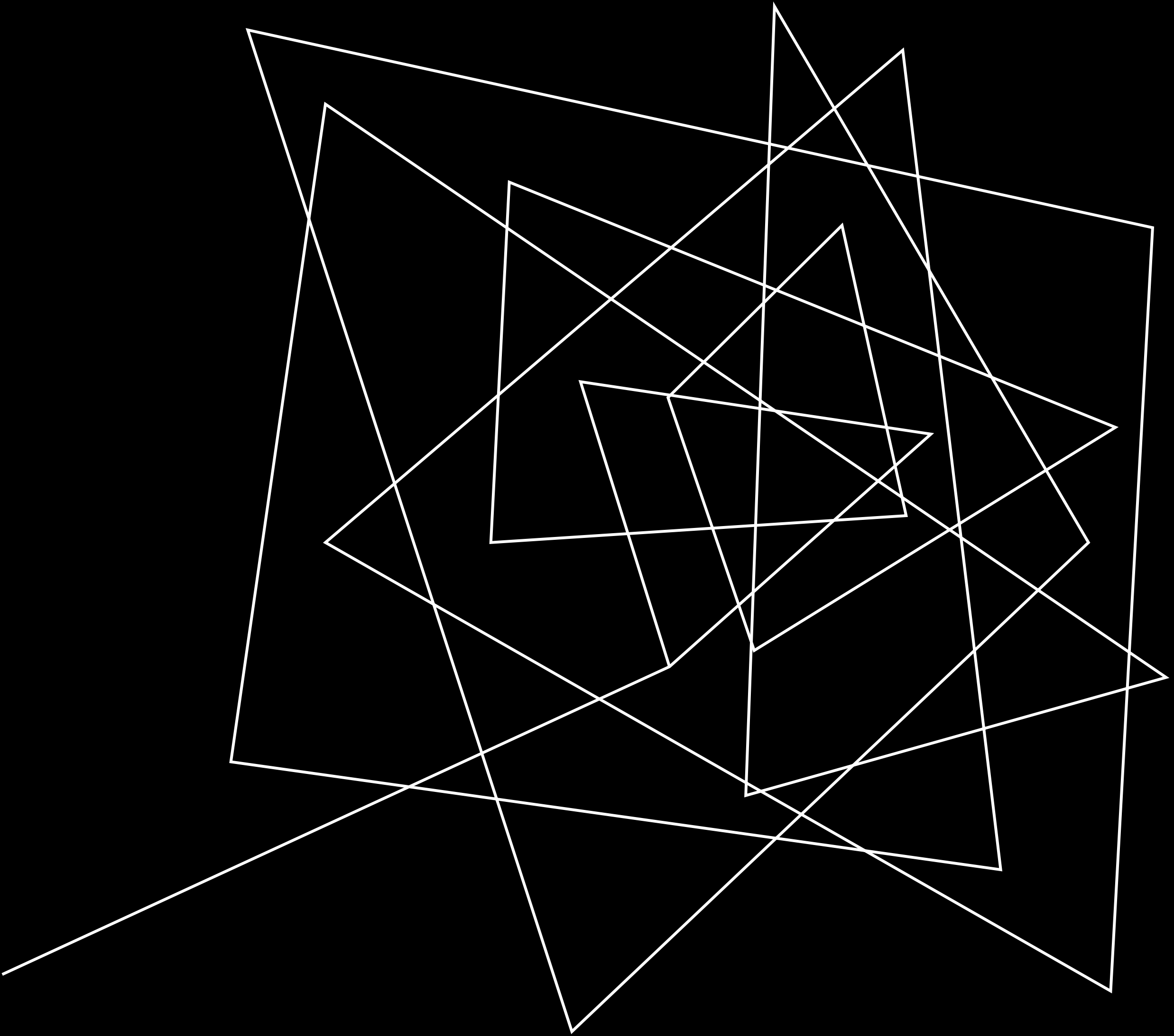
DATA SCIENCE PORTFOLIO

John Lingo

INTRODUCTION

I have been performing data mining, data analysis, and data visualization tasks for over 15 years as part of my role leading a contract engineering team that is responsible for developing, optimizing, and maintaining embedded firmware designs for high volume commercial products.



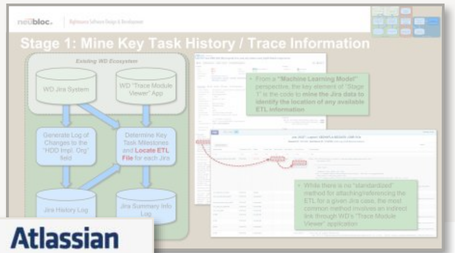
SAMPLE PROJECT – MACHINE LEARNING

“Demonstration of Feasibility” project to utilize Machine Learning techniques to direct firmware failure instances to the appropriate “firmware component team” for resolution

MACHINE LEARNING MODEL FOR FIRMWARE ISSUE TRIAGE - METHODOLOGY

Data Mining

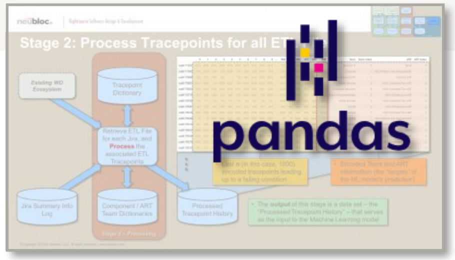
Extracted data from client's Jira system and custom cloud-based databases, using Python APIs



Data Processing & Cleaning

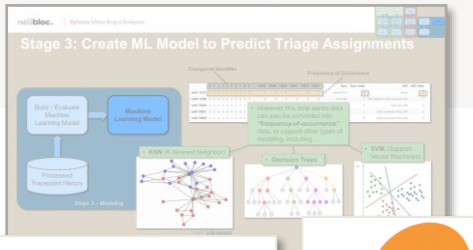
Created Python code to extract the last 1000 "trace points" leading up to a failure condition

"Curated" the input data set to eliminate cases where the needed information was unavailable or incorrect



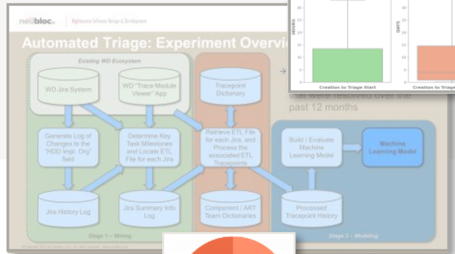
Data Analysis & Model Creation

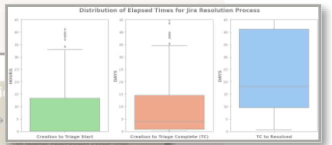
Utilized multiple Machine Learning models (**LSTM, KNN, Decision Tree, SVM**) to allow a "Component Team assignment" to be made based on a given input "failure signature"



Data Visualization

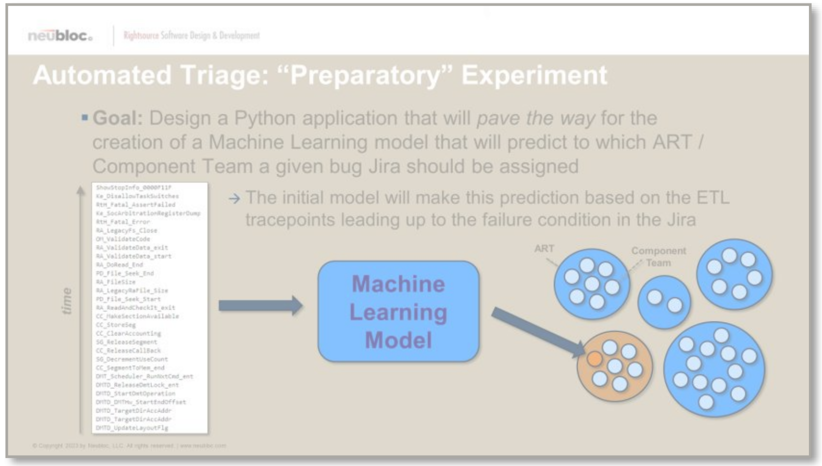
Created a client presentation that included numerous graphical depictions of the current / proposed processes and their effectiveness





Note: Some information above has been intentionally obscured to ensure protection of client intellectual property

MACHINE LEARNING MODEL FOR FIRMWARE ISSUE TRIAGE - RESULTS



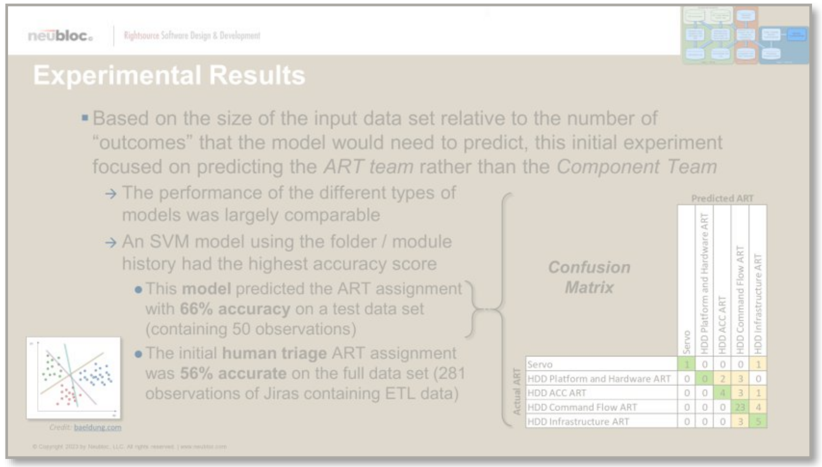
neübloc. | Rightsource Software Design & Development

Automated Triage: "Preparatory" Experiment

- **Goal:** Design a Python application that will *pave the way* for the creation of a Machine Learning model that will predict to which ART / Component Team a given bug Jira should be assigned
  - → The initial model will make this prediction based on the ETL tracepoints leading up to the failure condition in the Jira

```
ShowStopInfo_0000F11F
Ke_DisallowTaskSwitches
RtH_Fatal_AssertFailed
Ke_SocArbitrationRegisterDump
RtH_Fatal_Error
RA_LegacyFs_Close
OM_ValidateCode
RA_ValidateData_exit
RA_ValidateData_start
RA_DoRead_End
PD_File_Seek_End
RA_FileSize
RA_LegacyRaFile_Size
PD_File_Seek_Start
RA_ReadAndCheckIt_exit
CC_MakeSectionAvailable
CC_StoreSeg
CC_ClearAccounting
SG_ReleaseSegment
CC_ReleaseCallBack
SG_DecrementUseCount
CC_SegmentToMem_end
DMT_Scheduler_RunNxtCmd_ent
DMTD_ReleaseDmtLock_ent
DMTD_StartDmtOperation
DMTD_DMTHw_StartEndOffset
DMTD_TargetDirAccAddr
DMTD_TargetDirAccAddr
DMTD_UpdateLayoutFlg
```

Study demonstrated that the client's existing test process already generated sufficient "failure artifacts" to allow a Machine Learning model to be created and utilized for at least the initial assignment of a given failure to a "high-level team" for further investigation



neübloc. | Rightsource Software Design & Development

Experimental Results

- Based on the size of the input data set relative to the number of "outcomes" that the model would need to predict, this initial experiment focused on predicting the *ART team* rather than the *Component Team*
  - → The performance of the different types of models was largely comparable
  - → An SVM model using the folder / module history had the highest accuracy score
    - This **model** predicted the ART assignment with **66% accuracy** on a test data set (containing 50 observations)
    - The initial **human triage** ART assignment was **56% accurate** on the full data set (281 observations of Jiras containing ETL data)

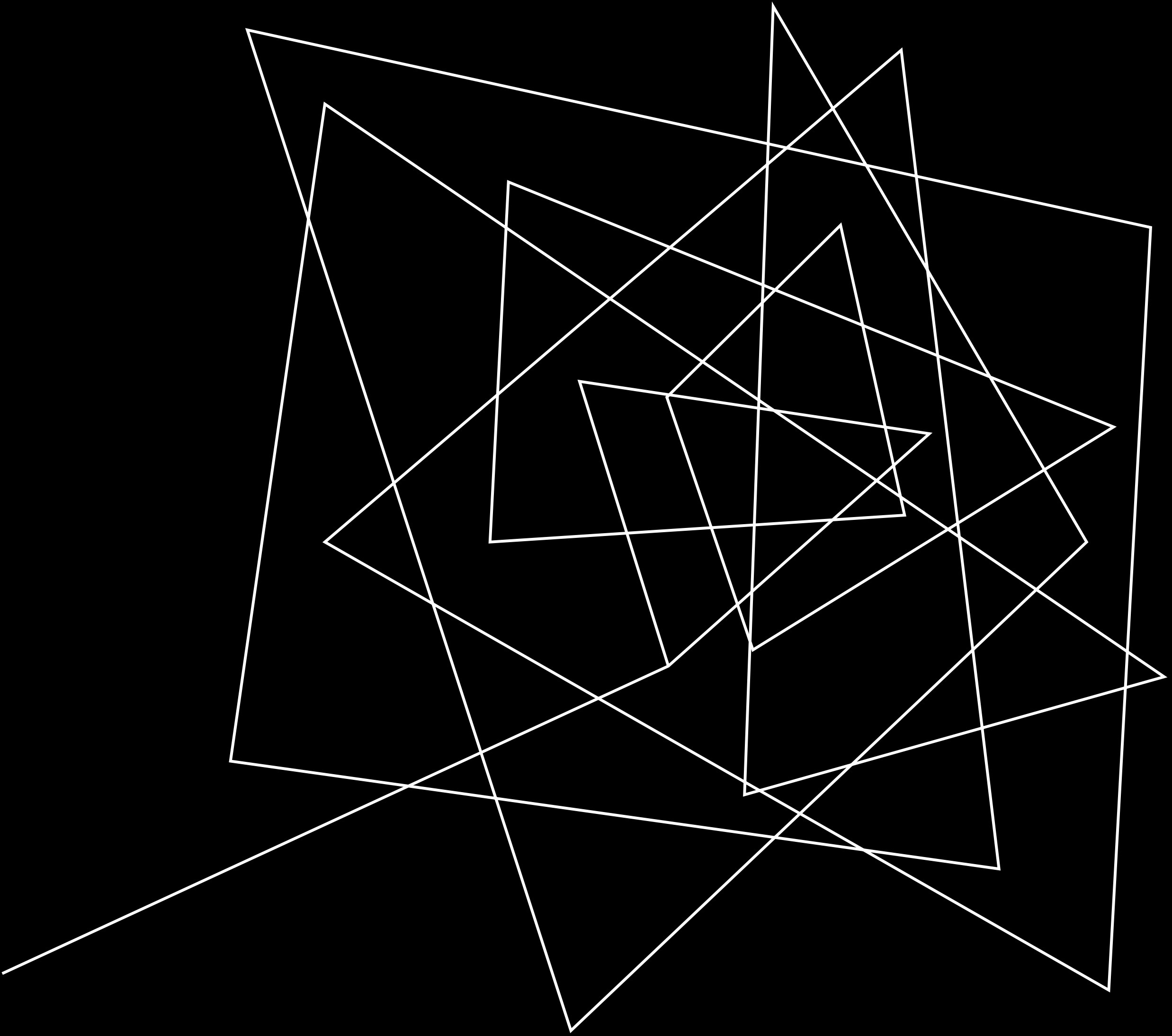
Confusion Matrix

Actual ART \ Predicted ART	Servo	HDD Platform and Hardware ART	HDD ACC ART	HDD Command Flow ART	HDD Infrastructure ART
Servo	1	0	0	0	1
HDD Platform and Hardware ART	0	0	2	3	0
HDD ACC ART	0	0	4	3	1
HDD Command Flow ART	0	0	0	23	4
HDD Infrastructure ART	0	0	0	3	5

Credit: baeldung.com

Study demonstrated that the accuracy of such a prediction, even with limited training data, *already exceeded the accuracy of the client's existing human-driven triage process*

Study also recommended numerous "next steps" for further improving on the accuracy of the model

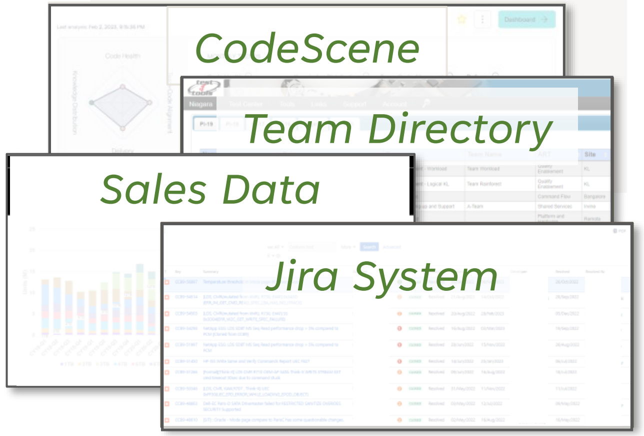


SAMPLE PROJECT – DATA MINING / ANALYSIS

Study of the relative cost of firmware development at three “lower cost” development sites

STUDY OF FIRMWARE DEVELOPMENT COST - METHODOLOGY

Data Mining	Data Cleaning	Data Analysis	Data Visualization
Extracted all required data from available client databases / systems, to assure client of study's objectivity	Utilized numerous "filters" to eliminate "outlier" data points Performed multiple actions to account for "missing" information as accurately as possible	Utilized both Excel and Python to aggregate and process this data	Presented results to the client in predominantly graphical format, using a PowerPoint slideshow



neübloc.

Part 2 Methodology *(continued)*

- Data from each Jira query was exported to a spreadsheet that calculated/graphed how many "QSR gating" Jira cases each team had open in the weeks leading up to QSR achievement

neübloc.

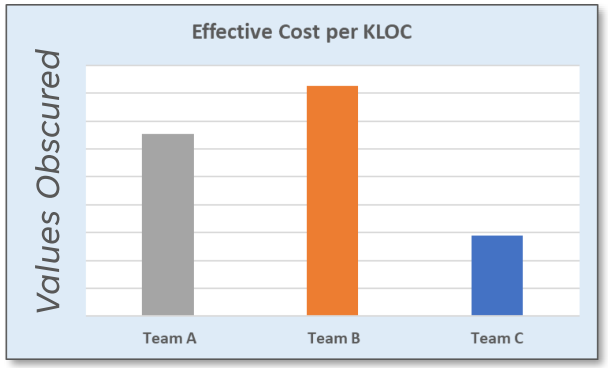
Part 2 Methodology / Observations

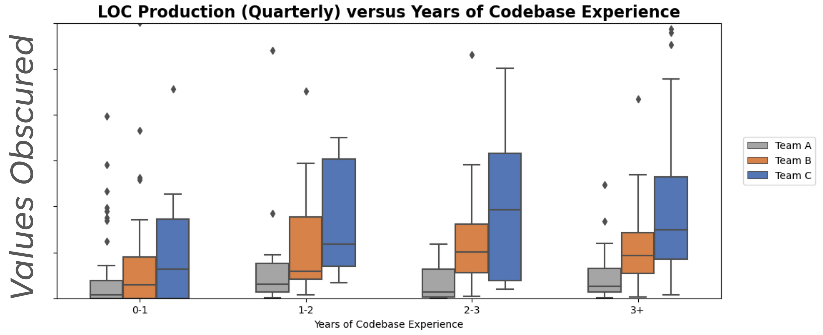
- The data from each of the three drive programs (four QSR dates) studied was then aggregated using a normalized QSR milestone:

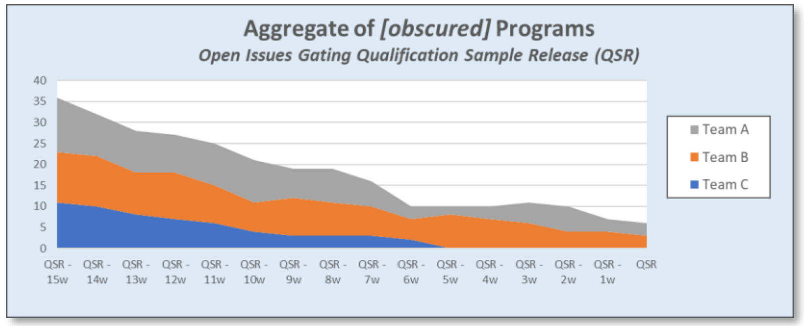
- 15 weeks ahead of the eventual QSR date, each of the three teams had (on average) a comparable number of open issues

Note: Some information above has been intentionally obscured to ensure protection of client intellectual property

STUDY OF FIRMWARE DEVELOPMENT COST - RESULTS





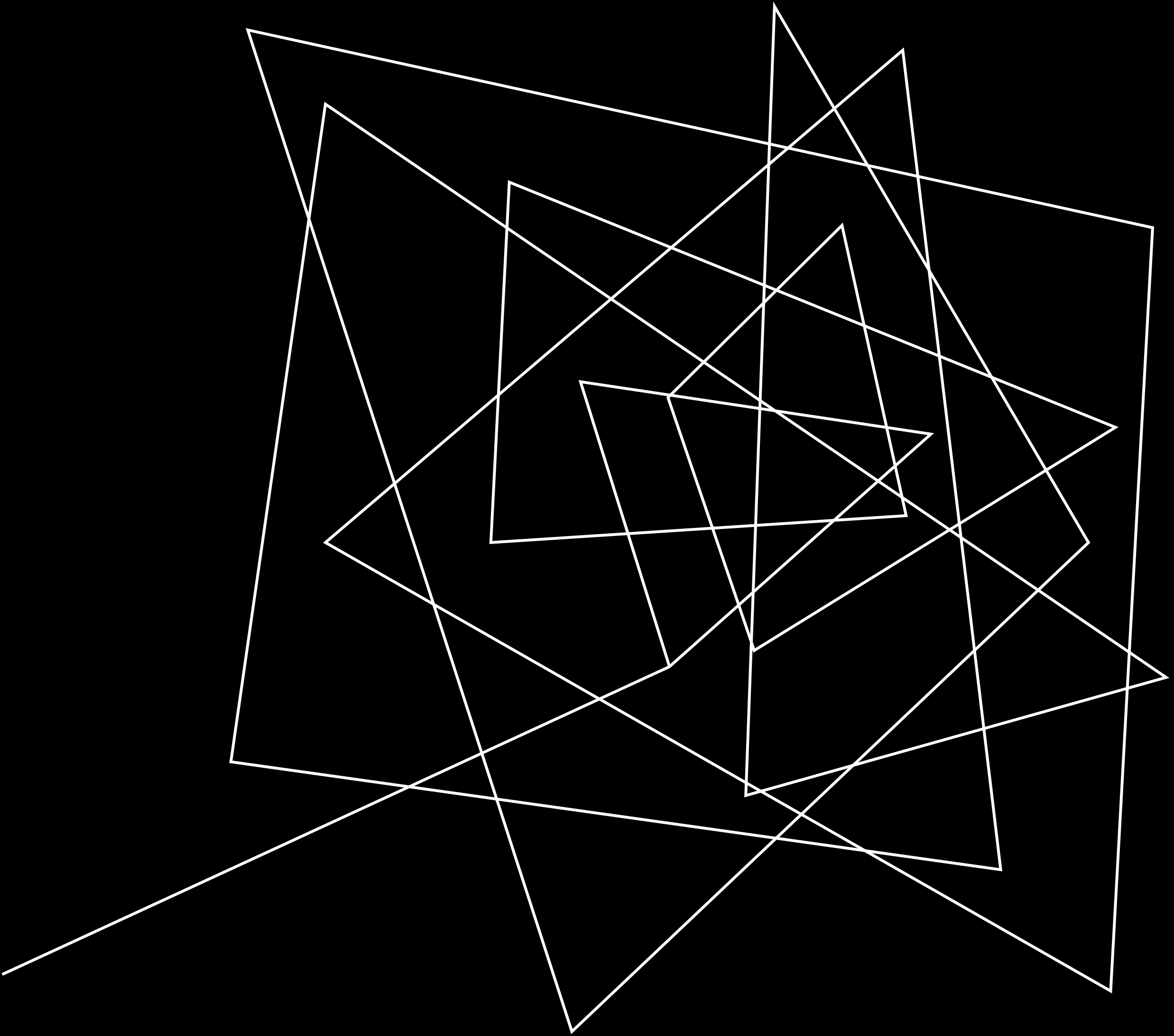


Study demonstrated that "Team C" was producing code at a significantly lower overall cost to the client (even though the "Team C" designers had the highest average yearly cost to the client on an *individual* salary basis).

Study demonstrated that the higher productivity of "Team C" members existed even shortly after the time a designer joined the team, and this productivity advantage persisted over time.

Study identified that across multiple recent client products, "Team C" had resolved all of their issues that were gating customer qualification samples significantly sooner than had the other two teams.

The study also translated this "earlier delivery" into a potential cost savings figure for the client ("if all teams were performing at the level of Team C").

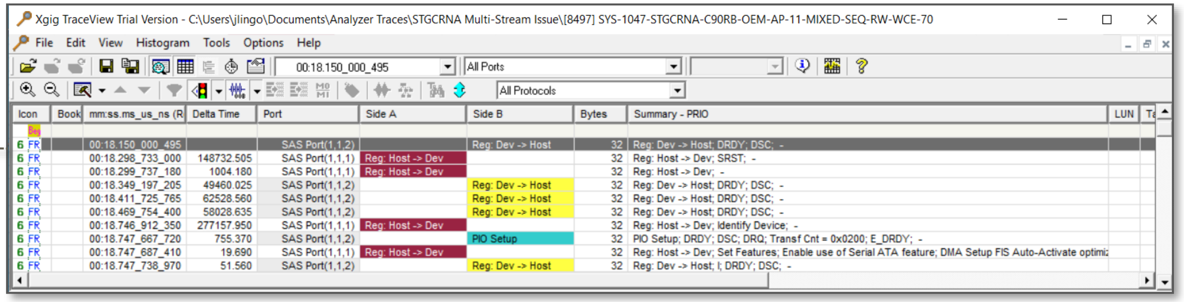


SAMPLE PROJECTS – DATA VISUALIZATION

The following projects utilized MS Visual Basic and the ChartDirector library to support visual analysis of data related to storage device performance, for the purpose of issue resolution and product optimization

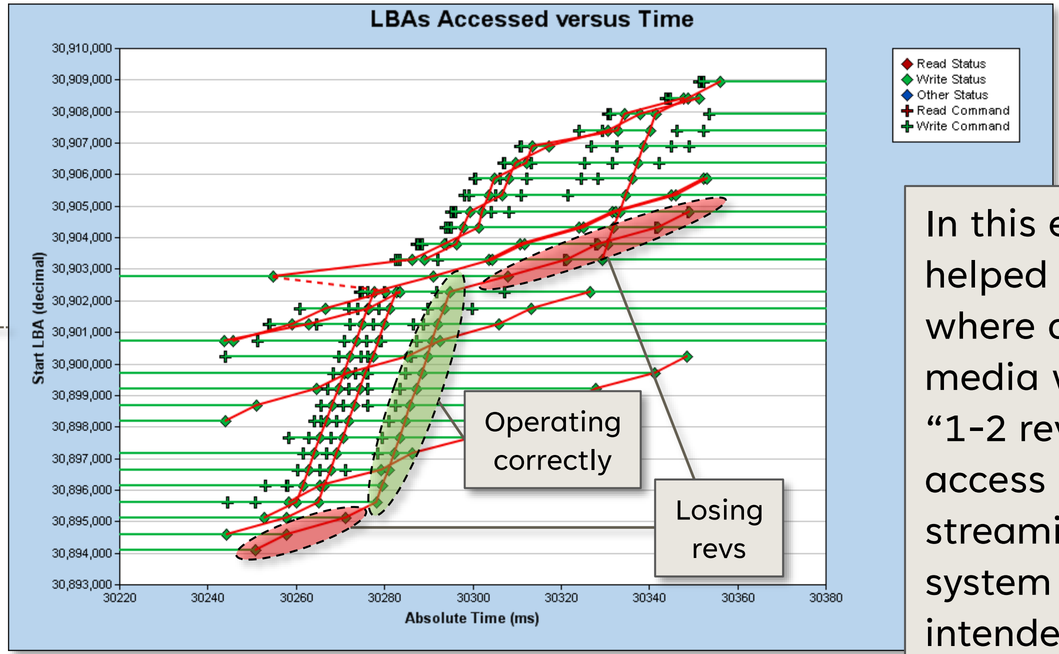
VISUALIZATION PROJECT 1: BUS TRACE VIEWER / ANALYZER

Problem: 3rd party bus trace viewer did not provide sufficient insight to allow designers to identify / understand storage device performance issues


Xgig TraceView Trial Version - C:\Users\jlingo\Documents\Analyzer Traces\STGCRNA Multi-Stream Issue\[8497] SYS-1047-STGCRNA-C90RB-OEM-AP-11-MIXED-SEQ-RW-WCE-70
File
Edit
View
Histogram
Tools
Options
Help
00:18.150_000_495
All Ports
All Protocols

Icon	Book	mm:ss.ms_us_ns (R	Delta Time	Port	Side A	Side B	Bytes	Summary - PRIO	LUN	Ta
Beg										
6 FR		00:18.150_000_495		SAS Port(1,1,2)		Reg: Dev -> Host	32	Reg: Dev -> Host; DRDY; DSC; -		
6 FR		00:18.298_733_000	148732.505	SAS Port(1,1,1)	Reg: Host -> Dev		32	Reg: Host -> Dev; SRST; -		
6 FR		00:18.299_737_180	1004.180	SAS Port(1,1,1)	Reg: Host -> Dev		32	Reg: Host -> Dev; -		
6 FR		00:18.349_197_205	49460.025	SAS Port(1,1,2)		Reg: Dev -> Host	32	Reg: Dev -> Host; DRDY; DSC; -		
6 FR		00:18.411_725_765	62528.560	SAS Port(1,1,2)		Reg: Dev -> Host	32	Reg: Dev -> Host; DRDY; DSC; -		
6 FR		00:18.469_754_400	58028.635	SAS Port(1,1,2)		Reg: Dev -> Host	32	Reg: Dev -> Host; DRDY; DSC; -		
6 FR		00:18.746_912_350	277157.950	SAS Port(1,1,1)	Reg: Host -> Dev		32	Reg: Host -> Dev; Identify Device; -		
6 FR		00:18.747_667_720	755.370	SAS Port(1,1,2)		PIO Setup	32	PIO Setup; DRDY; DSC; DRQ; Transf Cnt = 0x0200; E_DRDY; -		
6 FR		00:18.747_687_410	19.690	SAS Port(1,1,1)	Reg: Host -> Dev		32	Reg: Host -> Dev; Set Features; Enable use of Serial ATA feature; DMA Setup FIS Auto-Activate optimiz		
6 FR		00:18.747_738_970	51.560	SAS Port(1,1,2)		Reg: Dev -> Host	32	Reg: Dev -> Host; I; DRDY; DSC; -		

Solution: Created a tool to process the trace to display *time* and *media location* (LBA = Logical Block Address) data for key (Read / Write) commands, including identification of streams of sequential media accesses (see red lines at right)



In this example, the tool helped to identify cases where accesses of rotating media were experiencing “1-2 revolution gaps” in access time, instead of streaming data to the host system continuously as intended

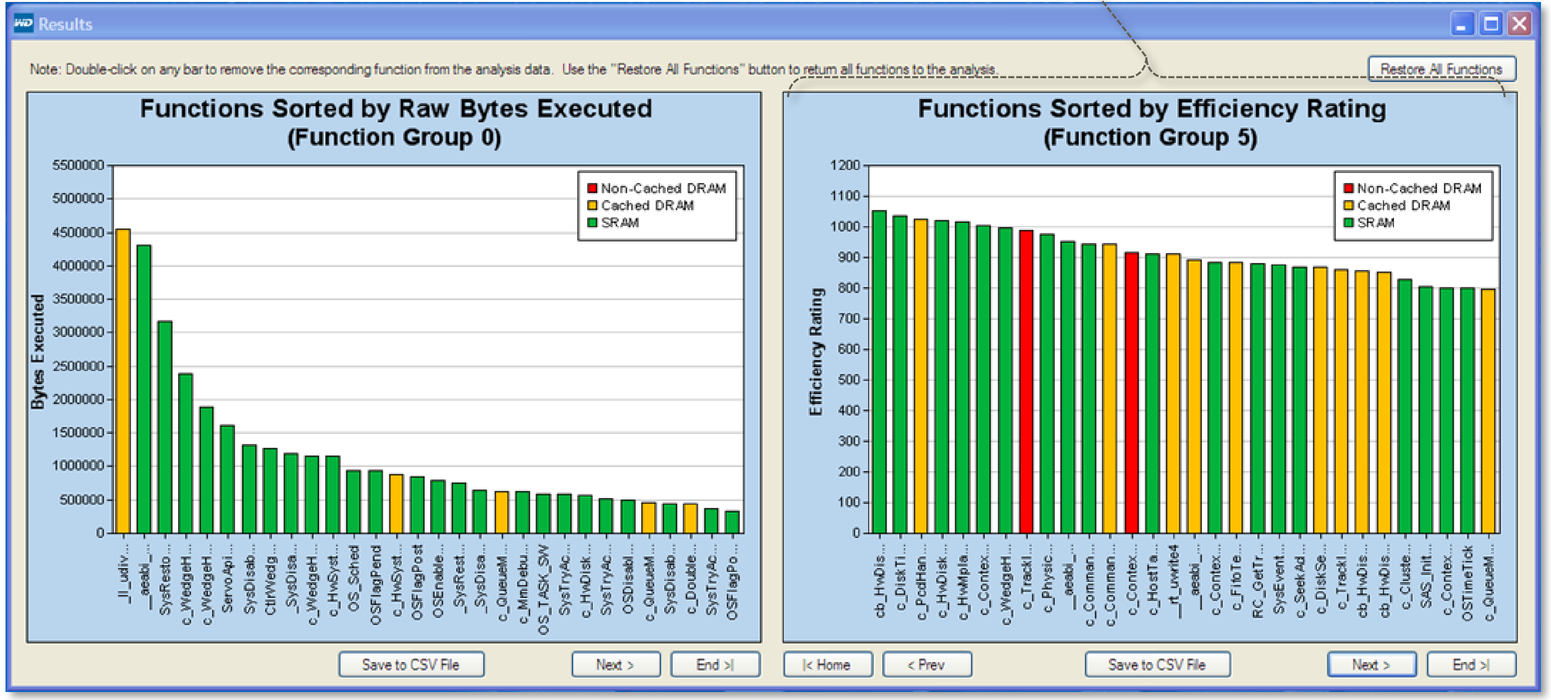
VISUALIZATION PROJECT 2: MEMORY MAP OPTIMIZATION

Problem: Embedded systems typically operate in "memory-constrained" environments, where it is important to ensure that the software functions that have the greatest impact on device performance are located in the highest speed memory (in this case, SRAM)

The "Efficiency Rating" is based on the cumulative number of bytes executed in a function, relative to that function's size

- Based on this metric, the red and yellow lines below indicate potential opportunities for further memory map optimization

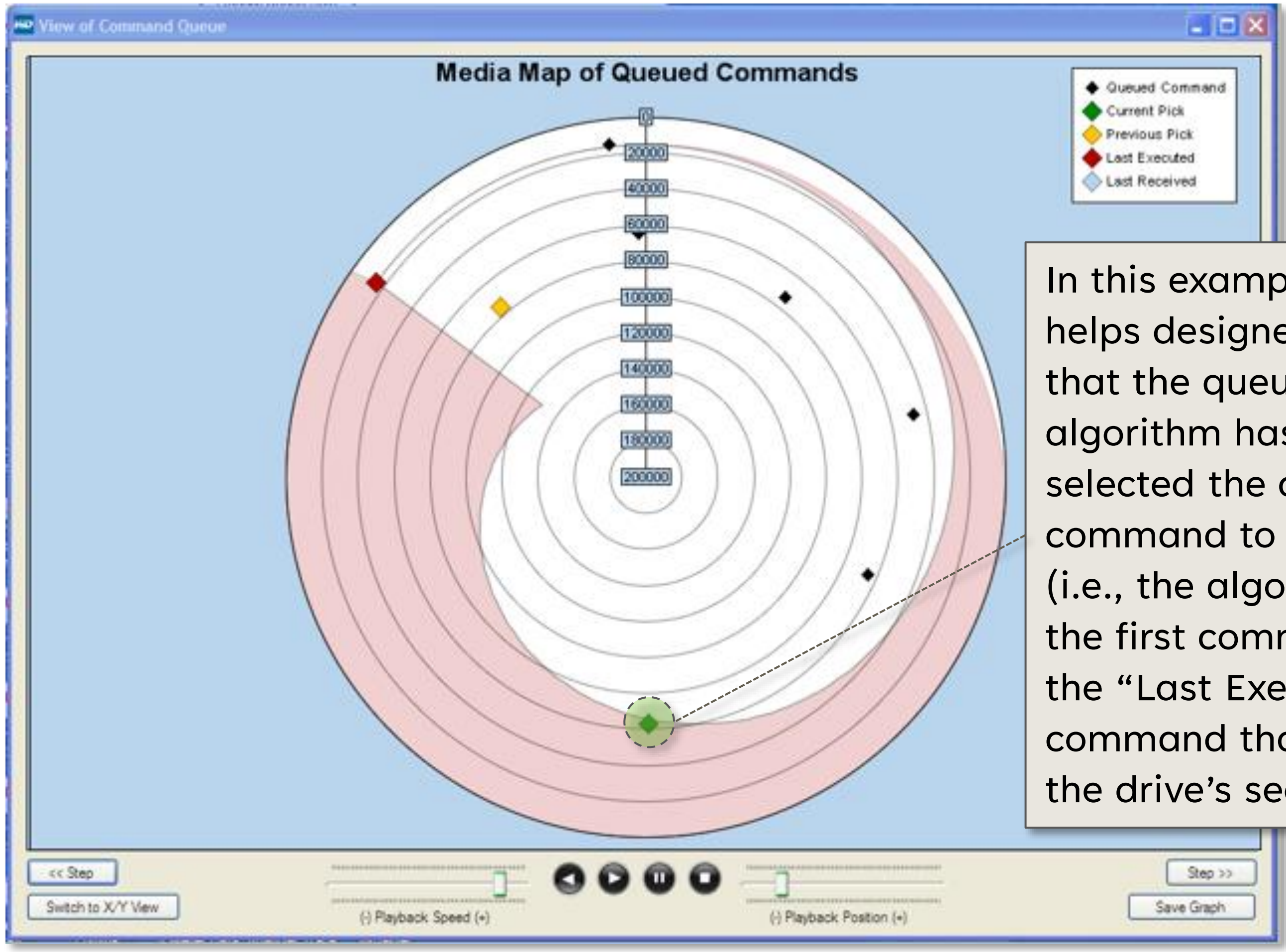
Solution: Created a tool to process an emulator trace of a critical code sequence, then use the memory map from the compile / link process to map this code to its associated memory type – this allows designers to identify opportunities for improving performance by changing the memory assignments of key functions


Note: Double-click on any bar to remove the corresponding function from the analysis data. Use the "Restore All Functions" button to return all functions to the analysis.

VISUALIZATION PROJECT 3: QUEUE SORT PLAYBACK

Problem: Disk drives select their "next access" from a queue of hundreds of pending commands, and it is often difficult for firmware designers to visualize the operation of this 2-dimensional process

Solution: Created a tool to play back a trace containing information on the state of the drive's command queue over time, including the drive's "seek profile" (see red envelope at right, which represents how far the drive's read/write heads can move as the media rotates away from the "Last Executed" position)

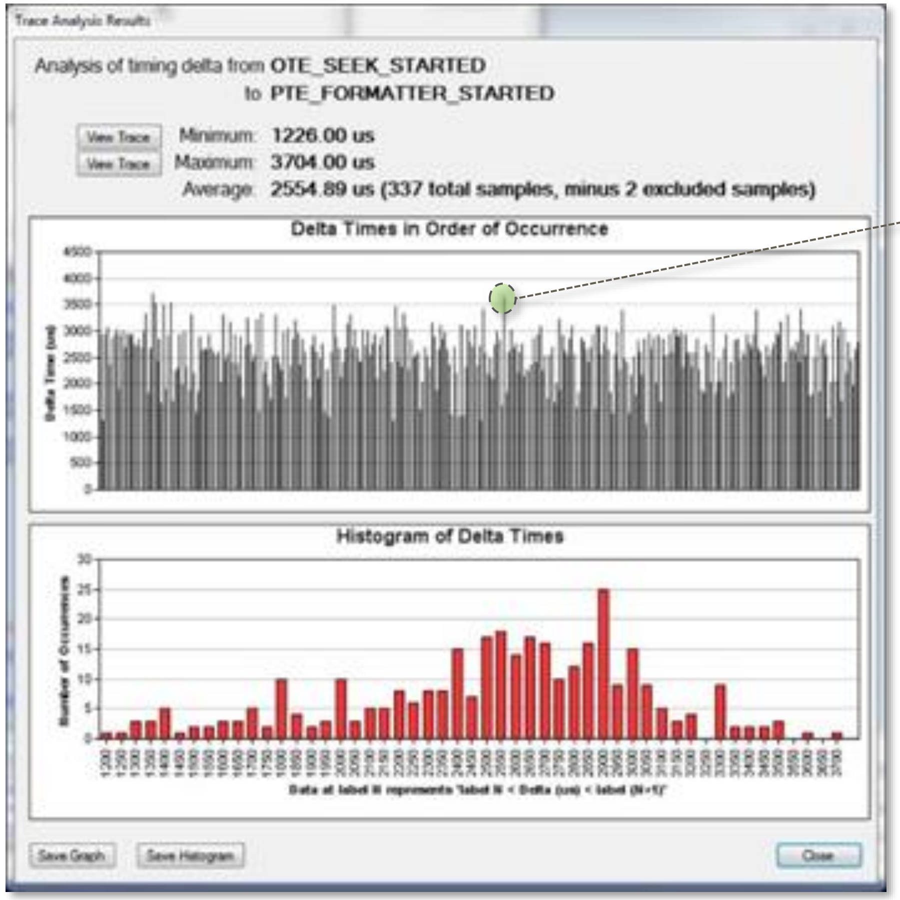


In this example, the tool helps designers recognize that the queue sort algorithm has indeed selected the correct command to execute next (i.e., the algorithm selected the first command after the "Last Executed" command that falls within the drive's seek profile)

VISUALIZATION PROJECT 4: TIMING OF KEY SEQUENCES

Problem: Embedded systems often have key code sequences that must be executed within a short period of time. While the firmware design may satisfy this requirement the majority of the time, designers must be able to identify the cause for cases in which the timing requirement is NOT met.

Solution: Created a tool to process an emulator trace, create a dictionary of all "timing points" contained within that trace, and allow users to select two timing points in order to receive information on the time delta between all occurrences of those two points in the trace



Analysis of timing delta from OTE_SEEK_STARTED
to PTE_FORMATTER_STARTED

Minimum: 1226.00 us
Maximum: 3704.00 us
Average: 2554.89 us (337 total samples, minus 2 excluded samples)

By clicking on a given bar on the "Order of Occurrence" chart, users can navigate directly to the start of the associated code segment in the trace

The "Histogram" portion of the display helps designers identify any modalities or other anomalies / outliers in the data

SUMMARY

The common theme of my projects over the past 15 years has been the improvement of business outcomes by:

- Extracting key information from complex data sets, and
- Building a supporting narrative around the resulting findings, and
- Conveying that information in an accessible (typically graphical) manner to my peers and management

THANKS FOR TAKING TIME TO VIEW THESE SAMPLES OF MY WORK

John Lingo

john.g.lingo@gmail.com

EVERY GREAT DEVELOPER YOU KNOW GOT THERE BY SOLVING PROBLEMS THEY WERE UNQUALIFIED TO SOLVE UNTIL THEY ACTUALLY DID IT.

- Patrick McKenzie