# Methodology

Our team has chosen to use Extreme Programming (XP) as its process methodology. XP is an agile methodology that encourages pair programming and is test driven. These address some of the risks which our team faces and helps ensure the quality of the product. It differs from other agile methodologies in that it leans towards development practices whereas others lean towards project management practices.

We chose XP for several reasons. The first of these is because of pair programming. Pair programming creates a “knowledge share” between programmers working side-by-side. This means that we will have multiple experts in different areas of the project. If a team member is being assigned an unfair workload because he is an expert in a certain area of the project which needs a significant amount of work, then another team member who has pair programmed with him can come help him with that workload. Pair programming also reduces the time spent learning, as the two programmers working together can share the knowledge they acquire between themselves efficiently, this reduces the impact of our risk that none of our developers are very strong at developing the customer-facing front-end.

The second reason we chose XP is because of the focus on test-driven development. This will lead to higher quality code, ensure that our code matches the requirements, and hopefully a more stable final product. In addition, it will allow us to hopefully catch bugs as they appear in the code.

The final reason we chose XP is because it is an iterative process that is customer-facing. We want our product development to be as transparent as possible. In this way, we can address concerns and issues as they come up. Also, this will allow us to guarantee the delivery of portions of the product as they are developed, thereby removing the opportunity to “dump and run” when the project comes to its conclusion. By choosing an iterative process, we can work with the Wegmans team at the end of each iteration to get it into their environment so that they can continue development on the product after we finish our final delivery. This should reduce the risk of the Wegmans team being unable to import the project into their environment after the project is complete.

# Components

Extreme Programming is an agile methodology and contains many of the same artifacts and components as SCRUM and other agile methodologies. The main components utilized are separating development into iterations, having a backlog of tasks, and making use of a burndown chart to help with time management.

XP emphasises pair programming in order to reduce intrinsic knowledge, improve code quality, and increase development speed. Because of the reduction of intrinsic knowledge, it mitigates the risk of not knowing how a portion of the software works in the event that a team member is down and out.

Test-driven development is another important component of XP. This means that tests will be written before the code is written, in order to ensure quality code. It should give the team a clearer insight as to what they need to develop, as well as a means of testing the code once they write it. In addition, these tests can be run during each iteration to make sure nothing breaks as a result of new modifications to the codebase.

# Modifications

The only modification that we are making to XP is that we will not be likening our application to an existing, real-world process. Instead, we will be doing an up-front *high* level architecture. We will then present this architecture to the Wegmans EA (Enterprise Architecture) team for review. In this way, we can make sure that we address high level design concerns before we begin our development iterations.