# Customer engagement summary

* Our customer engagement has occurred through the weekly tutorials
* We had good initial discussions with the customer to get an idea of the specific functionalities the customer wanted implemented
* The customer helped prioritise our stories which we then used to plan our releases and each individual iteration according to the functionalities most important to the customer
* Since then we’ve not been able to get as much from the customer because our development has lagged behind the targets we set for ourselves, and our initial stories didn’t focus enough on specific deliverables for the customer
* We’ve been able to set new goals with the customer in terms of what to deliver when, due to our falling behind our iteration plans
* As of late we’ve mostly just discussed certain aspects of the web interface design with the customer to tailor it to their needs and help us in coming up with more specific user acceptance tests
* From here we need to continue to work with the customer in setting out deliverable iterations leading to a second release that encompasses the highest-priority functionalities that were established at the start of the development process

# Extreme programming summary

* We’ve tried to adopt as many XP processes as possible, but some have been very difficult to implement due to the lack of a conventional shared working environment
* Obviously our focus on stories and subsequent release/iteration plans was consistent with XP planning and design processes
* We’ve logged the time spent on tasks and produced simple burn down charts to better track the project, but this hasn’t been as useful without proper stand-up meetings and retrospectives where team members are more accountable
* We’ve used several spikes in the form of additional servlets and blocks of code to quickly and easily explore the means by which the service will function, including getting information submitted by the user, storing and retrieving it with the Datastore, making that information available to be output, and integrating the Google Accounts API
* Some of the spikes will be used to inform new classes and methods, and some of the servlets should be dispatched completely
* We’ve endeavoured to use test-driven development where possible, although it’s been complicated by the fact we’ve focused on the web interface initially, which doesn’t lend itself to unit testing as much
* When we’re able to begin refactoring the messier code blocks into separate classes and methods we will be able to make better use of unit tests to inform how the classes and methods should operate
* We’ve used standard development tools to test, build and integrate but we haven’t made use of more sophisticated build and continuous integration tools because of the team’s unfamiliarity with such tools and the difficulty in setting them up over multiple computers
* Such tools would definitely be worth learning to employ properly if we had the time
* We haven’t pair-programmed and our code reviews have been informal over the internet, which is far from ideal, and more stand-up meetings including code reviews are required to increase the team’s shared understanding of the project