This journey that we've been going down, being a test case for the viability of changing our development model, has been an interesting one. There has been a couple of bumps down the road that I'll talk about later on. This review or retrospective will serve as a reflection over the first iteration of this project. Here, we’ll go over the things that we felt went over well and what things we felt didn't go over well in an effort to grow as a team to be more successful on our next iteration, our next project, etc. Doing this will help the company decide if it wants to completely transition over to a scrum-agile based approach to software development. It needs to be noted that this is a pilot program, there will be bumps and bruises. The company’s leadership needs to determine whether those bumps or bruises are something that can be overcome. I am simply reporting on what we experienced. Let's get started.

The First part of this project included a meeting with the client, Amanda. it was attended by myself, Ron the scrum master, and Cindy, the product owner. Cindy asked a couple of vague questions in order to get an idea about what Amanda wanted to do and I jotted down couple of notes that I thought would help get our epic going. Afterward, Cindy ended up doing informational interviews of end users to see what they wanted out of a travel website. The actual process of doing the interview went smoothly, but I think what we did with the information afterward led us down the wrong path. The information that we got from end users should have been forwarded to the client Amanda so she'd be able to understand what users wanted to see. I feel like towards the end of the sprint things got off track. Instead of taking the project in a direction that end users wanted, i.e. seeing a list of top destinations, we ended up doing something entirely different, i.e. making a slide show of the most desirable destinations. I think if we would have taken what end users wanted to see in a travel website to Amanda, our final product would have been a bit more representational of what end users wanted.

The next couple of issues that we ran into was in the test and development phase with Brian and Nikki. Here we started noticing some interdependencies in the things that we wanted to implement. Specifically the profile that end users wanted to see implemented into the website and how it would feed into several areas: travel suggestions based on what's in the profile and travel history, what sort of travel medium the customer would want to go by, say a cruise or by plane or rail, etc… Being able to identify these interdependencies in the beginning would have helped Nikki plan the intricacies of how the program should work. At the end of the sprint, the slideshow did work which is something that we want. However, I feel like more could have been done to implement variables that would be coming from the profile feature of the website. Even if these variables were commented out. They would have helped prepare for the integration of a profile feature in the future.

In the e-mail that I wrote during module four, I addressed some of these interdependencies. However, looking back at the email, I should have also sent this e-mail to Nikki, especially since she's the one that's going to be developing the code to run the website. The answers that I end up getting from emailing Christy and Ron would have helped set her up for success instead of having to run thru me, as the tester. Reflecting on the role of the tester and developer, they are both dependent on one another. You can’t really talk about one without including the work that the other does. I feel it necessary that anything that we end up sending to either person, developer or tester, should just be sent to both since they work so closely together.

Reflecting on the change of direction that Christy announced, it felt like Nikki and Brian were almost immediately taken by surprise and we could see that they were initially not open to the idea. Once we ended up taking a look at what needed to be changed exactly, it turned out the difficulty level in implementing this change was negligible. Initially, the reaction from Nikki, Brian and Ron were something that I would expect to see from someone working within the confines of a waterfall model, where changes like this might be detrimental to the project. The scrum-agile model made it easier to accommodate the change with little to no effect in the timeline. Even Brian’s test cases didn't need to be changed drastically. Nikki let Brian and Christy know in an e-mail about what things were added to the code, what things needed to be looked at, and the possibility of Christy adding some simple questions to her interviews to avoid a change in direction toward the end of an iteration. All sides were addressed quickly, with functioning code being delivered in the meantime.

Reflecting on some of the scrum-agile principles and what they allowed us to do within the project, there were three principles that really helped in our teams case. The first was delivering working software frequently. We were able to see the byproducts of what we were doing and this allowed us to make adjustments as required. The second was welcoming changing requirements. Even though this appeared to be one of the more difficult things to do, I think it really illustrates why we may need to adopt this type of development model, especially in an industry that ends up changing quickly based on trends. If we were in a waterfall type of development model I don't think we would have been able to accommodate change for the client. The third has to do with us being a pilot program, and that's feedback loops. This allows us the opportunity to really learn And grow from the things that we've experienced so far.

Now whether the scrum agile approach was really appropriate for this project is a completely separate question. I think the fact that we had a change toward the end of a project, the scrum-agile approach helped us deal with the situation. If we were operating in a waterfall style approach to this project, it could have potentially led to the loss of this customer or to the loss of her competitive advantage that she wanted to implement into the product. Another thing that happened was the suggestion of a mobile application. Given how popular mobile applications are, I feel like the scrum agile approach allowed us to stick this idea in a backlog for later use. A waterfall approach would have caused us to completely neglect this idea. However, despite all of the mentioned things, this project didn't seem like it would be outside the scope of a waterfall approach to accomplish. The scrum agile approach just gave us flexibility to accommodate changes and ideas that were presented to us. Both the customer and ChadaTech benefited from this.