Growth Logger Pitch

Trevor Young

Trevor.Young@onsemi.com

# Purpose

Create a program with tools described below and solve a real-world problem with code. Every day, every shift needs to perform a pass down. It includes one person, sometimes a supervisor, collecting data from multiple people that write down all operations, problems and notes for the day on a piece of paper and sometimes, on special days, a folded paper plate. Reduce time spent and create a standard practice for this task.

# Scope

This document is intended to be the first “Plan” section of the plan-check-act-do loop for this project. Identify problems with the current method. Brainstorm solutions and communicate what the program aims to achieve and define an outline for the project. Email [Trevor.Young@onsemi.com](mailto:Trevor.Young@onsemi.com) with topics concerning the program.

# Definitions

* Pass down: Email created each day containing all operations on shift within department.
* Sub user: Technician that is using GL to record their actions and operations.
* Main user: Technician that is using GL to collect all GL data from Sub users to create the pass down.
* GL: Growth Logger.
* GL data: At the end of the day, sub user will have an output to send to main user.

# Tools

* C++: Powerful programming language. Probably overkill for this application. Looks good on resume.
* Java: Virtual machine programming language. Making it write once run anywhere… but not IOS.
* SQL: Databasing software. Probably not needed here/unnecessary complexity to code.
* Swift: Programming language for IOS powered devices.

# Current method short comings

* Main user must fetch data from each area of growth. Furnace, red room and seed coating. They will also need to:
  + Be able to read sub users hand writing.
  + Understand non-standard terminology that can change day to day.
  + Transfer hand written data to an email.
* Majority of technicians are using pen and paper to record notes and operations, moving away from the company goal of neutral emissions/green initiative.
* Miscommunication leading to:
  + Incorrect data entry.
  + Missed data entry.
* Time taken:
  + Findings sub users
  + Waiting for email or text response from sub users.
  + Transferring data from written word to email.
* The time taken also effects the accuracy of the pass down. If it was a faster process, it can be done closer to the end of the day, thus not missing anything that was logged the last hour of the day.
* Main user has to be on the look out for duplicate data. Example, main user unloaded the same furnace as the sub user. Both recorded this action.
* Sub user may have even more sub users they are collecting data from before they give data to main user. This creates a game of telephone effect. Or “he said she said”, if you will.
* Pass down non-standard.
  + Some pass downs will have information others do not. For example, furnace op may include what was taken out of furnace, GW or SB. Others may not. Some may include furnaces out of service so on.
  + Everyone is using their own template for pass down. Can’t find info in the same place on each pass down.
* Lots of repetitive actions and wording. Note: This is why I think code is a good solution.

# Goals

Note: This document will not outline “how” GL will address these tasks as much as the goals GL aims to accomplish. This is so I have flexibility as to how it is done.

* Reduce time spent on pass down for main user.
  + Print operations and actions to a simple output that the sub user can SMS text or email to the main user. This is the GL data’s goal.
  + Every other goal helps cut down time spent on this task.
  + Collects GL data from each sub user and does not allow duplicate data.
  + Data sent to main user is standardized making it easier to check all data is entered correctly.
* Take work load off main user. Note: Sometimes pass down is done by supervisors.
  + Sub user sends GL data to main user with email or text at the end of each day.
  + Main user does not need look for and or ask for GL data from every sub user.
  + Main user does not need to look for duplicate data entries.
* Minimize communication errors.
  + Introduce standard operating procedures with a work instruction.
  + Limit and standardize sub user GL data. Use standard terminology by creating prompts rather than leaving naming tasks up to each individual sub user.
  + Reducing the need to communicate will reduce chances for miscommunication.
* User friendly/makes the task easier.
  + All forms of output should be easy to use files such as text files.
  + Sub user selects a prompt and the data is logged.
  + Main user can attach file or simply copy and paste GL data into pass down.
* All report to one main user.
  + Everyone is a sub user. No sub-sub users reporting to a sub user that reports to main user.
  + Sub user simply send GL data to main user.
* Easier to record data.
  + Example: technician running furnaces loads a kit. Opens GL and selects “loaded GT kit” and only needs to type furnace number.