- Hello. My name is Jeff Miller. I am president and owner of Senoia Engineering Solutions – an independent cost engineering consultancy that focuses on electronic subassemblies used within an OEM's final products (specifically automotive and home appliances).

- We endeavor to help OEM clients find ways to reduce their costs thru design changes, manufacturing changes, and/or supplier sourcing changes.

- Today, I will be speaking to you about some cost analysis considerations when estimating and sourcing embedded software.

- The topic of embedded software is a HUGE area of study – hence, I won't be able to cover the entire topic in the short time allocated for today. Rather, my goal is to give you a high-level overview of some things to consider when evaluating it and estimating the cost.

INTRODUCTION

Jeff Miller
President and Owner
Senoia Engineering Solutions, LLC

Education

- B.S. Electrical Engineering – Kettering University (Flint, MI)
- M.S. Industrial Engineering – Purdue University (West Lafayette, IN)
- MBA – Rollins College (Winter Park, FL)
- Certificate in Executive Management – University of Notre Dame (South Bend, IN)

Experience

- 35 years of multifunctional experience within the electronics industry
  - *Automotive:* GM, Ford, Standard Motor Products, Panasonic
  - *Agriculture:* John Deere
  - *Major Appliance:* Whirlpool Corporation
- Over 12 years in cost engineering & VEVA of electronics & electrical assemblies.
- As a consultant: work(ed) with home appliance OEMs, automotive Tier 1s, cost engineering software companies, and management consulting firms.

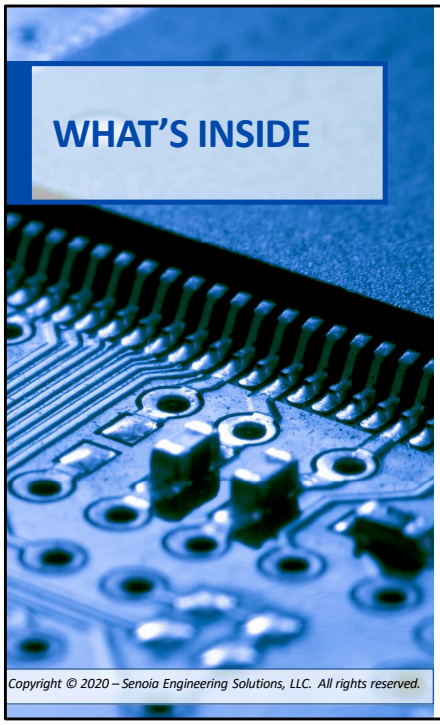
Other

- Board Member and Secretary / Treasurer – Society of Product Cost Engineering & Analytics (SPCEA): www.spcea.org
- Contributing author: "Realistic Cost Estimating for Manufacturing – 3rd Edition". Published by the Society of Manufacturing Engineers.

- A little about myself: I've held multifunctional roles (design engineering, purchasing, quality, manufacturing) within the electronics industry for over 35 years – of which, over 12 years of experience in cost engineering.

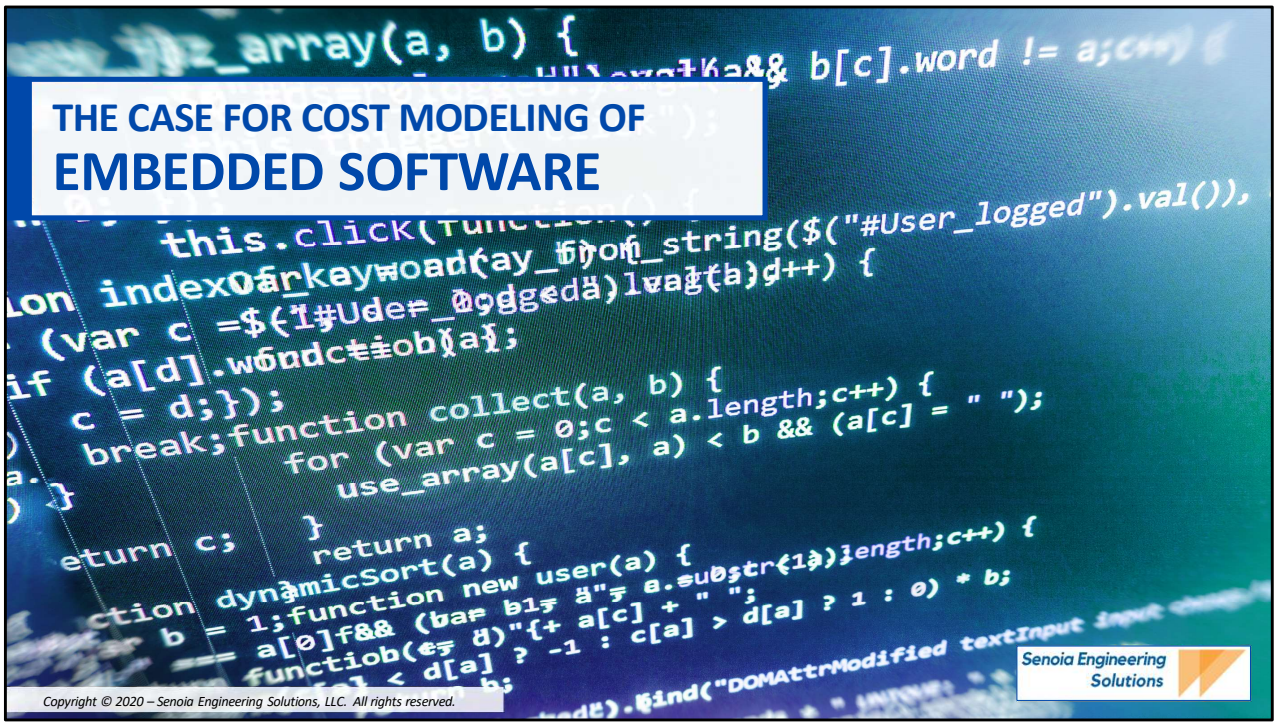
- I retired from Panasonic Automotive in April of 2019 and subsequently started my own consulting practice.

- I am proud to say that I am a Board Member and Secretary/Treasurer of a new professional society – SPCEA. Chris Domanski and I started this new society after listening to needs and concerns from last year's conference.

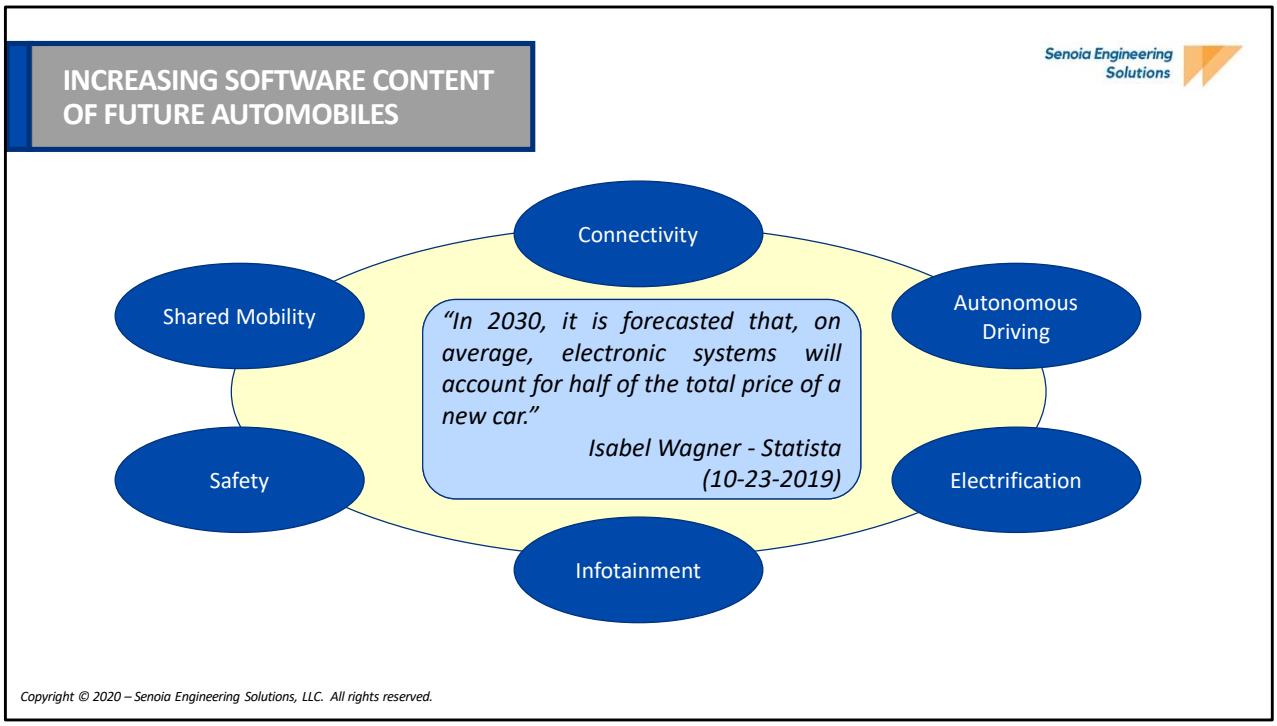


Senoia Engineering Solutions

My agenda is shown on this slide.



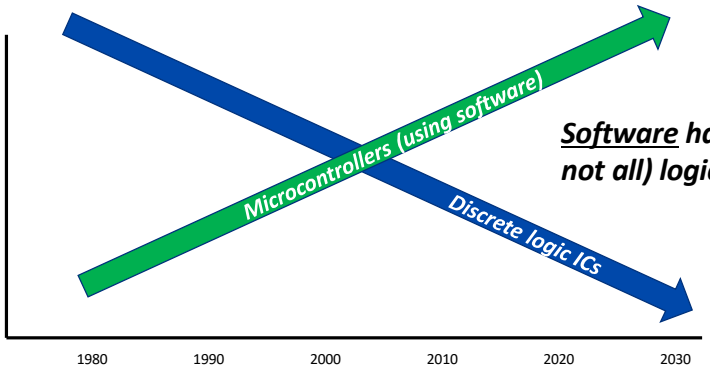
So, let's start. Why should we be concerned about the cost of embedded software ?



- It should pose as no surprise - unless you've been living under a rock for the past decade – that the electronics content of today's automobiles is skyrocketing.
- Advances in shared mobility, connectivity, autonomous driving, electrification, infotainment, and safety are driving an unprecedented level of electronics content (both hardware and software).
- I searched on Google to find a good summary comment – Isabel Wagner (of Statista) sums it up the best. Ironically, she will be presenting after me today and will be going into much more detail about this topic.

INCREASING SOFTWARE CONTENT OF FUTURE AUTOMOBILES

Senoia Engineering Solutions



***Software* has replaced hardware for most (if not all) logic function implementations.**

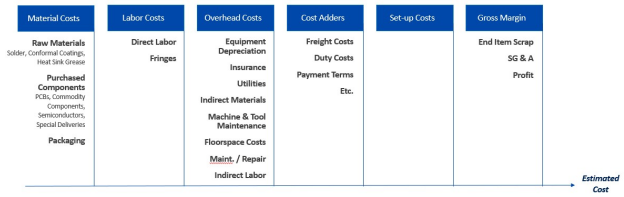
ALMOST ALL ELECTRONIC CIRCUIT ASSEMBLIES INCORPORATE A SOFTWARE ELEMENT

- The more senior EEs out there probably remember using small scale ICs to implement logic functions. Who remembers the SN7400 quad NAND gate IC ?
- As microcontroller technology evolved, programmable devices have developed more capability and have dropped in cost, replacing all those logic ICs with software. Today – even simple sensors have software integrated within them.
- Software content is a key cost driver within most (if not all) electronics.

CURRENT STATE WITHIN THE AUTO INDUSTRY

Senoia Engineering Solutions

HARDWARE



- Structured methodology in estimating cost.
- Design is usually defined (prints, BOMs, etc.)

SOFTWARE

- Initial specifications are vague and change frequently.
- Design (i.e. size of code) is usually not understood.
- Development costs are usually paid up front (in engineering budget) – not part of piece price.
- Cost estimating relies heavily on tribal knowledge.

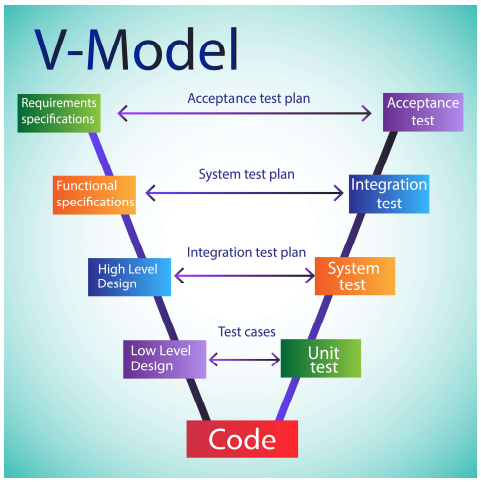
HOW CAN WE ESTIMATE THE COST OF EMBEDDED SOFTWARE WITH THE SAME DISCIPLINE AS WE ESTIMATE HARDWARE ?

- Many clients that I work with have very good, structured processes for estimating the cost of components.
- We analyze material costs, processing costs, overhead, etc. of hardware with a high level of precision and we “sweat the details”.
- When it comes to software, we don’t have that same level of discipline. Specifications are vague and there are frequent surprises by the cost. Tribal knowledge is still the dominant method.
- Software is usually paid for out of engineering development. Admittedly, cost engineers are generally focused on piece price – however, there are big savings to be had in development costs if we apply the same lens.
- How can we estimate the cost of embedded software as we do with hardware ?

In order to estimate the cost of software, we must understand the development process.

V-MODEL OF SOFTWARE DEVELOPMENT

Senoia Engineering Solutions



GENERAL DEVELOPMENT PROCESS:

- Understand requirements (features and functions)
- Divide project into major subsystems.
- Decide what functions can use 3rd party "building blocks" or re-use code.
  - Main sections: GUI, middleware, drivers, etc.
  - Specialized functions may have cost (i.e. map data, voice recognition, etc.
  - Typical infotainment project may have 8-10 third party blocks.
- Estimate Internal Effort
  - Integrating building blocks vs. writing unique code.
  - Estimate size of unique code.
  - The SW engineer wants to minimize the amount of new and unique code.
    - Proven code should have fewer quality issues.
    - Minimize cost of maintenance.
- Estimate Level of Testing

- There are many development processes in the software industry: V-model, waterfall, Agile, etc. For purposes of our discussion, let's talk about the V-model.

EFFECTS OF SAFETY AND SECURITY

Senoia Engineering Solutions

Domain	Safety Level				
Automotive (ISO 26262)	QM	ASIL-A	ASIL-B/C	ASIL-D	--
General (IEC-61508)	--	SIL-1	SIL-2	SIL-3	SIL-4
Aviation (DO-178/254)	DAL-E	DAL-D	DAL-C	DAL-B	DAL-A

Level of Inherent Safety Risk

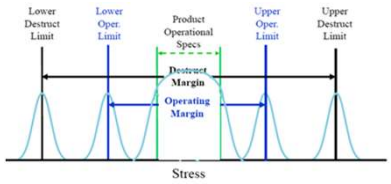
- Various industries have defined standards to govern safety and security.
- ISO 26262 defines a hazardous event in terms of Severity, Exposure, Controllability
- Effect on software development
  - Increased complexity to ensure safety and security.
  - Increased testing to validate software.
  - **INCREASED PROJECT COST**

- Safety and security requirements drive the cost of development due to how fault conditions are detected and managed – the more critical the system, the more effort needs to be expended to ensure meeting the safety and security requirements.
- Industries have safety / security standards specific to their industry.
- In automotive, ISO 26262 is the dominant safety standard, where QM and ASIL-A represent the lowest level of safety requirement, ASIL-D the highest.
- The higher the rating, the more care that has to be applied to how the software detects and manages fault conditions. This means more software complexity and more testing, which results in increased project cost.

TESTING – HARDWARE IN LOOP

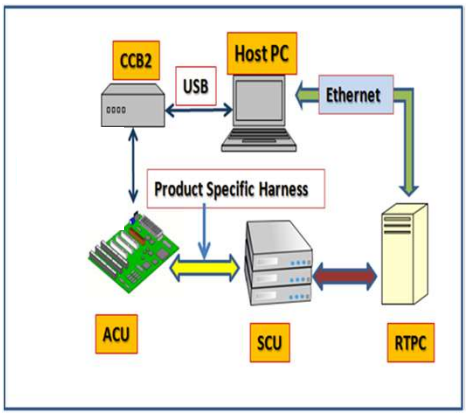
Senoia Engineering Solutions

Hardware-in-the-Loop (HIL) simulation is a technique that is used increasingly in the development and test of complex real-time embedded systems.



Key Attributes:

1. No need for physical system (digital twin)
2. Capability to simulate all scenarios (in spec, out of spec conditions)
3. Automatic and unattended regression test
4. Knowledge captured in the test cases (learning cycles)
5. Stable, repeatable, and accurate test results
6. Leverage component models (simulated models), test scripts and test cases across platforms and projects



AUTOMATED SYSTEMS (SUCH AS HIL) MUST BE USED FOR TESTING OF COMPLEX SYSTEMS

- Automated systems must be used to test for bugs in complex systems.
- Not enough just to test to specification. Software reaction to fault conditions must also be evaluated.
- Manual testing is not practical in order to catch all of the permutations and combinations of possible fault conditions.
- Obviously, more complex software requires for complex testing.
- Safety / security systems require even more.

COST DRIVERS

Senoia Engineering Solutions

FUNCTIONAL COMPLEXITY

Domain dependent:

- Infotainment
- Braking system
- Automatic parking
- …

CODE SIZE

- How to compare 2 sets of features between them, how to compare 2 projects?
- How to measure productivity?

LEVEL OF CODE REUSE

- How much code is new and unique versus adaptation of existing code

PROGRAMMING LANGUAGE

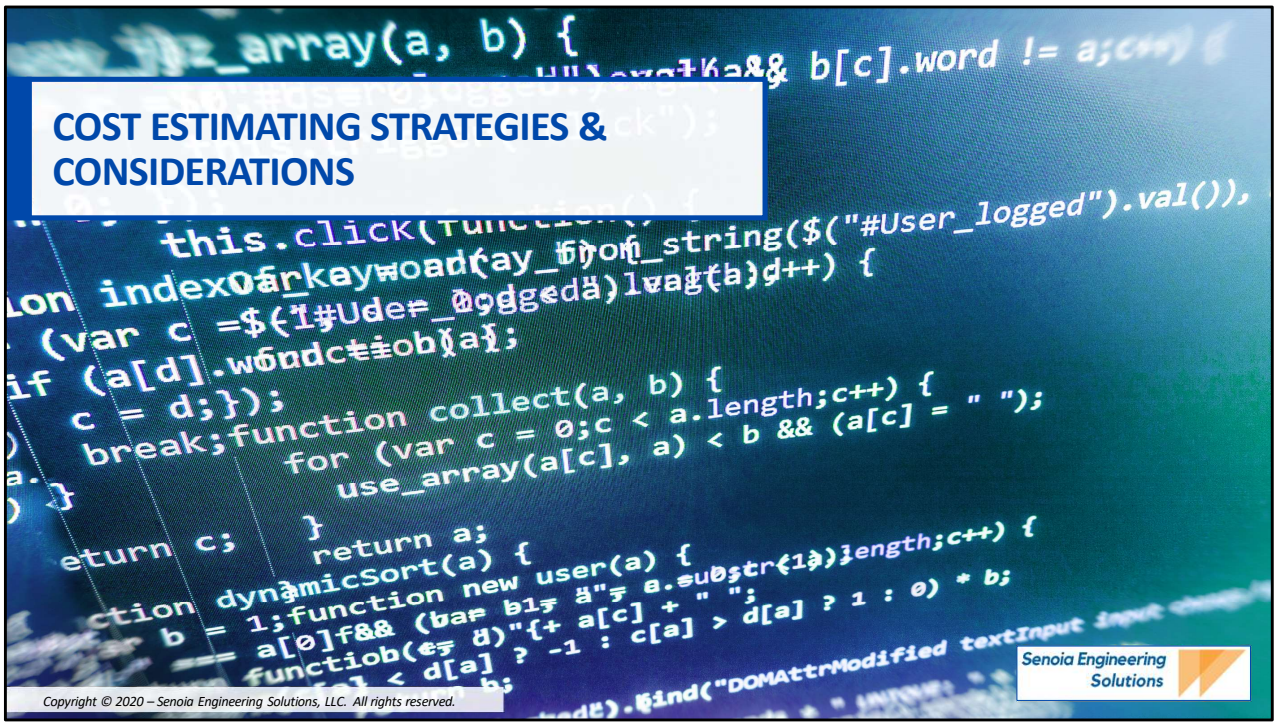
- C is frequently used for simple or more real time projects
- C++ , Java or C#, when lots of memory management is required or for more advanced functionalities

DEVELOPMENT TEAM PRODUCTIVITY

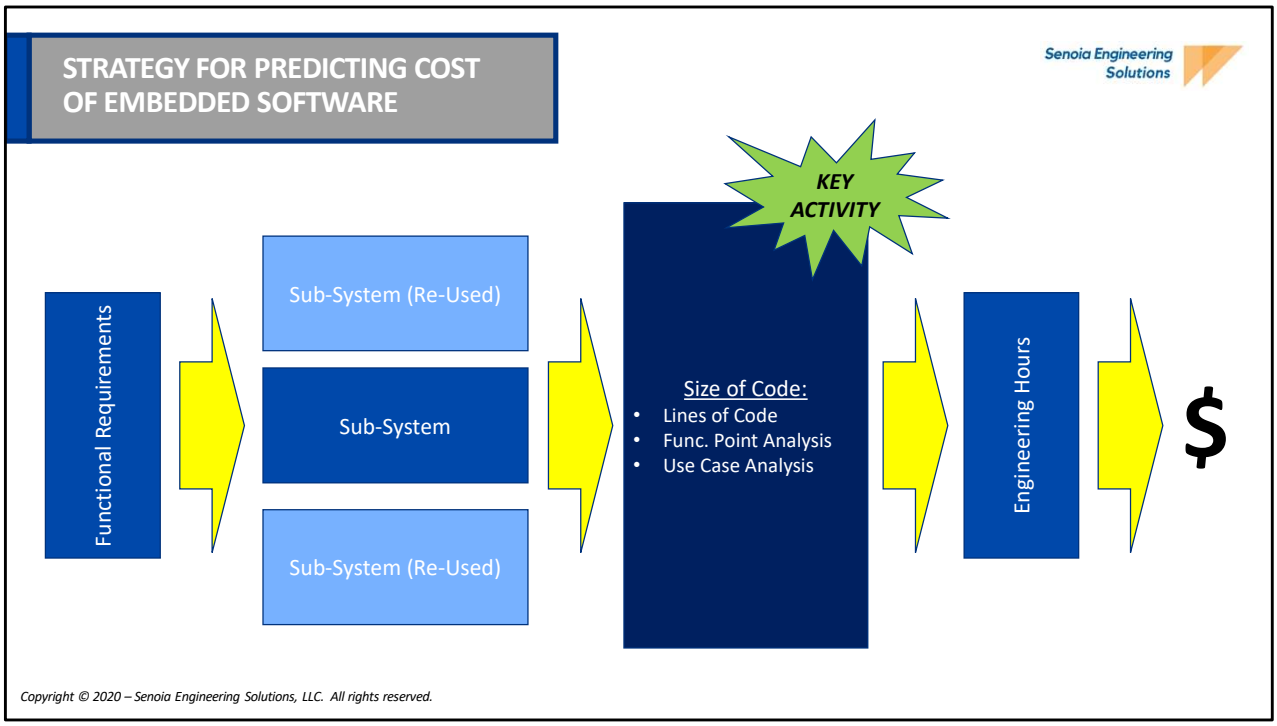
- From the maturity of processes point of view
- From the people experience and their domain knowledge point of view

LEVEL OF TESTING / CHANGE MANAGEMENT

- Cost drivers in code development include these items.
- Modern cost estimating software include these items as inputs to the cost model.



Now, let's talk about some strategies and considerations. . .



- The key is to translate the specification requirements into engineering hours.
- Functional requirements must be understood clearly, as this translates to understanding the size of the code.
- The size of the code can be determined in a variety of ways (i.e. lines of code, function point analysis, use case analysis). Regardless of what method is used, the size of code represents a KEY ACTIVITY !!

STRATEGY FOR PREDICTING COST OF EMBEDDED SOFTWARE

Senoia Engineering Solutions

TRIANGULATE BETWEEN THESE INFORMATION SOURCES

- I recommend this strategy for predicting the cost of embedded software.
- Basically, we want to triangulate between data from multiple quotes, internal SW engineering experience, and predictive cost models – just like we do for hardware and physical parts !!
- Next, I'll deep dive each of these . .

CONSIDERATIONS – MULTIPLE QUOTES

Senoia Engineering Solutions

HOW WELL ARE FUNCTIONAL REQUIREMENTS UNDERSTOOD ?

- Initial terms and conditions are important.
- Transparency
- There are very few automotive-centric S/W companies.

STATEMENT OF WORK (SoW)

- First SoW is usually not good - misses many things.
- Anticipate that there will be multiple changes to SoW
- Capture lessons learned from early projects so that these can be carried over to other projects.
  - Minimize the learning curve.

"MARKET RATE" OF COST DRIVERS:

- Expertise: Jr. Developer, Sr. Developer, Project Manager, etc.
- Engineering Rates:
  - Example: $ 70 / hour for base level engineers, $ 200 / hour for senior level engineers or high-demand specialists.
- Estimated man-hours to execute project.
  - Example: 1000 lines of code per engineer per month.

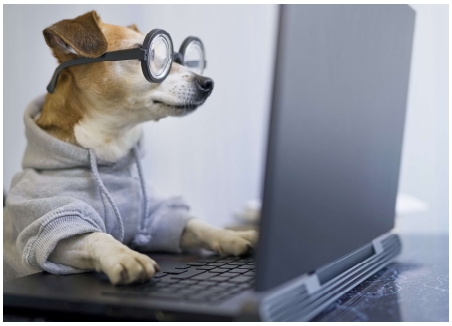
- In order to use data from multiple quotes, a certain level of groundwork is required. Initial terms and conditions and a good SoW are important in order to have the level of transparency and granularity to make the quote data useful.
- It always surprises me that suppliers hide behind "confidentiality" when it comes to reviewing source code.
- Software changes frequently. Be prepared for many changes to the SoW.
- Understand market rate of skills: levels of expertise, rates per hour, etc.

CONSIDERATIONS – COMPARE TO INTERNAL SOFTWARE ENGINEERING DATA

Senoia Engineering Solutions

CROSS-REFERENCE WITH INTERNAL SOFTWARE DEV. TEAMS

- If available – use it !!
  - Independent technical reference point.
  - Consistency across organization
- If not, develop the capability !!
- Characteristics of an ideal cost engineering candidate for software:
  - IT or SW developer who is interested in something new.
    - Preferred: internal candidate who is familiar with functionality of existing products and software.
  - Financial acumen / good analysis skills.
  - Good communication skills - must be able to communicate to leadership.



- Large companies (such as the Big 3 automakers and Tier 1s) have internal software engineering teams. Use this expertise as a cross reference when evaluating a project. If you have this resource available, USE IT !!
- If not available, you should strongly consider developing a cost engineer with software focus. Ideal candidate.

CONSIDERATIONS – COST MODELS

Senoia Engineering Solutions

USE COST MODELS TO CAPTURE LESSONS LEARNED

- Commercially available software is a good repository for lessons learned.

UNDERSTAND THE LEVEL OF CODE RE-USE

- Easiest to estimate: COTS (Commercial Off-The-Shelf) software / off-the-shelf with added extension(s)
- Most difficult to estimate: new, clean sheet project with no re-use.
- 3rd party code / data is very commoditized.
  - Typically use "market price" as part of the estimate.
  - Many building blocks (application programing interfaces or APIs) are available for free or for minimal cost.
    - RTOS, Linux, Android, Apple CarPlay, drivers for SoC.

USE OF A "MAJOR FEATURE" LIST

- Itemized list of software features and associated cost.
- Allows determination of ROI by feature – evaluate potential for income vs. cost of developing the function.

- Cost modeling software is an ideal repository for lessons learned.
- It is important to understand the level of code re-use. If most of the code is off-the-shelf or pre-existing, then it's easier to estimate. The hardest project to estimate is one in which you are starting from scratch.
- Keeping an itemized list of software features and cost allows for ROI determination – useful if making tradeoff decisions.

IN SUMMARY . . . .

Senoia Engineering Solutions

YOU CAN ESTIMATE THE COST OF EMBEDDED SOFTWARE !!

- Have a good understanding of functional requirements.
- Understanding size of code is key.
- Build internal knowledge and expertise.
- Capture data and lessons learned from projects for use with future projects.

SPCEA
SOCIETY OF PRODUCT COST ENGINEERING & ANALYTICS

SOCIETY OF PRODUCT COST ENGINEERING & ANALYTICS (SPCEA)

- Special focus group for electronics & embedded software has formed (October, 2020):
  - Adapt methodologies from other industries to automotive applications.
  - Establish standards for estimating embedded software.
  - Coordinate with other professional societies.

WWW.SPCEA.ORG

- You CAN estimate the cost of embedded software.
- SPCEA is a new professional society. We're forming a special focus group to drive this standardization effort – join us !!

Q & A

Questions ?

Thank you. It was a pleasure speaking with you in today's conference.