**Implementation**

**Methodology**

As explained in design section, we choose to implement this application using Prototyping methodology. This approach allowed us to build a working prototype pretty early in the process. Because our application is a collaboration of many different web technologies, first step was to make sure we can actually make them work together on something remotely resembling fantasy sport game. If that’s the case we would then mutate the working prototype toward the final project objective

**Prototype (Version 1)**

1. A simple database with a few mock tables and simple relations between them was created in Microsoft SQL Server Compact on localhost.
2. Connection to database was assured.
3. New .Net Web API 2 with Entity Framework project was created in Visual Studio 2013 with a connection string to mock database.
4. One ORM Entity and Web API Controller with GET, POST, DELETE and UPDATE methods was created.
5. Application was tested until worked
6. Simple HTML and Javascript client which would consume the API was created.
7. JQuery Ajax methods for GET, POST, UPDATE, DELETE request towards Web API were crafted, tested and refactored to acceptable form



Figure 1 JQuery Ajax Request Method

At this point we had a “working prototype” of an AJAX application consuming REST like web service resources from the persistent storage. We knew we can do this on localhost. Next step was to make sure we can deploy this on internet.

**Prototype (Version 2)**

1. SQL Server was created on Azure Cloud North Europe datacentre and the localhost mock database was migrated there.
2. Visual Studio Web API project was configured with a new connection string to the database on the cloud.
3. A web server was created on Azure Cloud North Europe datacentre and Visual Studio was configured to publish Web API there.
4. Deployment was tested.
5. Simple web forms and tables were created for our client. At this point we were turning the client into “database admin pages”