Meeting 10th of February @ 12:30

Wigger forgot to send something to the tutor.

Tudor was absent.

Stefan: Last meeting you talked about test cases on a higher level than machine design, how is this possible?”

Tutor answer: for instance: we have a number of disks of two different colors, and at the end they have to be sorted. You can test efficiency, time, some high level requirements can be checked. Always if you specify, you have to check whether the spec is met. Make this a habit, to check whether requirements are met. I was told that you had a lecture where lecturer told about the “V”. You know now what we think about testing,

In almost no summaries thoughts about high level requirements. No goals, no machine, no group process (high level). First think about the basics, high level. What struck me about this group, was that the summaries were a collection of sentences. Compare your submission with the tutors better version. You might be required to do it again. This will help you in knowing what to do, when to do it. What is striking, is that no-one read the technical guide for documentation. There was only one that had a separate title page. ← That was not in the agenda that we didn't have (directed @ Wigger).

Tutor leaves for 2 minutes

Conversation about what is in technical guide for documentation, and what everyone missed in their summary.

Tutor returns

In group 14, we had only 4 people, 3 didn't show up. Tutor intended to ask for volunteer, was not needed, some random joined them.

“We had a question about the presentation: “We were wondering how long they have to be¨”. The presentations are fairly short, something like 10-15 minutes for 3 people together. From what I remember from last year, it was a 4 min presentation per person, it will be no more than an “elevator pitch”. This means you have to present the crux of what you are doing in the time that an elevator takes to get to the next floor.

The group presents their progress and divide that over 3 people, who all give their presentations. You have to convey something, but it cannot be too long, you have to be very careful about too much detail, but you CAN NOT be just on the surface. Some people may find it hard to present such things, especially those in a mathematical background. But you have to convey your results and plans and the wonderful work you have done

Can we give you the document for Friday tomorrow? Yes, I need to give feedback on the Work Plan, because it will influence you for the next weeks. So I want it before Thursday 0900 hour, then

I will take a look at it, and give you feedback on Friday.

Tutor: I need 2 days for feedback, and I am only here for two days a week, I will read your documents at home, I DO NOT want a .doc file.

Wigger didn't hand in his document in OASE correctly, it should be in group 16. Beware: you can hand in in another group.

The problem of the submission folder; you cannot see what others uploaded, only yourself.

Problem with Wigger prob solved

The documentation for the first exercise is ok if you have the documentation about

Your doc should have the spec of the problem, the way you intended to solve it, the way you solved it, and that you checked your solution

Every document according to technical guidelines.

What about the code? Nobody wants to read code, it's not there for humans. It should be in the appendix, but I should not have to read it (tutor). You should write documentation in such a way that I don't have to read.

How should we do this? Make it at least convincing (ie formal proof). If the loop is too easy, its okay. Think of invariants for loops, do not write a loop without invariant, invariants prove correctness.

And pseudo-code to explain? I think pseudo is a good intermediate between code and language, its good for depicting your algorithm. You don't want code, but not too many words either, pseudo good. Programming used to be taught in pseudo-code, was way better Q_Q. If you think it helps, do so.

You understood already, that in specifying your problem you also have to check your specs.

We will get the PP2 this afternoon (after 1400). You will also get a locker then, with a key.

Tutor room is @ MF 5.106

Tudor was ill, sent a message that he was “very sick”.

With the documentation: Most documentation should be about the exercise, how far do we need to go with documenting the compiler? You made the compiler yourself? For this occasion? No, for CS. It might be useful to have a single page of notes on the compiler. This is then the third appendix, compiler, PHP, assembly.

You can typeset your documents in LATEX if you want.

You can use an overhead projector, and blackboard for the presentations, if they are present. But a beamer and keynote is fine too.

Tutor will not grade presentations, people from STU will do so, it's all about the presentation for the grade (not content). You can get extra help for this on skills-lab. If you have a problem with writing, you can also do this on skills-lab.

There are many use-cases, they are a combination of user inputs and work the machine does.

User constraints are those things the user is allowed or not allowed to do in any state of the machine, ie use cannot pick up a disc when the machine is running.