Welcome to this course on Software Engineering. We all know the importance of software today, but we need to know how to build such systems and how such systems are built. What do we have to do in terms of thinking about users, how do we do the coding, how we do the testing and so on and so forth. So this course is a primer to understanding how to design such software systems, how to do some of the implementation and some aspects of the testing and so on. So what are the expectations from you? As a learner, you are expected to put in at least 6 hours per week on the course so that you can get the maximum out of it. And from our side, we will ensure that there are activities and there are dialogues that happen throughout so that you do not feel that it is one long monolithic video that you are watching. So what is software engineering? So if you look at the history of software engineering, it started off as an art. So you had good programmers who wrote good code and you had bad programmers who did not write good code and over time the expertise of these good programmers, they were documented over time which led to processes, models and strategies for effective software development. And thus the field of software engineering was born. So in this course, we will be looking at three important categories of topics. So one is the software processes. So what are different processes which are used in software development? You will be learning about processes like the waterfall model, the agile model and what are common practices in these models which are used in the industry. A second category of topics is tools. So what are tools which software teams use to build software? For example, what tools are used to capture requirements, how is software planning done, how is development done, how is testing done. So throughout the course, you will be learning about different tools which are used in the software development process. And third, we will also be talking about code. We will be talking about how you can organize your code, what are best practices for writing, clean and effective code. This course has been created using a learner centric MOOC model. What does that mean? It means that instead of simply having all the lectures in the beginning followed by all the assignments followed by the exams. What we have tried to do is to interleave these in a way that is meaningful for you to maintain your engagement. For example, there might be a video in which there might be a pause where we ask you to think about something and maybe write down your thoughts. So this pause is

very important for learning and that is called the reflection spot. Similarly, the videos might be short and immediately after the video we might want you to do some activity. So these are learning by doing activities, again which are very important for learning. Then there will be discussion forums in which we will pose questions to focus your discussion so that you can have a focus discussion instead of a scattered one. And finally, there will be lot of resources which will cater to your exploration in the direction of your interest. For example, some people may be more interested in coding whereas some others may be more interested in how a software behaves actually in the field. Whereas the third set of people may be more interested in talking to users of the software. Another key aspect of this course is the course project and through this course project you will be able to go through the entire software development lifecycle. So a software engineering course it can be very theoretical and the best way to get an understanding of what all you learn is to apply the processes, the practices, the tools in building an actual software. So in each week you will have some intermediate milestones which you need to complete to progress towards the goal of building a software product. So we hope you have a great time in this course and happy learning.