What are some potential benefits of agile software development compared to plan-driven "waterfall" software development?

On this question, we expect students to show a basic understanding of the iterative nature of agile and how this can be beneficial in IT projects, which we primarily covered in the second lecture of module 4. A key issue in many IT projects is that the problems to be addressed by technology are complex, moving targets, which are difficult to fully understand, and constantly evolve as the organization and relevant technology change. Further, the IT solutions must integrate into the existing complex and evolving system of technologies and work practices. In plan-driven "waterfall" software development, significant time is spent on planning out the technical solution in detail (sometimes for years), before any development and implementation is carried out. A risk is that both the requirements and the relevant technology is outdated before the software is implemented. It is also very expensive to learn that the solution actually does not respond to the actual needs of the organizations after significant effort has been invested in development. In contrast, agile software development emphasizes rapid development of so-called minimum viable products that can be used to quickly assess the usefulness of the solution, and from there, continue expanding the solution through small increments developed through cycles of planning, requirements gathering, design, development, and implementation.

What is "sociotechnical design" in relation to IT and digitalization projects? What can be different forms or scopes of design in such projects?

In lecture 3 and 4 of module 4, we saw that technology often offers poor usability for users, and that large organizational IT projects can fail due to an insufficient focus on how IT integrates with social arrangements within organizations. Also, the reading by Baxter and Sommerville discuss the need of a sociotechnical focus in IT projects. "Design" is a broad term, and in IT projects there are many things that are "designed". We have looked at how sociotechnical design in IT and digitalization projects can involve two things: 1) designing digital technology to fit well with human needs and organizational practices, and 2) designing both digital technology and organizational arrangements in tandem. We discussed that there is a broad range of sociotechnical design methods with different scopes, and we examined three concrete scopes: user interfaces, tools, and systems. Whereas the first two are oriented towards designing technology based on "social" needs, the third is about both designing technology and the social arrangements of organizations such as work routines, roles, and standards to best work together.

When an organization is in need of a new IT system, we often say that they have the choice of "buy versus build." For instance, the Central Norway Regional Health Authority (Helse Midt-Norge) made the choice of "buy" for their new patient journal system. Elaborate briefly on what this means and discuss some pros and cons of the two choices.

In the first lecture in module 4 we discussed different types of IT projects. In what we referred to as organizational IT projects, we saw that organizations often chose to "buy" a generic software solution, rather than to build one from scratch. Obvious benefits are that one saves development and maintenance costs by buying a generic solution, just like we as consumers most often buy rather than build the software products we use in our everyday life. A key challenge, however, is that the adopting organizations get less flexibility to design the solution to match perfectly with their specific needs. This is a major challenge for organizations as they tend to be quite different in terms of existing technologies and work practices. Also, the reading by Grisot et al., offers an example where a Norwegian health organization chooses to buy a generic solution, later to suffer from limited ability

to get their requirements covered in the generic solution as their needs were not aligned with other user organizations.