Concordia University
Department of Computer Science and Software Engineering

Hanieh QasemiBoroujeni

40057756

July 16, 2019

Two types of waste in software development

One of the most important types of waste in software development is Building the wrong feature or product [1]. This could include constructing a feature that is not often used by customer. Once I was working on an inventory management project, I encountered this problem. Our team was trying to build a perfect system, so we added some extra features to the project that we thought would be really useful for customer. For instance, we considered each product could have multiple units, which led us to build a unit of measurement system for handling and converting different units of each product. It took a lot of time, and we found out that our customer never used that feature, in that they defined all of the products in single unit. It was a waste, and the reason was we ignored user desiderata by keep working on features with low user value. We could avoid this risk by feature validation in which we should consider whether the problem is worth solving. Because, if users do not think this is a major problem, the solution wont be appealing. There are other ways of reducing this waste, for example we could benefit from customers feedback and know what they really want by frequent releases and participatory design. Another important type of waste is Psychological Distress [1], which is the costs of burdening the team with unhelpful stress. In 2009 I was working at this company which produced accounting software. It was full of stress; the manager was very obsessive about every unimportant thing. For example, instead of giving value to the business, he insisted on personal being on time at coming and leaving. Trust level was so down that he put surveillance cameras all over the place to make sure everyone was working. Every time he saw someone not staring at their monitor and moving their hands on the keyboard for a while he complained. Besides, the environment was clamorous, the deadlines were selected too short, and sometimes in the middle of doing some task we had to pause, rush into fixing bugs of another area, then back to the original task. As a consequence, no one was happy, and the result was obvious: Low team morale, rush mode, interpersonal conflict, and inter-team conflict. This problem could be solved by avoiding unnecessary stress. For example, the manager could impose less stress to the team if he was not so obsessive about unimportant things, or stress-related deadlines can sometimes be mitigated by reducing scope or extending the deadline, or facilitated mediation could help mitigating interpersonal conflict, or avoiding task switching as performance on these tasks is disrupted when a switch from one task to another is required. I learned

from these experiences that there are wastes in software development that we should recognize and avoid. We should reduce the risk of building the wrong feature or product by receiving frequent feedback from customers, and we also should avoid Psychological Distress.

References

[1] Sedano, Todd. Ralph, Paul; Praire, Ccile. "Software Development Waste". IEEE.