Capstone course AHOT

I’ve worked as a software engineer for a year now. I have worked in several different project during this time.

can plan and manage teamwork extending over several months and connect their own work with team deliverables

I work on multiple projects at a time, but I have been working on one particular project for this whole year now. We mainly develop the program with another backend developer. We have other people helping us. I delegate the more front forward jobs to front end coders who are free. We have a project manager who is responsible for correct documentation and for billing. This means they need to be always kept in the loop. We use Jira as our management system. We try to keep the tasks specific and smallish. This helps so that people don’t work on the same stuff and do works that is already done by someone else.

We use GitLab as out version control. Version control is essential for when more than one person is developing something. We develop new features in their own branches and when a stable version is ready the branch is merged to ‘Staging’ for testing done by other developers and client’s representatives. Developing new features in their own branches makes it possible to have more than one person work on the project without having to constantly inform others about pushing and pulling to the project.

recognize the roles and responsibilities of members in a project team and potential challenges in teamwork

The main challenge in teamwork is miscommunication or no communication. Problems always arise when information flow isn’t flowing. A project needs to have a clear leader or manager who oversees the project in large scale.

One person should be the communicator between the client and the team (can be the project lead or someone else). It is clearer to the client if one person is the connection to their way. I’ve worked as the primary connection of the client in a few projects even though I’m a developer and not a team leader.

In software project possible responsibilities can be integrations, interfaces, UI design, UX design, architecture, digital marketing (ex. SEO), backend, frontend, QA, testing, accessibility, content providing, billing, and servers. I would say a team need at least 3-4 people to share these responsibilities in any smart way.

know how to communicate topics of one's own discipline to stakeholders with diverse educational backgrounds / can analyse and communicate the progress and performance of the project to external stakeholders

I have first-hand experience of this as I need to be in meetings with our clients weekly. Often stakeholders don’t have much if any experience with software development or computer science. It’s important to not use too much jargon or workplace/field specific nicknames. Many things that are self-evident or no-brainer to me are not to the stakeholders. So, I try to avoid going into too much detail about technical aspects of the project. Stakeholders are more often interested in the general progress, time management and risk management than any technical details. Stakeholders are almost always in one email chain and information needs to be told in a way that the least technical person understands it. If more technical information needs to be conveyed or needs answers from the stakeholders. I as a developer need to have patients and creativity to come up with new ways to ask and tell about my work until I can be sufficiently sure I my message has been understood or I get the answers I need.

know from firsthand experience how a complex and possibly ill-defined problem is approached and solved through dividing work into phases with their own sub goals, milestones, and deliverables

I've had experience with this since the first day of work. Most of the assignments I've gotten are extremely ill defined. Usually, I get these ill-defined tasks from my supervisor. The first step is to get to know the project. I try find the problems and requirements that pop up to me. Then very fast it's good to have a meeting with the client to determine the priority of different aspects of the project. Usually what pops up to me and what is a priority to the client are very different.

I develop webpages and webstores. One easy way to divide the work is by pages. One page can be divided to front and back. If the page need integration I start by testing and creating the connection. Every call can be considered as it's on subgoal. Based on the knowledge I get from this it's easier to determine what my backend requires from the frontend and start the planning and dividing the frontend tasks from that. UI and UX design are done based on these requirements. View (if designed well) consist of components that can be reused. Every component could be considered a subgoal. Then lastly, I make the frontend and backend communicate to each other. The size of the tasks needs to be assessed throughout the development.

The task I listed above are more personal divisions. Often the task the client knows about are a bit larger like one page or one feature. This means the client will know and get to test a bigger portion than just one front component or the connection to interface. Task should always be made smaller or bigger when necessary.