Code of Conduct – Team 11

Adam Szymaniak, Boaz Bakhuijzen, David Potskhishvili, Jakub Pietrzak,
Marios Kailis, Midas van Veen

1 INTRODUCTION

This Code of Conduct establishes the shared principles that guide our collaboration process throughout this object-oriented programming project. By answering questions like “Why?”, “How?”, “When?” and “Where?” we work together, we lay the foundations for a positive, inclusive, and productive team environment where everyone can get the most out of their first collaborative software engineering experience.

2 SHARED TEAM VALUES

Growth-oriented approach First and foremost, we treat the project as a learning opportunity. During the course, we have the opportunity to experiment with new programming concepts, knowing that our mistakes carry minimal consequences. We treat setbacks as a natural learning process. We contribute towards creating an environment where everyone can seek help from other team members when struggling to grasp challenging concepts.

Healthy work-life balance We understand that maintaining a healthy work-life balance is essential for our mental well-being and keeping our minds full of fresh ideas. That is why we aim to set the deadline for meeting the knock-out criteria on Fridays. By keeping the weekends free of mandatory involvement in the project, we allow everyone to enjoy their well-deserved time off. Normalising this rule discourages members from pushing their first merge request with Java code just before the hard deadline on Sunday evening, as it creates a risk of not getting enough approvals.

Transparency We keep our communication open and honest. We aim to keep everyone up to date with the project and involved in the decision-making process. Everyone should proactively provide updates on the issues encountered while working on their weekly objectives, allowing us to work on the solutions together, which keeps us productive at all times.

3 ASSIGNMENT DESCRIPTION

The assignment is here to show what happens in the real world projects. It shows and teaches us how to communicate during a project, and how to handle it.

Because of this we as a team need to: communicate, work together, plan and organize. Besides teamwork, it teaches us new concepts like git and web development. To pass this course we need to have good collaboration and understand the given rubric, and deliver on those criteria given in said rubric.

To go into more detail, we will make an app that automatically splits the costs of an event among a group. The parts of the project making this work can be split roughly into three parts: - Server side code - Client side code - Jakarta persistence models

The Server side code is the code that handles all requests from the clients that come from the Client side code. It handles interactions with the database, and connects clients together in events and remembers which person is which, and much more.

The Client side code sends mostly requests to the server, then the server responds with information that was asked for. After that the Client side code will show that information in a user-friendly graphical interface, which the client can then interact with. And of course the client will remember some information, such as recent viewed events.

The Jakarta persistence models are models that correspond with the database schema, such that the server can use these models to interact with the database correctly, thus making sure the database stays intact. These models also make it easier to send data around, they are somewhat comparable to java object, they even look like it. By doing this you make the code more readable, and you have a container to put data in and send around in the code.

Then the Teamwork part. We hold weekly meetings to share information between us, for example telling what went good or what we need to work on. Also, during these meetings we can make decisions about the project, ensuring we make them together. To collaborate outside the meetings we use multiple platforms, such as Mattermost and Gitlab.

4 AMBITION LEVEL

We are aiming for an 8 or higher for this course. Those who did not have those standards said that they will try and keep up with the rest, thus not trying to hold them back.

We think an 8 for a grade feels satisfactory, while not being too far-fetched, but still being higher than average grade. Also, we find it a good target to show that we learned something, instead of chasing only the basic requirements and not doing anything more than asked. So, hitting this goal will feel more as if we really understand how it works, and having the feeling that we can do more than before.

This will ask some time and effort, but it will reward us with more real life experience than a simple test, because we learn to work together as a team, instead of as an individual. Thus, improving not only our knowledge, but also our planning and teamwork skills along the way.

5 PLANNING

When it comes to the planning, we first schedule meeting and discuss all points in the project which are required to implement.

The extremely important thing is that everyone have the same and right level of understanding in terms of project structure and project’s frameworks. We always try to ask each other detailed questions to verify that everyone follows the same track. This is extremely important since the same level of right understanding helps us to make adequate goals and plan our strategy for the current week which sequentially allows everyone to finish their work on time.

Only after aforementioned stage we begin with actual planning on the meeting. We first decide how we should distribute the work between people since 6 people working on the same feature is really hard and unnecessary in this project. However, this does not mean that each person or subgroup only cares about its current work since again staying on the same track throughout the whole course helps making reasonable decisions in planning.

After distribution we analyze the dependencies between works. Since one subgroup might wait for the completed work of another subgroup, it is important to understand whether such dependencies might take a place in planning considering world realities and deadlines. After that we usually draw a schema with all roles' distribution and tasks assigned to each subgroup. Moreover, we always like to complement our planning with some visualizations.

Then we need to find out whether our planning is flexible. Something always can go wrong during the work week and we might adjust our planning due to the external circumstances. if our planning is not flexible enough, then some people might end up not finishing their work which is understandably not desired result.

We decided to keep it flexible when it comes to the final say in the final deliverable and submits. Why not keep it flexible when we have enough self-organised people in the team.

Also, we should not neglect the power of GitLab when it comes to the planning. Such powerful as issues, milestones and labels help team track all current problems, workflows and navigate in the whole project.

6 BEHAVIOUR

We treat quite equally each other in the team since the whole understanding of the project and its structure is a necessary condition for each team member. However, we see each other strengths and consider it when it comes to the important decisions in our team.

We decided that there won't be any specific punishment for being late apart from helping minute taker with notes from the current meeting. If person does not contribute enough to the project, then it is likely that person will not meet knockout criteria. This does not mean that we do not propose help to our team members, but there are some cases when even meaningful help is not "key" to the specific team member, unfortunately.

There are always some cases when escalating to external parties is a necessary action. Obviously, when the behavior of the team member is unreasonably disrespectful and other team members' requests are not considered by this specific team member, then we will likely approach TA first. If TA could not help us with this issue, then we are likely to approach academic counselors and course hosts.

7 COMMUNICATION

The primary communication channel should be the Mattermost server. We should keep there most of the discussions about the project and ongoing work. Every debate about essential functionalities and core concepts everyone should understand, as well as important organizational matters, should be held on Mattermost. It is also a place, where we can ask our TA for clarification if we have any questions regarding the course.

We can also communicate through the Discord server, however we should aim to keep discussions there off-topic. Discord may also be used if we want to hold a long conversation that may devolve to spam and is not essential for our understanding of the project or for quick questions, which do not shape the overall state of the project. Gitlab can be used for commenting and reviewing the code of others, and to give or request feedback. If someone wants to change something in a merge request, they should communicate it on Gitlab, providing a sensible explanation. Gitlab issues can also be used as a planning tool - unsolved problems and weekly division of work should be posted as Gitlab issues.

If someone cannot participate in a meeting, Discord may also be used for video/audio calls, assuring that hybrid or online meetings are possible.

Discussion regarding any additional meetings should be held openly on Mattermost to ensure that everyone has a chance to participate or suggest schedule adjustments.

Chairman should submit an agenda for the current week before the meeting with TA, so everyone has a chance to read it. During the meeting minute taker should clearly state when they will submit meeting notes, it should be in the evening of the following day at the latest.

8 COMMITMENT

Commitment of a chairman and a minute taker can be primarily assessed by the TA's formative feedback. Agenda, quality of a meeting and notes of a minute taker form evidence of their commitment. Their work can be commented and evaluated by the team members under their merge requests that cover organizational matters.

In order to ensure that everyone writes code of sufficient quality, we use code reviews and Gitlab comments to propose changes that would enhance the quality of code. Every merge request should gain at least three approvals before mergin - this additionally ensures that nobody commits poor code and everyone pays attention to details.

Individual knock-out criteria, along with the abovementioned constraints, enforce that each team member must be committed, contribute weekly, participate in the creative process and deliver work of sufficient quality. If we have objections regarding the work of somebody, we should first attempt to discuss with them what we think should be changed. The discussion should be held in a calm, polite, respectable manner, it should be based on reasoning and there should be no personal attacks. If that does not help, we should talk to the TA and together solve the problem.

9 DECISION MAKING

We agreed that most matters that have to do with the decision making process will be discussed and decided upon during the weekly meetings. During the meetings, we talk about what has already been done and what needs to be done. We also discuss any issues that have arisen and everyone offers their own ideas of how to solve the problems.

If someone does not agree with something we are discussing or doing, they are encouraged to talk about what they think is wrong and offer their own solution. We will then discuss the different solutions and decide on the best one mainly by majority vote. If someone insists that our solution is not correct and we cannot conclude a decision, then we can seek the help of our teaching assistant to give us some guidance.

Moreover, since the time we have in the meeting is limited we also agreed to discuss our ideas on Mattermost/Discord group or Gitlab issues. Then we all can vote for that idea using emojis or provide feedback in order to approve or reject that idea.

10 DEALING WITH CONFLICTS

It is important to distinguish between a disagreement regarding the project and a personal disagreement.

If team members have a disagreement regarding the project, then we can all have a discussion about it and decide something that is fair for both sides and also beneficial for the project.

If two members have a personal disagreement, not related to the project, then it is best to keep it outside of the team meetings and discussions. They should try solving it themselves.

If they are not able to solve it themselves and the rest of the team still senses some tension between them, then we can all have a talk about it, not in a weekly meeting but in a separate meeting. We can hear both sides and with no bias try to help both of the members solve their personal disagreement.

Now if the conflict escalates and it is no longer in the team's hands to find a solution, then we should definitely seek the help of our teaching assistant. With their help, we might be able to work something out but even if we do not, the teaching assistant should be informed about the situation.

11 THE CONSEQUENCES OF NOT KEEPING AGREEMENTS

We decided to have the following consequences for not keeping agreements: They will have to explain themselves to the rest of the group (including the TA) and apologise. Examples of not keeping agreements are:

- Being more than 30 minutes late or not showing up to a meeting
- Not finishing work or not completing a task in time
- Not responding to emails / messages or not responding to phone calls within a reasonable time frame

If a team member does something that merits a consequence but said action has not been previously discussed, then we will hold a separate (online) meeting to decide on the consequence we will impose on the offending team member.

Enforcing additional consequences might only lead to animosity among team members, which is not productive.

Instead of enforcing more consequences, we will focus our effort on helping team members keep agreements. An example technique we will use is asking team members for follow-up questions about their agreement. Questions like "how?", "why?" and "when?" force the team members to make an initial commitment by thinking about their agreement. This acts as a commitment device.

12 OUTSIDE COLLABORATION

The need for an additional meeting besides the mandatory TA meeting will be decided weekly in our primary communications channel, Mattermost.
If a group of at least 3 team members decide they do want to hold an additional meeting, it is strongly encouraged to do so on **thursday at 12:00**. This meeting *must* be accessible online, either via phone call or discord call.
Pairs of team members can meet whenever they like, but they must make it known in Mattermost when and where they are planning on meeting, just to give anyone who's interested in joining in the chance to do so.