Problem Statements and Design Goals

**Problem Statements** and **Design Goals** help work through problem and solution definition. Once established they should be referred to throughout the project as a check balance. Are we still meeting our original goals? Focused on the problem at hand or are we in danger of scope creep?

**Problem Statement: Clear definition of problem being solved**

It is important that we understand the problem before defining a solution. Articulating the problem helps work through what is known about the problem and where there are gaps in knowledge. The problem definition should be a living document that is revisited and updated often as the problem becomes better understood.

A problem statement can express the value of a project and define the scope or focus of a project. An example problem statement might look like this:

The problem of... (problem description)

Affects... (the people impacted by this problem)

The impact of which is... (how are the people effected by the problem)

A successful solution would provide... (the **benefits** of a proposed solution)

**Design goals articulated**

Design goals help us stay focused on what we've determined to be most important in a project. They can serve as a quality check by making sure the designs meet the intended goals. These are typically at a high level and can be thought of as driving principles for a particular project. These can be thought of as related to if not the same as the benefits described in the problem statement. The design goals are likely fuzzy at the beginning of the project. The goals should be revisited and iterated on early and often.

Example Design goals (from **Rich Text Inline Edit Design Goals**):

- Don't change the display of information between editing and displaying edited information (i.e. wysiwyg).
- Don't require user to leave or change their context to edit information.
- Make it clear that changes can be made and to what.
- Give users edit tools that make sense for their specific context. Don't clutter with every possible edit tool.
- Smooth & easy keyboard interaction.
- Allow users to easily see and use the edit tools they need (as opposed to giving them many tools that aren't relevant or useful most of the time).
- Give users feedback and confidence about what is saved.
- Give users an easy way to back out (i.e. undo change & error recovery).
- Instant editing
- Allow user to complete their task without being distracted by software or losing train of thought (i.e. while they are reading text they can easily make a quick edit).
- Give users access to information they need to complete their task (i.e. information on other pages, etc.).
- Assist users in not making mistakes with their edits. If mistakes are made, we need to give them an easy way to identify and correct.