

Estimation Accuracy 2: GotoGro-MRM

Evidence of Task Completion

The task selected for estimation was the PDF report generator. This product required multiple subcomponents: UI button, trigger code, SQL, processing code to turn the SQL table into a visual report and the overall UI for displaying the report. Two team members did this task together and reported their timings as shown in **Figure 1**.

Product	Item	Description	Team Member	Est. Time	Actual Time
PDF Reports	SQL Select Item Statement	Statement to select a count of all items sold and tally their cost basis into a chart	Dylan	0.3	0.1
	SQL Join Statement	Statement to join the sales table with the member table to capture what a given member bought inbetween two time brackets	Dylan	0.4	0.5
	SQL Select Member Statement	Statement to select a specific member and present their information	Dylan	0.3	0.1
	Implement UI	Physically place the button on the screen as well as two date boundary fields	Nic	0.5	0.6
	Trigger Code for Button	Writing the python script which will trigger the SQL statements are return the results when the button is clicked	Nic	1	1.5
	Processing Code for Output	Writing the code that will format the return values of the SQL statement, including formatiing into charts in necessary	Nic	1.5	3

Figure 1: Reported time vs estimated tome for PDF generator.

Screenshots of the finished product are shown in **Figure 2** and **Figure 3**.

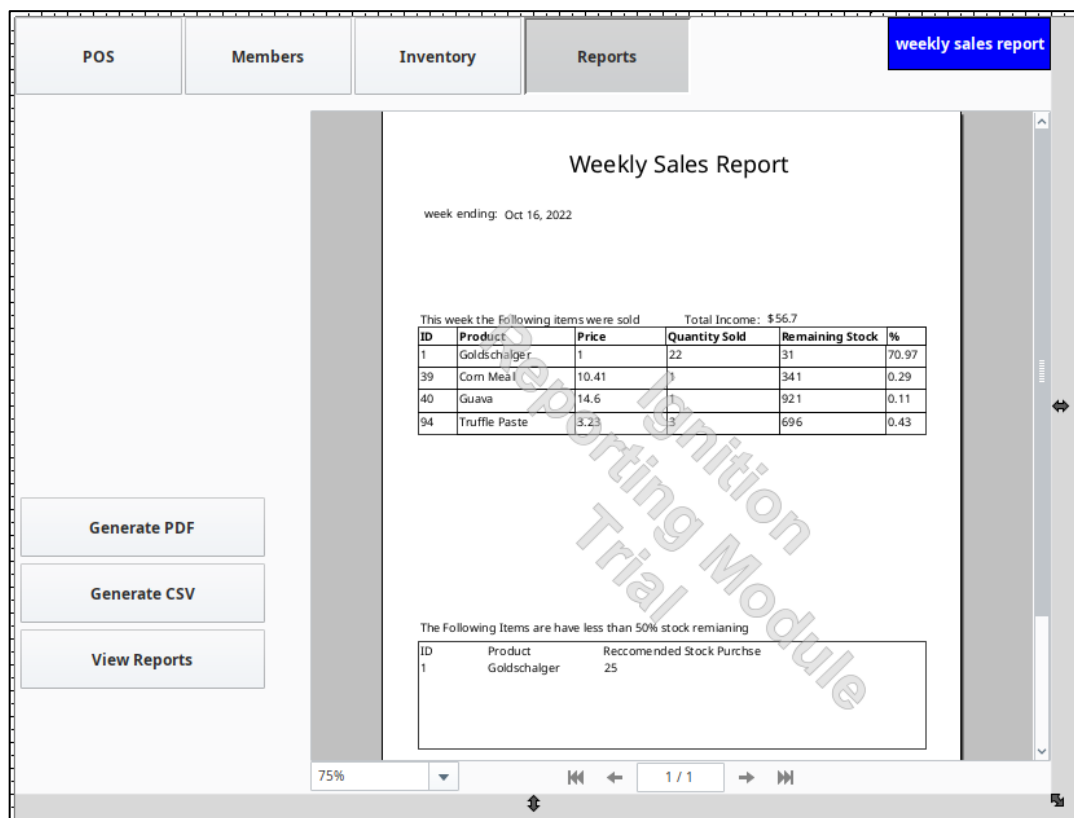


Figure 2: Report interface UI showing button and display window.

Weekly Sales Report

Week Starting: @StartDate@

Week Ending: @EndDate@

This week the Following items were sold:

ID	Product	Price	Quantity Sold	Remaining Stock	% Sold
week Header					
@Ite	@Name@	@Price@	@sold@	@stock@	@percent_sold

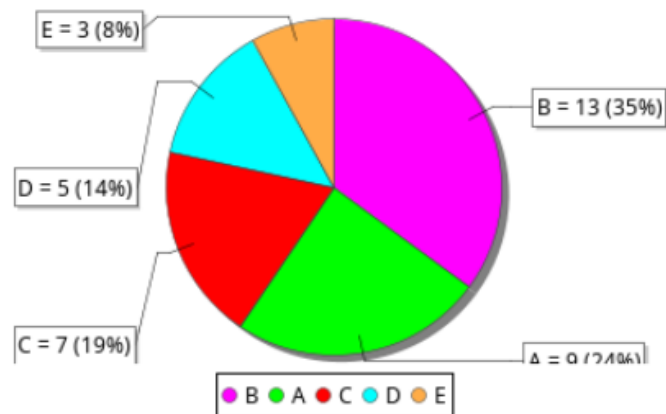
week Details

Total Weekly Income: \$@week.total.income@

week Summary

Table Group

Income Breakdown



@Page@

Figure 3: An example PDF report as rendered by the software (no data).

Discussion

Referring to **Figure 1** we can see that the predicted time was 4 hours while the actual time was 5.8 hours. Overall, the estimation was inaccurate (greater than 10% off), but not too outrageous for a software task where there are constant changes in problem definition, scope creep and debugging problems.

For the most part, the tasks were reasonably estimated with the major outliers being both coding components. Coding in general is often underestimated and can balloon into large chunks of time spent especially when an unidentified issue crops up. This, in tandem with an evolving problem definition is what caused the last task to take double the length of time it was allocated.

To start the PDF was going to be a very basic reskinning of the csv report with data more easily visible, but the csv data dump was being made at the same time, so this was accounted for – we did not expect to be able to reuse the code. We were however expecting a much greater similarity than what we ended up with and this was partially because of the functionality of the ignition software which indicated that we could present the data in a much more visual way (see **Figure 3**). This meant that instead of following the framework for the csv report, we had to customise the code to return table rows and summarised tallies for the pie chart.

With this in mind, it can be seen that the time increase was mostly driven by a scope increase, which directly led to a better result. So, in this sense, the extra time wasn't an issue with execution but more with the planning phase. To improve an estimation like this in future there is something to be said for fully understanding the capabilities of the development software. Had the team spent more time familiarising with the reporting tools ignition offered a better PDF design could've been conceived and therefore driven a better estimation, likely closer to 6 hours and within the 10% target margin.

As a final note, we must also reflect on the methods used for this estimation were not the most accurate in the first place. Using a combination of size-estimate, analogy and the delphi method works best when the members involved have extensive experience with similar tasks/tools. As mentioned above, the team does not have expert familiarity with the ignition software, nor are we industry professionals. Over time, our understanding of how long it actually takes to work through specific coding tasks will improve based on more experiences.