Leadership, Team dynamics
& Establishing communication channels

Future Factory – ZZPP0920

Aarne Salmi – AB8596,

T-03 Gang De Farine 2023

est. 2023

GANG DE FARINE

Sisältö

1 Oh joy, it’s Future Factory time!

When the first thing course instructor says is to stay calm and everything will be okay, you’ll know it’s going to be bad times. The purpose of this guide is to share basics of establishing communication channels, handling team dynamics and communication.

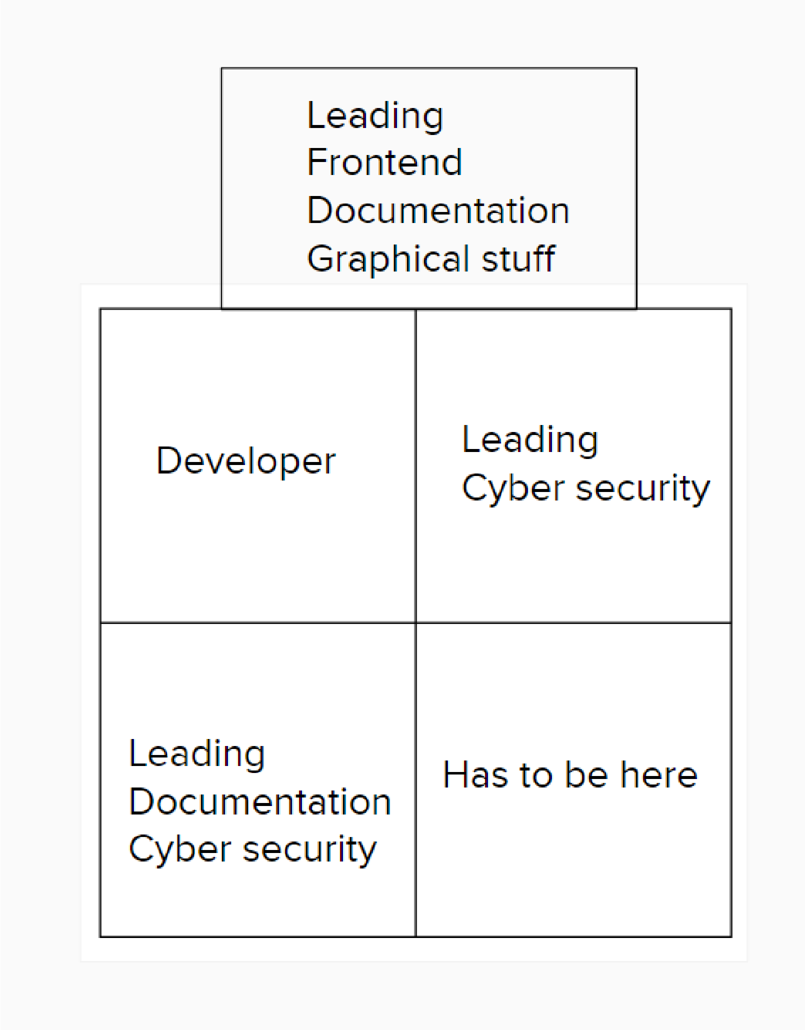
This guide is written by Aarne Salmi, and it is entirely based on my own opinions and working experience as a director, head of department, operative handyman and currently enthusiastic first year ICT engineer student in JAMK. I would like to note that while leadership and accountability is relatively universal, team dynamics and communication may vary so you must take that in account when planning your own team dynamics.

2 Team Dynamics

2.1 Who are these people?

You are being randomly allocated to teams based on your own interests that you completed on your survey. Let’s assume there are 4 randomly selected strangers in your team aside from you. The first thing that should be happening within the team is that each of you get to know each other. All of you are in this situation mostly because your degree depends on it and you are about to be thrown into wolves.

After everybody takes their turn to introduce themselves and what are they good at the team might look something like this:



Leading
Frontend
Documentation
Graphical stuff

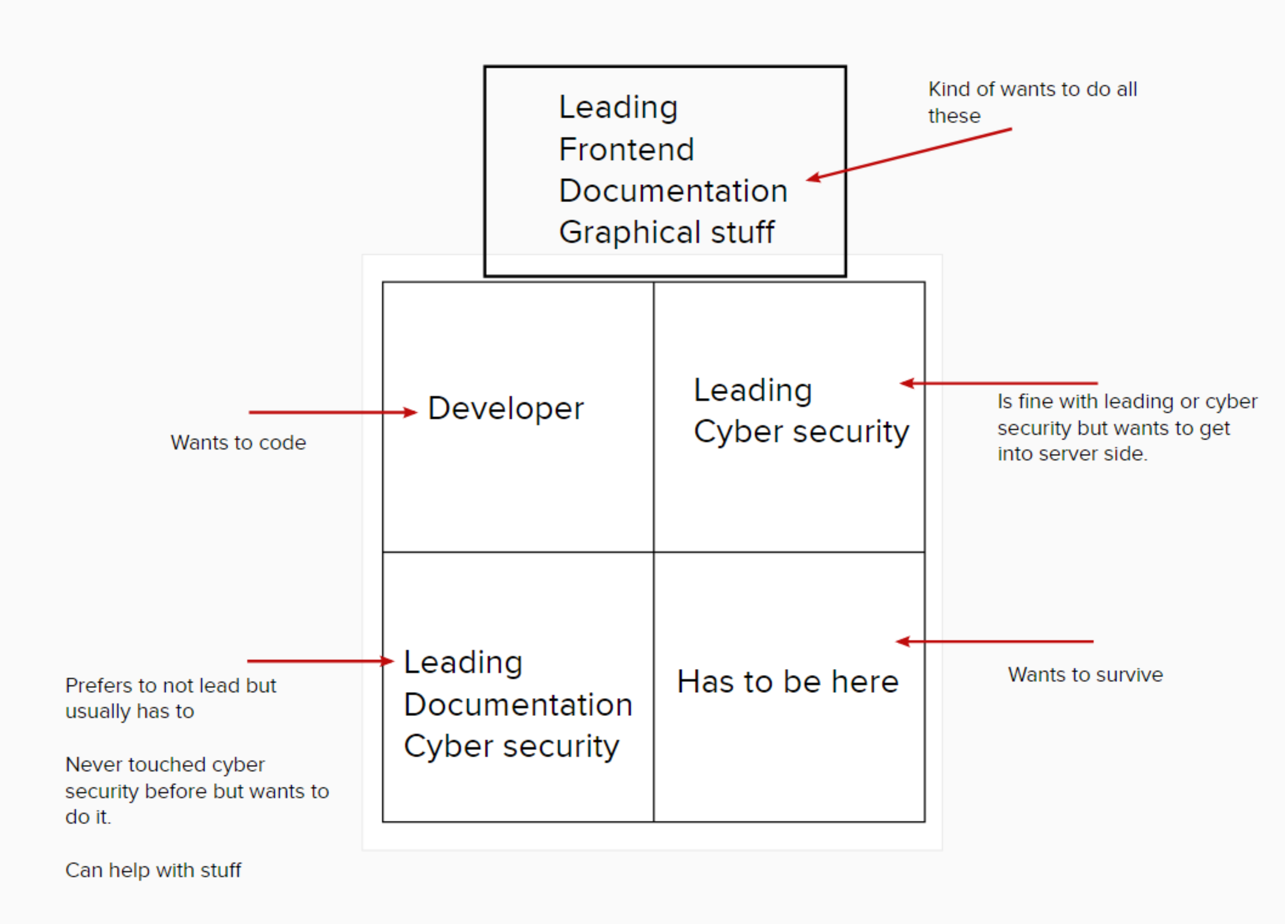
Developer

Leading
Cyber security

Leading
Documentation
Cyber security

Has to be here

Great, we now have a general idea about what's going on. Now it is our turn to ask what we actually want to do during this course and how motivated we are to do it. This is a very important question because for example most experienced person in cyber security does not necessarily have to be the head of cyber security. Giving the position to somebody who really wants to do it and get more experience during the course is valuable for increasing motivation and reason to do things properly. Being a battle buddy for somebody in tasks is common thing to do and highly recommended. After some clarifications and digging deeper the team dynamics may looks like this:


Leading
Frontend
Documentation
Graphical stuff
Kind of wants to do all these
Wants to code
Developer
Leading
Cyber security
Is fine with leading or cyber security but wants to get into server side.
Prefers to not lead but usually has to
Never touched cyber security before but wants to do it.
Can help with stuff
Leading
Documentation
Cyber security
Has to be here
Wants to survive

General motivation towards the course being a solid 2.5 out of 5 because it's a messy start, teams don't know each other, and they are not even aware what are they going to do.

Having clearly allocated roles and responsibilities in a small software development team is essential for several reasons:

1. When everyone knows what their responsibilities are, they can focus on their tasks and complete them to being with. This helps to avoid duplication of effort such as multiple people working on same task without being aware of it.

2. When team roles are distributed evenly, confusion, and unnecessary delays are minimized. This prevents situations such as having three frontend developers and not one of them taking action or developing 3 separate frontpages at the same time.

3. When everyone has a clear understanding of their role, they can communicate more effectively with other team members. This helps to ensure that everyone is on the same page and working towards the same goals. This is also very important for preventing creating battle of egos inside team.

4. Clearly defined roles and responsibilities help team members to take ownership of their work. This promotes a sense of accountability and ensures that everyone is responsible for the success of the project rather than one person “carrying” or ruining the team.

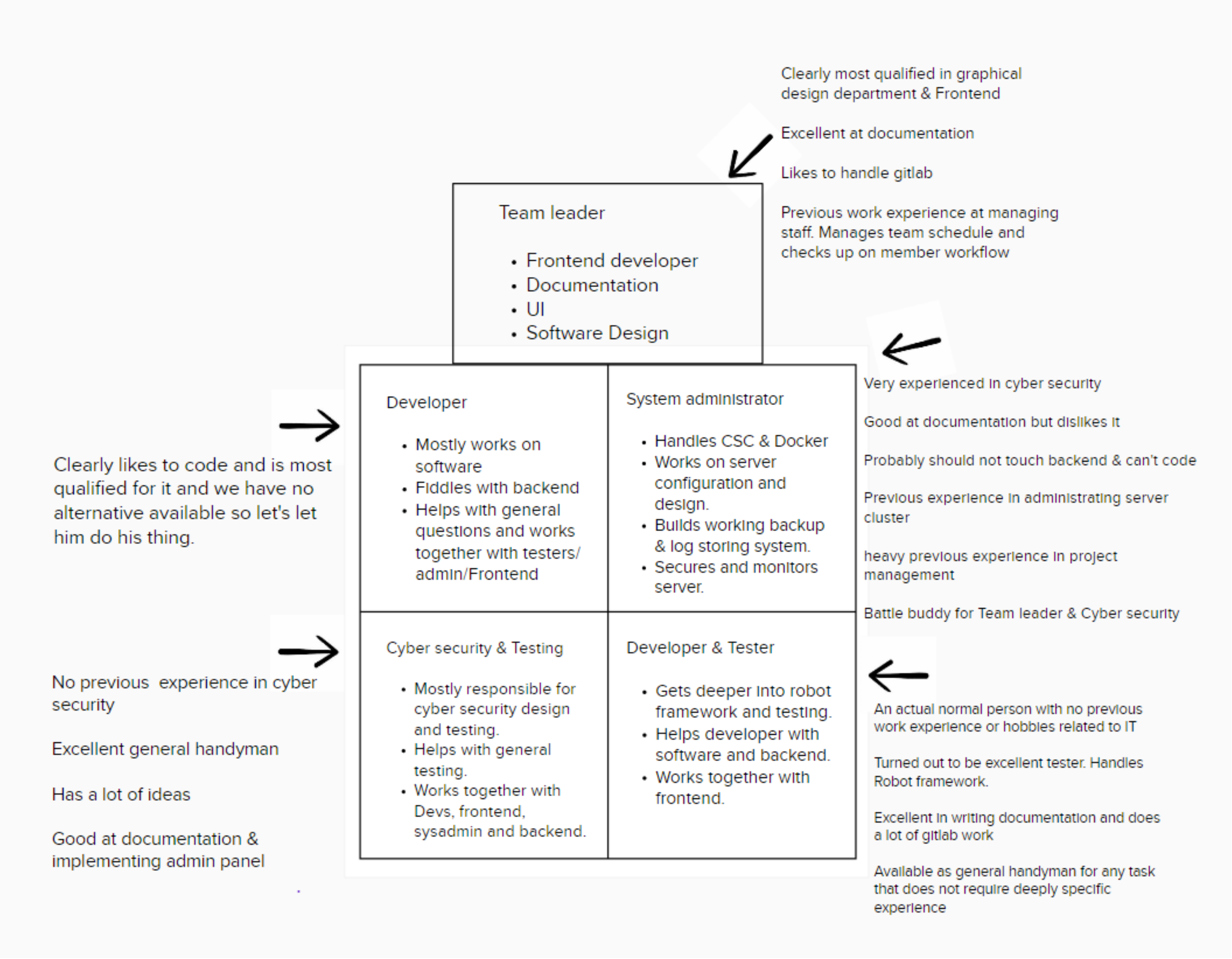
5. Reduced conflicts: When roles and responsibilities are not clearly defined, team members may step on each other's toes, leading to conflict and tension. Clear allocation of roles can help to minimize these conflicts.

6. Better time management. With clearly allocated roles and responsibilities, team members can manage their time more effectively.

In summary, heaving clearly defined and evenly distributed roles and responsibilities are extremely important for the team to function properly and ensure that members are willing and able to work towards the project.

It is also important to understand that projects and life within teams are dynamic. Once the role is generally agreed upon it can change as the project moves forward. It is generally rather difficult to plan ahead when you don’t know the topic or scope of the project. Most important part is clear communication and active participation within the team. Sharing or taking user stories or features outside your own allocated scope is perfectly fine as long as everyone are on the same page and agrees on it, there is a lot of handyman work in Future Factory.

As the course goes on and you'll learn more about the team, your team setup may look like something like this:



Team leader

- Frontend developer
- Documentation
- UI
- Software Design

Clearly most qualified in graphical design department & Frontend

Excellent at documentation

Likes to handle gitlab

Previous work experience at managing staff. Manages team schedule and checks up on member workflow

Developer

- Mostly works on software
- Fiddles with backend
- Helps with general questions and works together with testers/ admin/Frontend

Clearly likes to code and is most qualified for it and we have no alternative available so let's let him do his thing.

System administrator

- Handles CSC & Docker
- Works on server configuration and design.
- Builds working backup & log storing system.
- Secures and monitors server.

Very experienced in cyber security

Good at documentation but dislikes it

Probably should not touch backend & can't code

Previous experience in administrating server cluster

heavy previous experience in project management

Battle buddy for Team leader & Cyber security

Cyber security & Testing

- Mostly responsible for cyber security design and testing.
- Helps with general testing.
- Works together with Devs, frontend, sysadmin and backend.

No previous experience in cyber security

Excellent general handyman

Has a lot of ideas

Good at documentation & implementing admin panel

Developer & Tester

- Gets deeper into robot framework and testing.
- Helps developer with software and backend.
- Works together with frontend.

An actual normal person with no previous work experience or hobbies related to IT

Turned out to be excellent tester. Handles Robot framework.

Excellent in writing documentation and does a lot of gitlab work

Available as general handyman for any task that does not require deeply specific experience

With clear communication there is always work to do in Future Factory. The sheer amount of documentation and handyman work this course has quite wild. Also, if you don't know how to test things, write documentation and so on, just roll your sleeves and learn it. You are meant to sail in unknown waters, in fact, that is going to be the rest if your life as an engineer.

3 Setting up communications

3.1 What is required ?

Communication in Future Factory is relatively straightforward since you are allocated to a team of 5 to 8 people rather than being dropped into a mixture of 200 co-workers with different departments and inside politics. Here are the following necessities for smooth working environment:

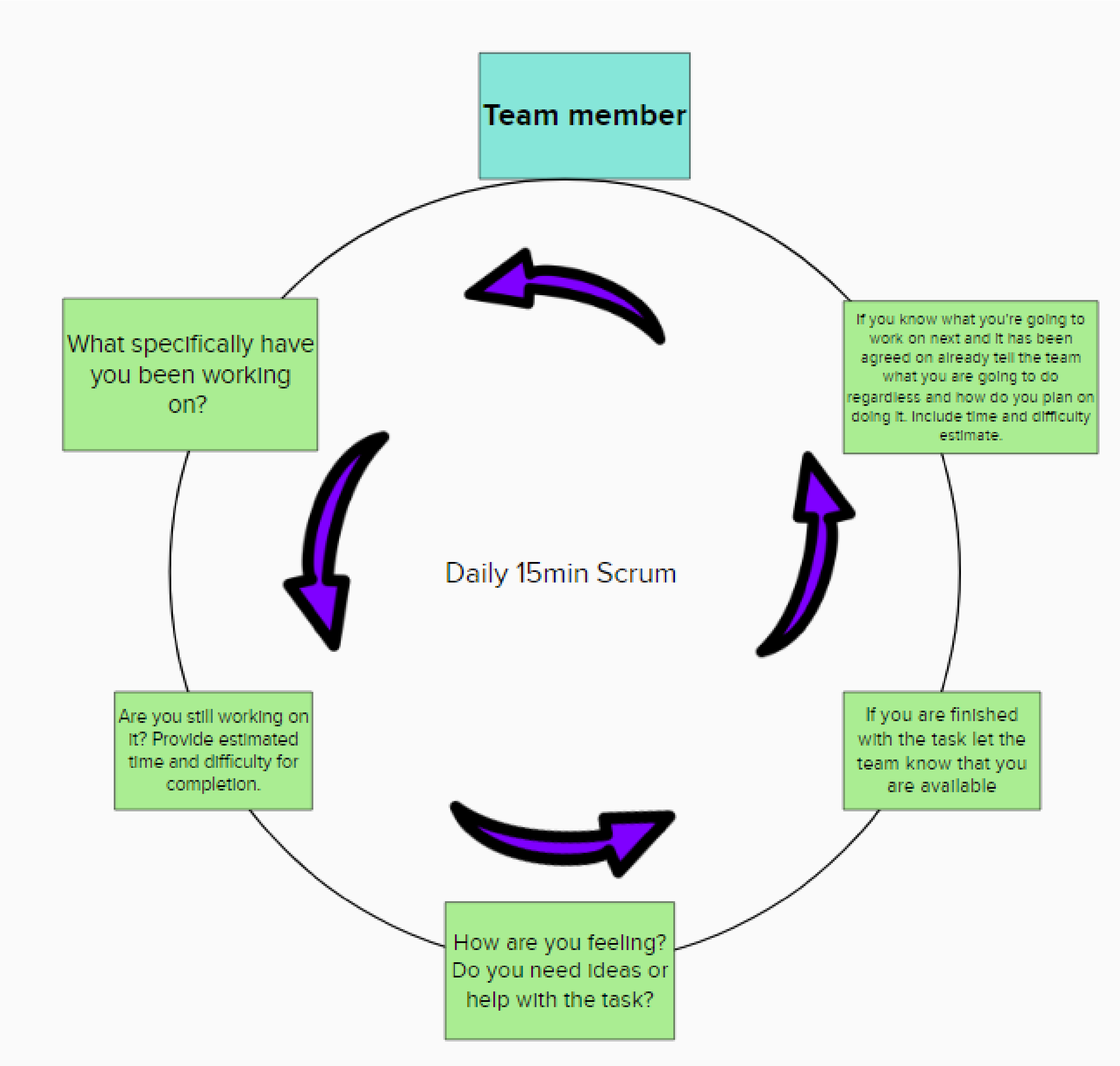
1. You need to have a software that everybody uses. As obvious as this sounds, you can't have a communication device that no one wants to use.
2. Your communication methods needs to be clear. It is highly recommended to not use messy logging systems or messengers that are mostly meant for two people such as WhatsApp.
3. Your communication means should be something that can be modified and accessed easily.
4. Streaming, file sharing and voice channel is a huge plus.

In T-03 case, we chose to use Discord to be our platform as it does everything mentioned above relatively well and it's easy to use. It is also worth noting that our enforced 15min daily scrums are not enough to keep track on the team and share ideas.

3.2 How to communicate ?

Grilling introverts can be great fun if you turn it into a sport. Future Factory has a bare minimum of 15 minutes daily scrum each Mon and Tue. If done correctly, it is nearly impossible to not know what each member is working on at all times if meetings are handled efficiently.

Team leaders are required to start the meeting at 09:00 sharp, assuming that your members are there in time or somehow message the rest of the team if they can't attend. There are many ways to pull this off but this chart is how I personally recommend doing it:



Team member

Daily 15min Scrum

What specifically have you been working on?

Are you still working on it? Provide estimated time and difficulty for completion.

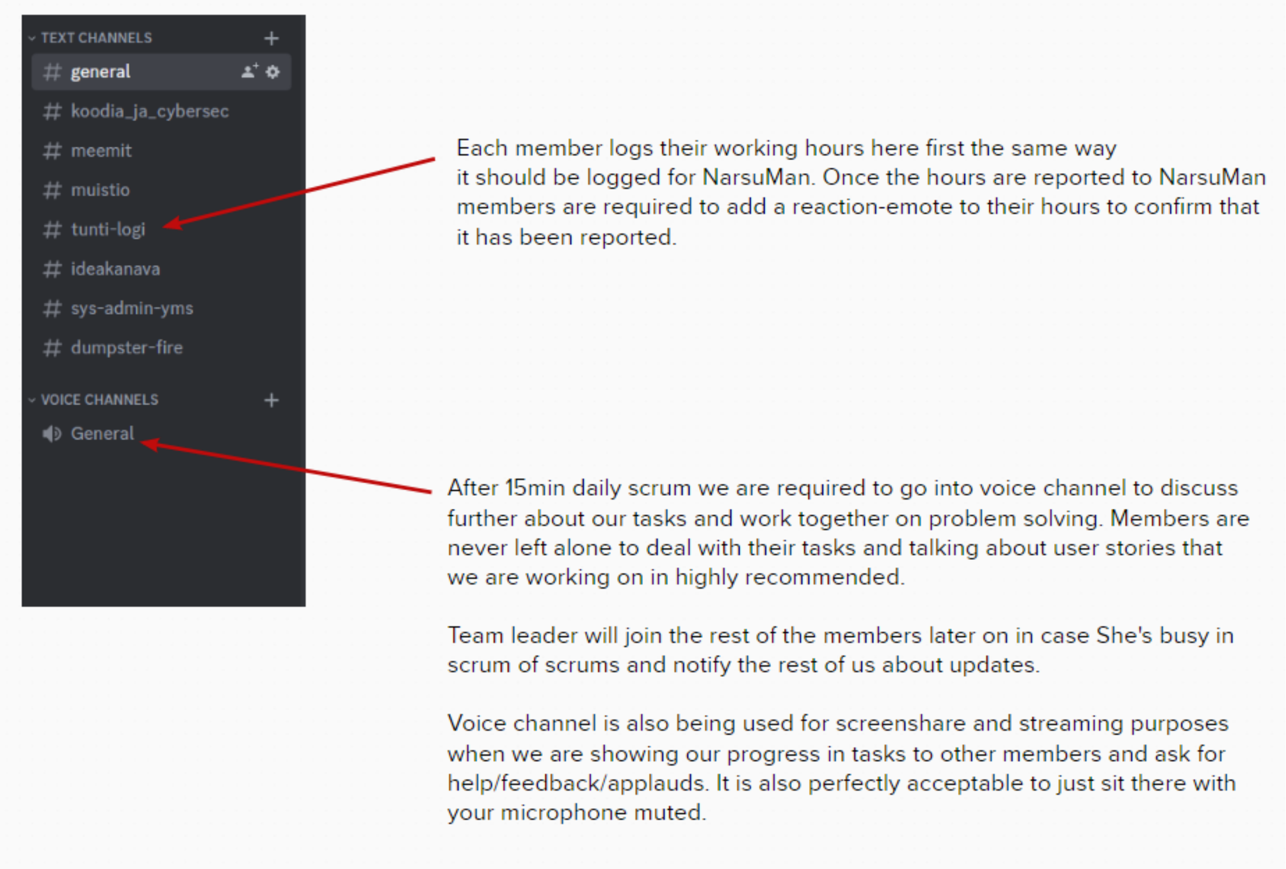
How are you feeling? Do you need ideas or help with the task?

If you are finished with the task let the team know that you are available

If you know what you're going to work on next and it has been agreed on already tell the team what you are going to do regardless and how do you plan on doing it. Include time and difficulty estimate.

Team leader will start the scrum by simply asking somebody how things are going and the discussion should then go through full circle by itself. Each team member is meant to give a full status on progress, whether they are feeling fine or low, are we going to be on time on so forth. Team leader is also required go through the whole circle. Once this is done properly and it is indeed tied to gitlabs roadmap and user stories, it is indeed hard to have two people working on same user story without being aware of each other.

After daily scrum is over and teams have agreed on today's tasks and it matches their gitlab roadmap and user stories, teams should move on to their designated communication platform regardless of what it is. If team does not have one, then you are doing things wrong. There are many styles when it comes to communication but not communicating is not part of the wide array of options you have available. Here is one example of how it can be done:



Each member logs their working hours here first the same way it should be logged for NarsuMan. Once the hours are reported to NarsuMan members are required to add a reaction-emote to their hours to confirm that it has been reported.

After 15min daily scrum we are required to go into voice channel to discuss further about our tasks and work together on problem solving. Members are never left alone to deal with their tasks and talking about user stories that we are working on in highly recommended.

Team leader will join the rest of the members later on in case She's busy in scrum of scrums and notify the rest of us about updates.

Voice channel is also being used for screenshare and streaming purposes when we are showing our progress in tasks to other members and ask for help/feedback/applauds. It is also perfectly acceptable to just sit there with your microphone muted.

This is not a standard, it’s just one way of doing it (And it works).

It is important for members to demonstrate their work to the team every once in a while. This is especially true for developers and system administrators since their updates may directly affect somebody’s work. Also, developers are never meant to do their work in secrecy and out of nowhere drop a fully functioning feature or even better, software. If teams have a solo-developer or system administrator it is extremely likely that your implementations are first of all, not going to work together and at some point someone’s working hours will get flushed down the toilet. This is one of those things that are not part of the “You get to choose” -list when it comes to developing software as a team.

Developers are encouraged to demo their work locally through screenshare and team members are encouraged to participate in looking at it and sharing their ideas. If whole team takes part in design, sharing ideas and helping each other through screen share through the whole Future Factory course your end product is going to be wildly different in every positive way instead of having just one person coding it in full secrecy. Same applies for system administrators. Administrators are meant to inform the entire team regarding what software and configurations are they about to deploy and what is their reasoning behind it. There is a high chance that somebody in your team can contribute useful ideas into it. Also, do not deploy security measures without mentioning about it for relatively obvious reasons 😊

Lastly, if you are not feeling well due to any reasons possible and there may be possible delays in your work, inform your team. No one needs to disclose why are they currently operating on low energy or disrupted attention levels but informing your team about it ensures that everybody are prepared for delays and you are very likely to be offered help if needed. Suffering alone in silence means you are undermining entire team.

4 Leadership & Troubleshooting

4.1.1 Everybody is a leader

There is leadership in all of us and everything we do in life requires leadership to some degree both in good and bad. In order to keep Future Factory field manual relatively short we will discuss about how leadership works in general. After that we will discuss about made up situations that could potentially happen to somebody.

4.1.2 Team Leader

As dramatic as it sounds, team leader is relatively similar to each individual member with the exception that team leaders are required to visit scrum of scrums. For team leaders in general I would highly encourage a compassionate and non-condescending approach for many reasons, but I would like to point out that this is a school project. Since team leaders are not in a position or even able, in any manner, to subject fellow team members into disciplinary actions, the tittle should be handled with grace.

Team leaders should be the one who actively check up on how things are progressing, keeps the entire team updated in their chosen manner and makes sure everyone has work to do. Although in some cases there may be weeks where there is very little to do in case team have advanced far. Working hours are also not meant to be allocated evenly and this is important to note. Sometimes developers need to work harder and sometimes server side must put in way more hours than other people.

Team leader needs to be brave enough to start uncomfortable discussions during daily scrum or outside that if required. If team has members that does not know what to do or are confused regarding how to help, it's team leaders' fault for not ensuring that members are on track and understand the task. In such case team leader need to fix this immediately. Team leader needs to be able to explain which user stories or features the team is participating in, in which order and so forth. User stories are also not being randomly selected, they need to be agreed upon by the whole team and then allocated to members by the team leader.

Leader's job in Future Factory is to be a messenger, ensure everybody has work to do, grant days off if needed and lead the communication proactively. In T-03 case, leader also ensures that gitlab and team's frontpage is up to date.

4.1.3 Leadership in members

While it is a very juicy thought indeed to just throw the team leader under the bus when going gets rough, unfortunately it does not quite work like that. Each member is required to have some degree of leadership in the project. What this practically means is that individual members need to be able to be accountable. What accountability means in Future Factory basically is clear communication. If a team member does not have work to do or does not know what to do, they have failed their team leader. Team leader shouldn't have the need to hunt his or her team members. Instead, team members need to communicate up and down the chains of command, which in our case is very lateral, efficiently and clearly.

In case there is nothing to do a member must inform the rest of the group that their task is finished, they are free to help and are unable to figure out what to do next. They may receive a day off from team leader if there is truly nothing much to work on but only after clearly informing.

By suffering in silence team member is failing the entire team. Each member must take the minimum level of responsibility and make themselves useful in some manner. Team members should also lead themselves during these dark times of Future Factory. There is no such thing as useless team member. Even if a team member had absolutely no technical skills whatsoever, there is a lot of things that needs to get done such as documenting, testing, requirements specifications and so forth. Future Factory is meant for dealing with the unknown and coming up with solutions. Very rarely in T-03 for example we knew precisely how to do things so what needed to be done is simply learn how to do it. Fire up that Indian youtube channel, Chat-GPT and google. Your answer is somewhere out there.

4.1.4 Made-up issues & example solutions

These issues are not related to any team or student in any shape or form. If they are relatively accurate, it is entirely out of sheer coincidence.

#1 We don't really know what each of us is working on.

Suggestion: Go through the scrum circle, if you still don't understand what other members are doing ask them to speak up more clearly and in plain manner so that they can be understood. If you don't know what team members are working on, your communication is broken and needs immediate fixing.

#2 I don't like some of the team members.

Suggestion: You don't need to like your team members, you need to get along and deliver the product to the customer. If there is battle of egos going on, then you have much larger issue and need to solve it either together with team leader and person involved or escalate it to course instructors for help.

#3 We have an expert in our team but his/her work is making our life harder out of complexity.

Suggestion: One of these two are very likely to be happening. 1.) Either your more advanced team member is simply horsing around and making intentionally your life harder than it needs to be by enforcing complexity on you. If this is the case, asking person to stop and addressing it publicly within your team should work. Or 2.) You have never asked your more advanced team member to explain clearly with screenshare and updates regarding what are they doing, why are they doing it and how does it work. This is entirely matter of communication and could be solved easily by simply addressing the topic.

#4 We have an expert that is carrying the whole course. Other members does not have much work to do.

Suggestion: Team leader need to take control of the situation and allocate workload and tasks appropriately. There is nearly unlimited amount of work to do in this course. Skilled people don’t get to have any extra freedom here and they are meant to follow same basic guidelines.

#**5** I’m not technical person, what should I do?

Suggestion: Documentation and gitlab work are least technical tasks to pick up. You could also learn a new skill and do that. Ultimately this is something that you should talk with your team leader or course instructors.

#6 None of us know how to connect database to frontend

Suggestion: Ask for help, fire up that youtube. You can’t just sit and hope that it goes away.

#7 Our team leader is inexperienced and isn’t doing too well.

Suggestion: Team members are supposed to support their leader. If you are way better at leading than your actual leader, which is possible, you are failing your leader by not helping him or her to execute the task correctly. There is no need for micromanaging but rather show the general direction and let them fly themselves. You could also ask instructors to help your own team leader.

#8 Our team dynamics are a bit off

Suggestion: It’s never too late to fix it. T-03 made role changes at S5. You need to voice your concerns.

#9 Our team skillset and level is VERY diverse

Suggestion: As it should be. It is your and team's job to figure out how to allocate tasks and work-load. Also, most advanced person does not need to do the task. They are excellent battle-buddies in case things get hard.

#10 We have members/leader that does not even want to do the course or work.

Suggestion: When you are unable to discipline or fire the employee, things like this can become very difficult situations to handle. I would suggest to immediately escalate this to course instructors and let them know how you are feeling about the situation.

Stay safe out there !