

# Bytecraft\_

## Project Proposal - Graph-based task management

CS-C2130 | Software Project

### Introduction

We are a new IT-consulting company on the Helsinki/Uusimaa scene. Despite the company being new, we have centuries of experience as Software Developers. Well, probably not centuries, but a lot, gathered in prestigious TOP10 consulting companies in Finland. But we hadn't yet met a company which puts the developer quality experience first, so therefore there was a need for Bytecraft - for Software Craftsmanship (<https://manifesto.softwarecraftsmanship.org/>) as the core value of our company.

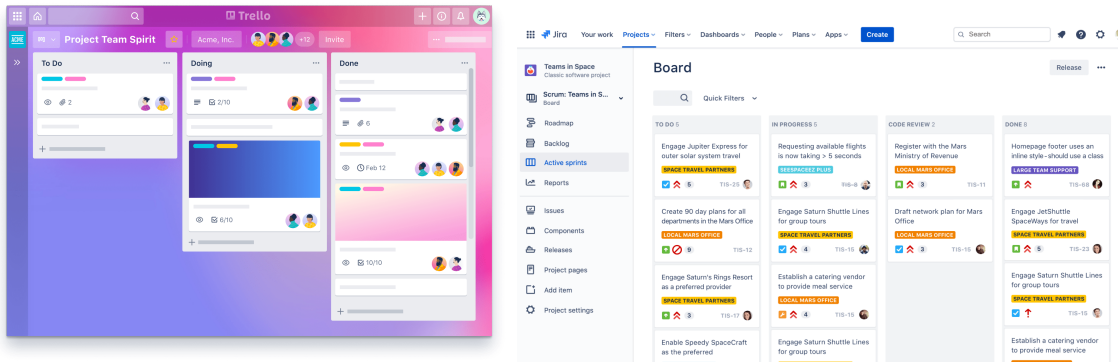
You might know us from last year, when we participated with a talented group of students and **our group received the quality award**. Some of the feedback from last year's students about Bytecraft were:

- *Uplifting spirit and lots of useful feedback*
- *Bytecraft's software craftsmanship -ideology and professional community were seen in the project loud and clear*
- *Course was overall positive. Especially the concrete support in form of code reviews and other help were important for the project's success*
- *Excellent! As cooperation partner and teacher/mentor, we received support always when needed and it was easy to communicate between the team and the client*

We aim to provide same kind of support for this year's group also and once again coach how to build quality software.

So now to the topic at hand. A lot of times at Bytecraft, we need to manage what to do next in our projects, while communicating to customers on time estimates and progress reports. Most existing project task management systems tackle these relatively well. The two most common systems currently in use are Jira and Trello, which show the tasks as several lists. This simple model works relatively well, but it **fails to visualize the dependencies** between multiple tasks. In

software development, it's usually required for technical reasons to implement tasks in a certain order, so visualizing these dependencies would help prioritizing tasks.



The competition - columns of tasks.

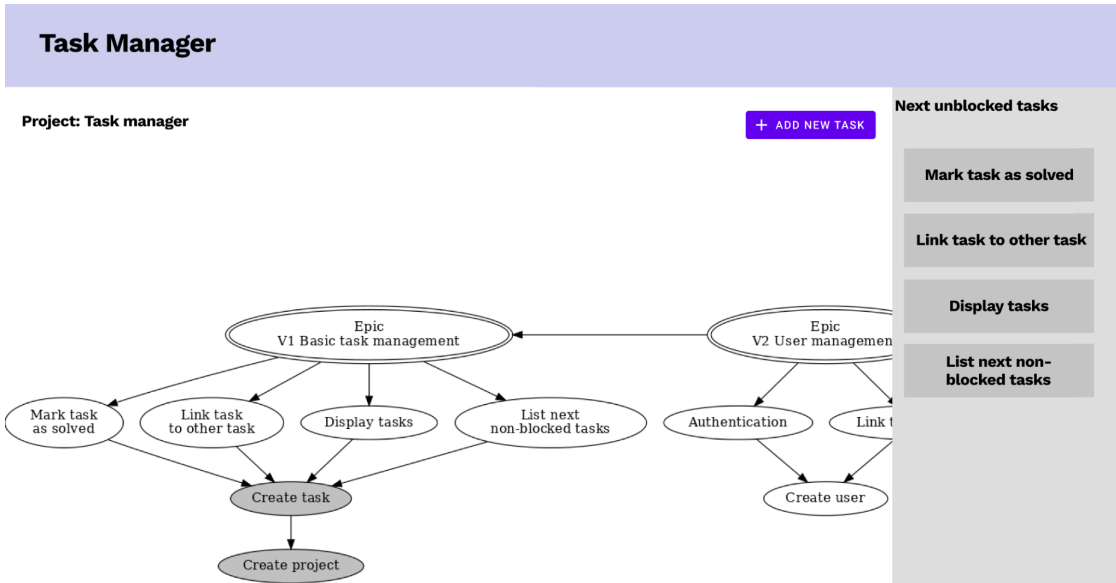
## Project - Graph-based task management

This project aims to solve that problem. Instead of relying on columns for visualization, the solution should render the task lists as a graph, with each node representing a single task. The edges between tasks signify various relations within the graph.

Standard operations for a project management software include:

- Create and assign tasks
- Change the status of tasks (new, in progress, done etc.)
- Add links to other tasks (task blocked by x, task is related to y etc.)
- Changing the order of tasks
- Add various things to tasks (labels, comments, ...)

This project is not any different. Aside from the standard features, the primary target for this project is **good usability and visualizing dependencies between tasks**, as current solutions hide dependencies pretty well. The secondary target is **automatically ordering tasks** based on which tasks provide the most value, whether that would be by completing features or unblocking other tasks. The exact technological choices, user interfaces and so on are left for the student team to decide. The most important feature here is **visual presentation**, as most other needs are already solved by existing solutions.



Rough draft of the project. The end result preferably looks better than this.

## Technologies

You can freely choose the best technology that suits the needs of the project. For example in the backend some technologies could be:

- Kotlin/Scala/Clojure (or any JVM-language)
- Python
- Node.js with JS or Typescript
- Go
- Rust

The UI should be accessible through any modern browser, which means that the frontend should be written with eg. React or Vue. Multiple pre-existing Javascript-based graph viewers / editors exist, and those should be evaluated and leveraged for this project.

We recommend Git for version control. The backend could be run on a cloud platform, for example Heroku. The group can freely choose how to run the app, it can mean for example ranging from Docker containers to serverless backend functions.

The technical support we can provide will be better for some technologies compared to others, as we aren't experts in everything.

## Requirements for the students

We challenge you to take initiative in designing and realizing a solution to our presented problem. We do not expect you to have earlier experience with the technologies, but choosing a familiar technology within the group could help for a better end result. Two attributes to highlight for the project are:

- Good usability and visualization is a key end result for the product
- Willingness to learn more on writing good software

It would be nice to have some existing experience with graph-based data, but that is not strictly required for this project. There are some non-trivial corners in the project, like managing persistence for graphs, but most of the project shouldn't be too hard.

## Legal issues

The results are published under open source license MIT. Signing the non-disclosure agreement (NDA) included in the Aalto's contract template is required. NDA is solely required for convenience reasons to be able to provide company premises as working space for students. The resulting code will be under an open-source license.

## Client

Following representatives from Bytecraft are available:

- Jaakko Hannikainen, Product Owner, Technical support (*jaakko.hannikainen@bytecraft.fi*)
- Antti Ahonen, Technical support (*antti.ahonen@bytecraft.fi*)

Also other additional technical support, depending on the chosen technologies.

Bytecraft office is located at Pasila, Helsinki. The railway station is right next to it. Depending on the Covid-19 situation, some meetings or learning sessions could be arranged there.

Bytecraft is a quite young company, but as earlier stated, we have a lot of experience in IT-consulting. We are also firm practitioners of clean code, clean architecture and all that jazz. Every detail of software building blocks matter: from a line of code to compositions of multiple services. Automating is a crucial keyword in achieving all this, be it automating the tests, build, pipelines or even code generation.

But we aren't just about code, we can also provide you with some fun events for team building. We've heard that pizza & beer is popular, maybe something on those lines (if Covid allows).

We are also highly interested in sharing the Software Craftsmanship -skills further in the future through possible apprenticeship or junior developer roles. Check more from:

<https://www.bytecraft.fi/> (sorry, Finnish only atm)