

SER316 Software Enterprise: Team Icebreaker Fall 2022, Session B

Version: August 29, 2022

This team assignment is intended to help you get to know your team, and prepare for your first sprint. There is no submission it is all on a GitHub repo, in Slack or your groups repo.

I expect (and your team should expect) that all of you read all of the kickoff material documents and gone over Memoranda and the DeliverableX.md document and preferably the individual setup already. Do not be unprepared this will just waste everyones time! If I see based on documentation that you were unprepared and did not read the documents that will impact your grade.

1. Individual Introduction

In your slack group channel (`_gen`), introduce yourself to your team and get to know your teammates. Pin these messages to your channel.

Example intro:

1. Who are you?
2. Where do you live?
3. What are your hobbies?
4. When do you have time to work?
5. How experienced are you in Scrum?
6. How experienced are you in Git/GitHub?
7. Did you already go over the kickoff material?

Part I. Story time

This is supposed to be a little fun "get to know each other" exercise. Do not take it too seriously but you will be graded on doing it. It is more about fun and working a bit on Git as a team without having to be worried to mess the project repo up.

Create a **public** repository on GitHub with any reasonable name and add all your team members to this repository. Also add ser316asu as a collaborator.

Below you find a couple of German words (I know I am mean, you will probably have to look them up).

Raesfeld Flughafen Berlin Bratwurst Hauptstrasse Brot Klimaanlage Rammstein Porsche
BMW Biergarten Ratskeller Dirndel Lederhose Schloss wettbewerbsfaehig Mehlhase Gebaeck
Baeckerei Hund Doener Apfelkuchen Schloss Eltz

Step 1 Create a dictionary.txt file which will hold the German words and their translation in English if there is a translation for it.

Step 2) Your task is to create a (funny) story with the given German words. This story is supposed to use at least 10 of the German words. Your story can of course be in English and incorporate the words in any way you want. This is supposed to be fun and lightweight, it is not about having a perfect story and has nothing really to do with SE just that you use Git and hopefully have some fun with your team members.

To do so you work on your story.txt file (put it in you repo). Each of you add one sentence at a time to the story and commit your one senntence. You can commit directly on master or create branches for it to practice working with branches – might be a good idea (you can work asynchronously on this), if you get a conflict resolve it so the story still makes some sense or not ;-)).

Each sentence should use one of the 10 words (or more). So overall you will end up with roughly 10 sentences and each of you should write 2 of them (your 2 sentences should not follow one another), so be 2 separate commits with someone committing in between. So you need to check in on this at least twice!

Each new sentence must be appended to the **end of the story**. It is not allowed to add new sentences anywhere else in the story.

When you are done then post your story AND your GitHub link to our #story channel with your group name as the first line.

Step 3) Individually, each of you should go to at least one other GitHub repo from another team (see links on #story channel) and contribute one new sentence to their story. It is your job to find out how you can contribute to a public repository!

One of you create a PullRequest.md document on your repository. Into this file each of you should put your name and the link to the PullRequest you created for the other teams repo! If this document is not there and your link is not there you will not get points for this.

Step 4) The team should review all incoming PullRequests and either approve or decline with a reason.

Step 5) Vote for your favorite Story with thumbsup on Slack #story.

Part II.

Some Project Setup

2. Scrum Master & Git Master

Create a poll in your group board to have a team vote of who should be Scrum Master, and create a separate poll for picking your Git Master. Remember, the Scrum Master and Git Master can never be the same person during one sprint. The following example is how you can create a poll in Slack:

E.G. /poll "Who should be Scrum Master?" "Batman" "Superman" "Spiderman" "Deadpool"

After you have voted, in your team's private Memoranda repository, please copy the DeliverableX.md document and rename it to Deliverable1.md and enter the name of your Git Master and Scrum Master in there. (You will not be able to do this directly on Master since Master is protected).

3. Create a Git channel

One group member needs to create a new private channel on Slack called "[groupname]__git"
People/bots to invite to the group:

1. each group member
2. me (the professor)
3. Github

In order to invite GitHub, type "/github" into Slack, and follow the instructions for setup. Next, subscribe to your groups repository. You might want to subscribe to all commits so you are always up to date. Check the possible setting and set it up the way you think is best for your team (you can always change it later).

This should help you later on to be informed about new commits, new Pull Requests etc.

4. Create a Standup channel (optional)

One group member needs to create a new, **private** channel on Slack called "[group-name]__standup"
People to invite to the group:

1. each group member
2. me (the professor)

This channel is a designated channel for only containing your Stand-up meeting information. Additional conversations should happen in your groups __gen channel. You have different options here but this is supposed to help you keep track of your meetings and make sure you do them. The idea is to setup reminders or a bot so you can be reminded on specific days to do your stand up meeting. This can be completely asynchronously. You do have to read your team mates posts though. But remember these answers should be short. It should not take you longer than 2 minutes to answer and not much longer to read your team members answers.

You have different options to setup reminders for your daily scrum meetings:

- <https://slack.com/slack-tips/run-daily-standups-or-check-ins>: this page shows how you can use Slack to build in reminders to run your stand-up (looking into how reminders work is a great thing as well and might help you for reminding you of other things as well)

- There are several Stand-up apps available, many of them you need to pay for though. Which I do not want you to do of course. I installed standuply already and hope they will give us an educational license. You would have to create a new Standup report and add your team members to it and the report should be posted to your standup channel that you created.
- You can use other Stand-up apps if you like and play around. Feel free to discuss options on Slack under #scrum if you find something you like or know already (I personally have not used them yet, but thought it might be a good idea here to use an app to help you remember the stand ups).

You can do it anyway you like. The only condition is that in your groups standup channel your stand up meetings are included (no other discussion just the stand up meeting notes).

5. Choose your topic

You should all by now have taken a look at the Memoranda software and the topics document on Canvas (if not no time than the present). Think about which one you would be interested in. You should discuss which topic you would want to do in the project (this will be your topic for the whole session, so choose wisely). You can do that in the __gen channel of your group, you might want to do a poll and discuss pro and cons. Once you have chosen your topic, add it to the Deliverable1.md file in your group's repo.

6. Documentation

If you haven't already, go to your group's repo and copy the DeliverableX file as Deliverable1.md file and fill out the info you already have: Scrum Master, Git Master, Topic you chose, etc.

Check off everything you have already done (see bottom of file), remind each other of todos. You might want to consider using /remind in your __gen channel for any relevant things, so all of you stay up to date and you do not forget anything.

7. Quality Policy

You should now as a team decide on your Quality Policy/Workflow for Git and GitHub in your project.

Look at the QualityPolicy.md file in your groups project repository. Describe your workflow in this document and commit and push (PR) until it shows up in your master branch. This is what you should stick to. Make sure it adheres to the rules I gave you. See also "GitHubHousekeeping.pdf" under the Project module on Canvas.