

Software Development for Consumer Electronic Devices

By: Gopal Miglani
President
SoftProse, Inc.
1776, 22nd Street, Suite 100
West Des Moines, IA 50266
Phone: +1-515-221-2220
Fax: +1-619-839-3859
Email: gopal@SoftProseInc.com
URL: www.SoftProseInc.com

To state the obvious and cliched, Consumer Electronic devices are becoming more intelligent and complex. The Convergence phenomenon is only accelerating this pace. Almost all the intelligence and a good part of the complexity of these devices is attributable to software. The good news is that companies with existing CE know-how are best positioned to develop, produce, market and support these next generation CE devices. The bad news is that they are frequently not equipped to design, manage and support software with the level of complexity that is inherent to these devices. This article focuses on the issues raised by this transition.

The first and most important step to successful software development for CE devices is a **realization** of its importance. Walk around your department and if you see any of these telltale signs, read on. a) Drawings detailing PCB layout and chip design, but no software design and documentation specifications. b) Version control on hardware drawings and specs but none on software. c) Not selecting hardware and software vendors in parallel. d) Only EE's developing code. e) Lack of any effort in the company or department to move to a single OS and tool set.

Develop and nurture a **software team**. Today's job market is on Internet steroids. If you can only hire after product budgets are approved, you have a problem. Most software engineers who know what they are doing when it comes to CE software development can write their own ticket.

Standardize on an operating system and tool set. It is more important to pick one of the major operating systems for all projects, department or company wide, than to jump from OS to OS based on price, tools, availability of existing components. I would attribute a minimum 100% productivity improvement to experience with the OS and tools.

Build a **software library**. Recent graduates, professors and researchers may refer to this as object oriented development. The truth is you ought to have a large manual, preferably on-line, detailing functions, classes or objects with well defined inputs, outputs,

globals, functionality and persons responsible. Object oriented does not have to mean C++ or Java. C or assembler is just fine. In fact, look before you leap into C++ or Java.

Using a **virtual machine** or **scripting engine** is the next logical step after having a reusable software library. Java is not the only game in town, there is also Inferno from Lucent and others. For scripting engines, consider Lingo from Macromedia and MHEG. And, it is always possible to develop a proprietary virtual machine or scripting engine of your own. While virtual machines and scripting engines free you from CPU architectures, be careful not to get locked into a virtual machine or scripting engine. While considering the cost of the VM or SE be sure to add the cost of the incremental computing resources required and subtract the cost of not having to port code to new architectures. In the long run, the latter will tend to out weigh the former.

Do not underestimate the **complexity** of software in CE devices. I like to say that this is rocket science and you need rocket scientists to pull it off. The devil is in the detail. Remember the browser, the email protocol, the ATM driver, the HTTP server, but do not forget the NVRAM driver, the fail safe flash upgrade algorithm, support for MPEG stills, the list goes on.

Document your software and follow a software life-cycle development methodology. At a former job, this methodology was put into place when we started out to obtain our ISO-9001 certification. We found to our surprise that engineers liked following the methodology and software developed using it was less buggy and delivered closer to projections.

When I think about the attributes of a CE device, I think of words like image, functionality, price, reliability and personality. Hardware is a major part of the price equation and software is a major part of the device's personality. Hardware and software rate about equal when it comes to the other attributes. Overall, hardware and software have to be given equal importance to produce a successful CE design.