TomThorsen/Lofthus
User “MJPhillip” was invited as a collaborator.