Continuous Delivery is a popular strategy adopted by most modern software development industries. It is, as described by Jez Humble and Dave Farley (2010), the process of get changes into production or into the end users in a reliable way. Changes by developers include adding new features, change configuration, bug fixing, and they have to be broken down into smaller parts that can be delivered respectively [1]. Getting changes into production requires the code to be sustainable, which means that the code should always in a deployable state. To achieve this, it requires automated software building, testing, and deploying, but we only deliver what is potentially profitable and beneficial to the business. Adopting Continuous Delivery allows developers to get feedback from market in time, which improves software quality and releases better products.

In our case study of Expedia, we had a deep conversation with an ex-developer from Expedia advertising group, to see practically how Continuous Delivery helps development work in big IT companies. Continuous Delivery is brought into effect basically under principle of agile framework and Scrum. Scrum is a development methodology for agile software development, includes PO, Scrum Master, and development team. It aims at responding to emerging requirements, and to adapt to evolving technologies and changes in market conditions. As a global travel technology company, the development team of Expedia has large number of unfinished tasks and projects. All of those tasks are conducted in the form of Scrum.

As for project management part, some teams in Expedia rely completely on Jira, an issue and project tracking software, while others make a combination of Jira and Mingo, which is also a project management software. He introduced how development teams make use of Jira as an example. Since teams are facing various requirements every day, both from project details or software developers. Requirements contain a general problem description, an expectation for the problem, or a specification of outcoming product. Some tasks derived by requirements are incrementally, which requires achieving a large goal step by step, hence they can be, and should be divided into multiple smaller tasks. Therefore, different requirements can be categorized into epics, stories, or tasks, based on size and complexity of requirements in descent order. Each of epics, stories, and tasks is assigned to different Scrums, through the “tasking and (?) meeting”, which allow developers understand the task size, what is requested and what is expected. Furthermore, A fixed bi-weekly Sprint is also a component in Scrum in Expedia.