**SAB’s Project Management Active Instructions Project**

**Needs and Wants**

**Thank You**

Thank you for your help and interest in making Project Management’s vision of Active Instructions become a reality. Our collaboration will lead to a very useful and powerful tool for SAB and manufacturing in general.

**Needs**

**Goal**

The goal of this project is to create an active set of instructions that can adapt to the true pace of assembly. This program will enable any number of builders to work on assembling the airplane without interference (Two builders will not build on the same part at the same time, two builders will not need to share tools within tasks, and no task is recommended until all predecessor tasks are completed.). Three constraints will accompany each task: a tool constraint, a part constraint, and a predecessor task constraint. These constraints will determine the steps available to the builder.

The output of the program will be steps that a builder can choose that will not interfere with tasks other builders have chosen. Included in each available step will be information explaining where in the master instructions a builder should reference.

**Details**

The following section will define how to address the goal of this project. All aspects of this section are suggestions on how to best accomplish the goal. All points in the details section can be edited if a better way to accomplish the goals is found.

Interface

There will be a graphical interface where builders can interact with the tasks and save the state of the program. This interface must be simple, clear, and easy to navigate. The style and color scheme of the interface must not inhibit operation. There must also be in import button where more tasks can be added to the system.

Multi-User Applications

Since this system will be used by many people, more than one task at a time must be able to be selected. Once a task is selected, the parts and tools used in the selected task must be reflected in the selectable tasks.

This could be accomplished by attaching three states to every task: not started, in progress, and completed. Selecting “in progress” would influence the visible tasks by removing tasks that share the same parts and tools. Selecting “complete” would free any tools and parts that could be used in future steps. Selecting “complete” would also make visible any instructions that depend on the completed step to be completed.

Saving

The current state of the program must be able to be saved. When the program is started again, the program should display the last saved state of the program.

Import

Because instructions will be generated throughout the build process, new instructions must be able to be imported into the system. The inputs of the program will be: Task Name, Task Reference Steps, Task Part Constraint, Task Tool Constraint, and Task “Predecessor Task” Constraint.

**Wants**

The following are functionalities that would make the system much easier to use and more helpful for our process. These ideas are not as fleshed out as those in the “Needs” section, so if you feel there is a better solution, please feel free to approach me with that idea.

* The system will track who the Builders and Foramen are on each task and the time a Builder spends completing a task (This information should include a start date and time and completion start date and time.).
* Included in each task will be a way (either link, embedded, or QR code) for builders to see pictures of the parts and CAD models of the assembly they will be working on. (All pictures and CAD models will be uploaded to the hanger computer.)
* Included in each task will be a list of parts and tools associated with the task.
* Quality issues will be tracked by the program and the Active Instructions will adapt to parts that are no longer able to be used.
* Because tasks may take multiple days to complete, there should be a way to “pause” a task so that the time taken to complete the task may be accurately recorded.
* There should be a “Notes” button associated with every task so that comments can be made if necessary.
* There should be a way to impose constraints on the entire system manually. (For instance, if the rivet gun breaks, I should be able to input that into the system and eliminate all steps that use the rivet gun from appearing to the builder)
* The following information will be able to be exported from the program (All information should be grouped by task):
  + Task
  + Builder Name
  + Foreman Name
  + Notes
  + Time Started, Paused, and Completed
  + Quality Issues (Should be able to export this information separately as well)
    - Part Name (Will be a drop down menu)
    - Builder Name
    - Foramen Name
    - Issue
    - Steps Taken to Fix Problem
    - Has Andy Myrna been Notified? (Y/N)
    - Is the Part Usable? (Y/N)

**Conclusion:**

Once again, thank you for taking the time to help us. This will revolutionize our organization and create numerous possibilities for real world implementation of Active Instructions.

If you have any questions, please direct them to [strickr2@illinois.edu](mailto:strickr2@illinois.edu).