Advanced Group Project - Proposal

The project we are proposing is a Web Based Agile Project Management Dashboard aimed at small to medium sized software companies. There are too many reasons as to why we've chosen to this to list them of all but the some of the main reasons are as follows:

* The skills involved are real world relevant - I'll go into more detail about the construction decisions we've made below.
* I personally have experience of implementing similar software at my placement and I have learnt a lot about managers/directors would like to see from such software.
* We've identified an interest in this idea being aimed at the small to medium businesses.​
* Once we have the barebones up and running we plan to use our own project to keep track of our project, so it can positively reinforce itself. And hopefully provide a wealth of agile data to be included in the project documentation.
* We have an extensive list of ideas that can tie in to our product that can be implemented as add-ins that we think could really make our product stand out. These ideas include, but are not limited to: an interactive Kanban board, an interactive employee time calculator (making use of employee velocity data and work currently assigned this iteration to produce an idea of whether an employee can take on more work - an add in we feel could be really useful for our intended market).

Hardware/coding decisions:

* We will not be using Oracle APEX for this project - despite on the face of it being the perfect base for this project we feel it is far to restrictive in a number of ways.
* We intend to use a MySQL database and build our own front end and links between the two.
* We will therefore be using the following skills:
  + ​Backend skills - SQL - (create, alter, insert, and drop scripts and queries as necessary).
  + Linking skills - PHP & JSON (to take the data from the database and give it to the web front end in a format that it understands).
  + Frontend skills - HTML, CSS, JavaScript (to display the data etc.).
* ​I'm sure you'll agree that being able to leverage these skills makes us more attractive to future employers.
* We will display project metrics using JavaScript graphs - I currently have a basic, proof of concept example working with Google Graphs.

What we plan to do:

* Work Agile
* Get the database up and running.
* Focus on getting one burndown working for one iteration in one project first of all and then build on that - by adding multiple single iteration based burndowns, a release based burndown and a velocity chart. Resulting in a web page that displays comprehensive project data.
* Build a customer front end that includes a set of forms to enter project info into the database​.
* Once this is in place we want to able to show more than one project's data at a time - so a manager/director can quickly get an idea of how well multiple projects are doing.
* With this in place we want to make the page customisable by holding each graph in a toggle-able widget- this is slightly fluid idea at this point as we might decide to implement a widget type display from the very beginning.

Our main focus is to display project data in JavaScript graphs and we plan to do this as follows :

* ​With data in the database we will make a connection to the database in PHP and run a SQL query in PHP to return the desired project data, namely effort points remaining and a corresponding date as a key value pair.
* This data will be turned into a JSON array and passed to a JavaScript graph.
* The graph will use each part of the key value pairs from the JSON array as coordinates to plot on the graph.
* The graph will draw a line between the coordinate creating our burndown graph.
* We also plan to show a second line on the graph that relates to the total effort points assigned to that iteration/release - so if the burndown line goes up you can quickly see how that relates to the total amount of work, i.e. whether it only goes up because more effort points were added to that iteration/release, please see the below/ the attachment for a screenshot of my example.​

Other things that we thought about:

* ​​We want to keep this as lightweight as possible that means keeping installation as simple as possible. So we plan to offer our product as a "Cloud" based product, wherein we host the customers project data. We are aware of the security implications implied by this decision which is why we plan to offer an in house installation via database scripts based on the customer's Database management system - Oracle, MSSQLS etc.
* Ideally we'd like this Cloud based service to be a easy to use self service type set up, requiring little interaction from us by keeping a 'stock' of databases going on the server using triggers.