**Performance Measurement**

**Nilanjan Das**

The performance measurements are carried out by the following steps.

* **Code Snippet:** The stopwatch code snippet is added to the intended places. For example………
* **Item Sell:** Below script is used for item sell.

<https://tom.wincor-nixdorf.com/projects/IKEAIPOS/repos/tools/commits/86b70f1090c5578ce03c5c64fd9260ce4c20f7b8>

The scripts are developed in Autoit. For this Scite editor is installed. In the script, we log in to pos for 50 times. With the first login, we sell one item. With second login we sell two items…like this, in 50 th login, we sell 50 items.

* **Item Sell Data Processing:** We then collect the log from the log file. We store the logs of interest in a separate .txt file. We have prepared a separate Autoit script which can read the time stamp logs from logfile which we have saved. Our Autoit script provides the time stamp as output and saved it in a separate .txt file which we have chosen. After the raw data is collected in excel we have processed it according to our needs.For our case (testing of IKEA BV8, Core, BV7) we have taken values after 3 decimal points. We have used the following macro codes. The functionality of these macro codes is as follows. First, this macro put two extra rows between item sells for each login(Column A in Fig. 1) (e.g. two-row wise spaces after one timestamp, two row-wise spaces after two-time stamps.. two-row wise spaces after 49 timestamps). Then we have taken time stamp values after 3 decimal points (Column B in Fig.1 ). In the next step, we have calculated the sum of all timestamps for each login cycle. We use this macro code.

For this macro to work we have put an extra blank row as our first row in excel. Here c1 and c2 represent the first two rows of the column where we are expecting our summation to be displayed. $C$1272 is the first blank row after the very last item in the time stamp list. In the very last step, we remove the rows containing 0's and blanks (Column E). Below is the picture of the excel for further reference.

Fig1: Excel for Item SellTime Log

* Individual Tax Time: For individual tax time data collection we have developed the below AutoIt script.

<https://tom.wincor-nixdorf.com/projects/IKEAIPOS/repos/tools/commits/8853f49952872cf3bb82ca42a5253a668e23ecdb>

After collecting the data from the log files we keep the data in an excel file. For each item, we get two entry for individual tax time. Like before we pushed two blank rows after each login cycle. The difference is for each login cycle we get twice the entry compared to sell time. For this, we pushed two blank rows after two rows, two blank rows after 4 rows so on. Below is the macro code used for this data processing.

<https://tom.wincor-nixdorf.com/projects/IKEAIPOS/repos/tools/commits/bb7be6b76b5b4f73e92f145ab5292962e5561de9>

* Transaction Level Tax: For transaction-level tax, we have created this the below AutoIt script.

<https://tom.wincor-nixdorf.com/projects/IKEAIPOS/repos/tools/commits/86b70f1090c5578ce03c5c64fd9260ce4c20f7b8#AutoitScripts/TotalTax.au3>

For this script to run like before we have to select the log file and also we need to select a target location where it provides the output with transaction-level tax time. For each set of transaction-level tax, we get two timestamps. So we need to add this two-time stamp. For this operation, the excel looks like this.

* After-Tax Add: For this step, we simply add the tax time from the previous two-step (i.e. Individual tax time and transaction-level tax). So the excel for this step should look like below.
* Ratio: In this step, we were calculating the ratio between tax time and total sell time. The excel looks like below.
* Graph: Finally we prepare the graph for the ratio which we have prepared in the last step. For drawing a graph we are using excels in build features.