Risk Management

Task: Construct user interface

Likelihood: Low (We have all constructed several website interfaces using Django)

Severity: High (If you can't access the site, it is no use)

Consequences: The website won't work or won't improve efficiency.

Work-around 1: Go back to the old way of doing business for a bit.

> Difficulty: Easy. Joe has been an employee for a while.
>
> Impact: This would largely increase the workload of Joe, and would restart many of the problems we're trying to solve by creating this application.
>
> Pros: Simple. (No technology involved, pen and paper, Joe is used to doing it like this)
>
> Cons: Similar to the impact, Joe would resort to the same problematic way of keeping track of tools. He also would have effectively hired us for nothing, which would make him angry.

Task: Add and remove tools from database

Likelihood: Low (We have all worked with models and databases in Django before)

Severity: High (There need to be tools on the site)

Consequences: The tools won't be visible and we will be unable to determine which tools have been checked in or out.

Work-around 1: Construct a simple model that only shows the tool, not how many are available, if it's checked in or out, etc.

Difficulty: Easy. Making a model should be quick for this.

Impact: The company would be unable to determine the inventory at any given time.

Pros: Simple to implement.

Cons: The company will have to manually check the inventory and increase workload.

Task: Log when a tool has been checked in/out in the database.

Likelihood: Low (We have all worked with databases in Django before)

Severity: High (The ability to check out tools is the core functionality of the site)

Consequences: Incorrectly changing the 'checked out' status of a tool will cause conflicting records for Joe, and could possibly affect the database. It will render the site's purpose useless.

Work-around 1: Joe manually keeps track of when tools have been checked in or out.

Difficulty: Medium. It's relatively low effort but requires accurate bookkeeping and can take time to manually look at the records for each tool.

Impact: If Joe resorts to manually logging the status of a tool, he will likely have conflicting records with the tool database.

Pros: Doesn't require any technology, and can be logged and organized as Joe likes it if his opinion is to change.

Cons: Inefficient and not automatic. Where a database can quickly store all information about a check-out/check-in, Joe will have to log any info he wants by hand.

Task: Reorder a tool.

Likelihood: Medium (We have never really written a system that places online orders for tools)

Severity: Medium (Tools need to be replaced but don't break often, and extra tools aren't always the highest priority)

Consequences: Joe no longer has a tool in stock that was once in his online inventory.

Work-around 1: Joe manually reorders a tool. (No use of the application)

Difficulty: Easy. Joe can contact his selected tool supplier and place an order.

Impact: Manually ordering tools would slightly increase Joe's workload, as he has to keep track of the order statuses, and make the calls/emails/etc himself.

Pros: Simple. Joe has complete control over the order.

Cons: Joe may forget to place an order, he has to keep track of the order manually, and needs to keep his employees in the loop.