

Embedded in Thin Slices

Build an Embedded Systems Consulting Company (Part 7)

Writing the Proposal

In the final part of this article series, Bob looks at the most critical task for success once you have your foot in the door with a client: Creating a winning proposal.

By
Bob Japenga

A few years back, we were bidding on a new phase of an existing project. We designed all of the hardware and software for this device. This phase was to incorporate connectivity to the Internet in the device and develop the back end. The purpose of the back end was to provide an interface to be used by both manufacturing and end users. Since the advantages of Internet connectivity had not been proven in the marketplace for this type of product, this phase was being developed incrementally. By-the-way, since that time, this feature has been the key differentiator in driving up sales. We got the contract for the embedded portion. We developed a quick and easy back end as part of the pilot program. This simple back end was designed for engineers in manufacturing and in the field. Once the pilot devices hit the market, they generated a lot of interest and resulting sales. Now was the time to make the back end robust and fully user friendly.

We developed a full proposal for this work. The proposal included the three items that we believe are mandatory for all proposals that we send to a customer. Those are: a) Statement of Work (SOW); b) Preliminary Specification; c) Assumptions that are made in the proposal. Our reputation was—and still is—stellar with this customer. Our price was actually lower than our competitor's. Our competitor had developed scores of similar back ends for IoT devices. We had developed just three. The competitor's proposal did not contain a SOW or a specification. The proposal basically said: "We are experts. We have done this before. We know

what you want." The first two statements were accurate. We did not get the job.

Now, almost three years later, we have been given the chance to do the job over. You can imagine why. Starting a project without a clear idea about what you are supposed to do (the SOW); how you are going to do it (the Specification); and what assumptions are being made in the SOW and the Specification is a recipe for disaster.

Let's look at each of these three to help you avoid these pitfalls.

STATEMENT OF WORK (SOW)

A SOW defines the scope of the work to be performed; the specific tasks that will be performed; the milestones; the deliverables; and the schedule. With as much detail as you have time to create, you need to write down all of these and include them in the proposal. Let's look at each of these in a little more depth.

Specific Tasks: During an embedded systems project, both you and the customer will perform certain tasks. The SOW should contain both your tasks and your customer's tasks. Defining what your customer will do is as important as defining what you are going to do. The task description should be as specific as possible. For example: Create a storyboard of the user interface including—but not limited to—icons, approximate font size, locations of text/icons and user interaction.

When creating the list of specific tasks for the proposal, you need to balance necessary and unnecessary detail. I find that if I ask myself these two questions when creating the

list of specific tasks, we don't have to list all the tasks that will be performed: a) If this was not defined at this level of detail what could go wrong? b) What tasks will I perform that are missing from the list? All that said, in costing the proposal, the more detail provided, the more accurate you can be in your estimate. Generally, we list those tasks that we know (or suspect) the customer needs to know.

Responsibility: We often provide a list of all of the tasks in rows and then in columns provide the level of responsibility of each organization has to the specific task. The specific tasks can be footnoted to provide the level of detail without clogging up the table. For example, we might be responsible for the storyboard and the customer may have the responsibility to review it.

Schedule: Using a project management tool you can create a detailed schedule of the tasks with the dependencies. We recommend that you provide as much margin to this schedule as you can possibly get away with. We have never lost a contract because of schedule. Normally it starts a discussion with the client about how you can pull the schedule in. When scheduling tasks that are completed by someone other than yourself, you should make all subsequent tasks dependent upon that delivery.

Customer Deliverables: All deliverables should be included in the SOW by words or color coding. In the example that follows, we are using color coding and footnotes. What detail the deliverable will contain should be provided by referencing your own internal standard operating procedure or an outside authority. We use IEEE.

EXAMPLE SOW

Let's work through a specific example. Let's imagine that

we are going to create a device that would use an existing OBD2 reader and provide a digital readout to display specific engine parameters on a display that are either missing or broken on your car. (This is not a real project but it sounds interesting.) The customer wants the device to only display one parameter. Consider for example, my 2001 Taurus. It does not provide outside temperature on the dash. But the OBD2 interface provides intake temperature which is basically the same as outside temperature. What might a SOW look like for this project? Let's assume that the scope of this project is to take the design up through production.

Table 1 represents a detailed statement of work for this simple project. You may choose to limit what the customer actually sees in the proposal. But delineating all of the tasks internally during the proposal and assigning an estimated cost to them is critical for the success of your company.

THE SPECIFICATION

The specification is one of the most critical parts of any proposal. You might ask "Bob, how can I specify the product when I haven't done the requirements analysis?" You have to have done enough analysis so that you can put a boundary around the project and protect your company from financial black holes. The purpose of the specification within the proposal is to create clear boundaries. For our example project, there are many questions both at a system's level and at a software level. Here is how I would go about creating the specification in the proposal for this project:

- The device shall communicate to the OBD2 reader via Bluetooth 3.0 devices.

Task (Red indicates a deliverable)	Responsibility	Schedule	Cost (man-hours)
Create a Project Plan ¹	Us	1 day After receipt of order (ARO)	4
Perform Requirements Analysis	Us	2 days ARO	8
Create Use Case and User Interface storyboards ²	Them	5 days ARO	0
Perform a Requirements Review ³	Us and Them	5 days ARO	4
Create System Block Diagram	Us	7 days ARO	4
Create a Software Requirements Specification ⁴	Us	7 days ARO	8
Create a Preliminary HW Design	Us	10 days ARO	4
Create a Test Plan ⁵	Us	10 days ARO	2
Perform a Preliminary Design Review (PDR)	Us and Them	11 days ARO	2
Create a Detailed HW Design	Us	1 day after PDR	4
Perform a Critical Design Review (CDR)	Us and Them	5 days after PDR	4
1. Defined by MTI Standard Operating Procedures and IEEE 1058.1-1987 2. Including but not limited to: icons, approximate font size; locations of text and icons; user interaction 3. Defined by MTI Standard Operating Procedures and IEEE 1028-2008 4. Defined by MTI Standard Operating Procedures and IEEE 830-1984 5. Defined by MTI Standard Operating Procedures and IEEE 829-1983			

TABLE 1

Detailed here is fictional example of a Statement of Work (SOW). The example imagines creating a device that would use an existing automobile's OBD2 reader and provide a digital readout to display specific engine parameters on a car's display that are either missing or broken in a car. The SOW assumes that the scope of this project is to take the design up through production. You may choose to limit what the customer actually sees in the proposal. But delineating all of the tasks internally during the proposal and assigning an estimated cost to them are critical steps toward the success of your efforts.



ABOUT THE AUTHOR

Bob Japenga has been designing embedded systems since 1973. In 1988, along with his best friend, he started MicroTools, which specializes in creating a variety of real-time embedded systems. MicroTools has a combined embedded systems experience base of more than 200 years. They love to tackle impossible problems together. Bob has been awarded 11 patents in many areas of embedded systems and motion control. You can reach him at rjapenga@microtoolsinc.com.

- The PCB can be at least 3 cm x 2 cm.
- The device shall be designed and tested to work on the following OBD2 readers:
 - Yongtek ELM 327
 - Kufung OBD2 Reader
 - One other reader to be defined by the customer at PDR
- The device shall provide 4 digits of display with one decimal point.
- The display shall be visible in both daylight and nighttime operation in an automobile dashboard.
- The digits shall be at least 1 cm in height.
- The device shall display up to 10 specific engine parameters which shall be selected by scrolling with a single push button. The 10 parameters shall be defined by the customer at PDR.
- The specific engine parameter selected

Task (Red indicates a deliverable)	Responsibility	Schedule	Cost (man-hours)
Release HW for PCB layout	Us	1 day after CDR	2
Mechanical dimensions for the PCB defined.	Them	1 day after CDR	0
Layout the PCB	Third party	5 days after CDR	Pass-through charge estimated at \$2,000
Review the PCB layout	Us	6 days after CDR	2
Create a Software Design	Us	6 days after CDR	12
Perform a Software Design Review	Us and Them	6 days after CDR	2
Rapid prototype the interface to the three OBD2 readers	Us	10 days after CDR	10
Code the software	Us	10 days after CDR	12
Test the software modules	Us	15 days after CDR	12
Build the prototypes	Third party	20 days after CDR	Pass-through charge estimated at \$5,000 for 5 boards
Fit check with the plastic enclosure	Us and Them	20 days after CDR	2
Checkout the prototypes Us		2 days after receipt of prototype hardware (ARPTHW)	6
Integrate the software with the hardware	Us	5 days ARPTHW	10
Final plastic enclosure ordered	Them	5 days ARPTHW	0
Perform electromagnetic interference pre-scan	Us and third party	10 days ARPTHW	8 hours plus \$2,000/day for the lab
Make changes to the hardware design based on the checkout, the pre-scan and the integration.	Us	15 days ARPTHW	4
Re-layout the PCB	Third party	20 days ARPTHW	Pass-through charge estimated at \$1000.
Review the layout	Us	21 days ARPTHW	2
Build the preproduction boards	Third party	40 days ARPTHW	Pass-through charge estimated at \$5,000 for 5 boards
Checkout the preproduction boards	Us	41 days ARPTHW	4

shall be sticky across power downs.

- The device shall be powered by a battery which shall last for 1 year and be easily replaced by the user.
- The customer shall design the enclosure.


THE ASSUMPTIONS

Another critical piece of a successful proposal is the assumptions you are making. Some are very project specific. If you noticed in the example SOW in Table 1, there was only one run of plastic by the customer. That needs to be explicitly listed in assumptions. Also, you might have noticed that there were three turns of the board. If everything went perfect and the design was quite simple, one turn might suffice. How do you price something like that? You do it with an assumption that lists the cost per turn. You can provide the customer historical data that says "With a PCB this complex, we typically take three turns of the board." Another assumption might have to do with schedule. If the customer cannot deliver the plastic enclosure within the

proposed schedule, there would be a day-to-day slip of the schedule.

Finally, you should provide general assumptions that are part of your boiler plate proposal. For example, we provide a statement about who owns the intellectual property. We also provide a statement about how we will bill outside services. These should be clearly written and honed over the years as we did. Build up a template for your proposal and keep improving it.

CONCLUSION

My old boss told us that we cannot make money on development. He was right. But you can make a living at it. And I don't regret the exciting challenges and fun we have had for almost 30 years designing a wide range of embedded systems. One of the reasons for our success was well written proposals with SOWs, Specifications, and stated Assumptions. We covered a lot in this series. But even with all that, we only did it in thin slices. 

Integrate the software with the hardware	Us	2 days after receipt of preproduction hardware (ARPPHW)	4
Perform preliminary EMI testing	Us and third party	5 days after receipt of ARPPHW	8 hours plus \$2,000/day for the lab
Perform Systems Test	Us	5 days after receipt of ARPPHW	8
Iterate the hardware design and software design as a result of the System's test and EMI testing	Us	10 days after receipt of ARPPHW	8
Re-layout the PCB	Third party	20 days ARPPHW	Pass-through charge estimated at \$1,000.
Review the layout	Us	21 days ARPPHW	2
Final plastic delivered	Them	40 days ARPPTHW	0
Build the production boards and mount in the plastic enclosure	Third party	40 days ARPPTHW	Pass-through charge estimated at \$4/board for quantity 1,000
Checkout the production boards	Us	41 days ARPPHW	2
Integrate the software with the hardware	Us	2 days after receipt of production hardware (ARPHW)	2
Perform final EMI testing	Us and third party	5 days ARPHW	8 hours plus \$2,000/day for the lab
Perform Systems Test	Us	10 days ARPHW	8
Create Release Notes and a Version Description Document	Us	12 days ARPHW	4
Release to the customer⁶	Us	13 days ARPHW	2
6. The following documents will be delivered at final release: <ul style="list-style-type: none"> • Software Project Plan SRS • Hardware Schematics • Gerber Files for PCB • Software Test Plan and Results • Release Notes and Version Description Document 			
Customer acceptance	Us (handling questions) and Them	30 days after release to customer	12