Chapter 17
Mission Critical Embedded System Development Process:
An Industry Perspective

Stefano Genolini
TXT e-solutions, Italy

Matteo Crippa
TXT e-solutions, Italy

ABSTRACT

While analyzing currently available international research about embedded system development, it seems that as the complexity of embedded systems is continuously increasing, the major problems regarding their development remain always the same: vague requirements, insufficient time to develop, lack of resources, and complexity management. With the focus on the development process, it is shown, with examples coming from 20 years of experience, the industry perspective of a company managing such problems by adopting a consolidated set of good practices.

INTRODUCTION

Whilst a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs, an embedded system is a computer system designed for specific control functions within a larger system, often with real-time computing constraints. Development of embedded software applications is becoming more and more complicated due to the increasing richness of features that are required to be managed.

An international on-line research[1] about embedded systems development conducted a periodic survey in 2012 by contacting industry developers; and presented a periodic report summarizing the major topics in embedded development industry. According to this research the most critical issues perceived by embedded software developers are:

- Incomplete or vague requirements (63%).
- Insufficient time for development (45%).
- Insufficient resources (41%).
- Design complexity (41%).

DOI: 10.4018/978-1-4666-6194-3.ch017

What is interesting to note is that, during the last several years, the major problems always remain the same: vague requirements, insufficient time to develop, lack of resources, management of system complexity, among others.

It is thus possible to argue that:

- Vague and incomplete requirements and design complexity are the consequence of increasing complexity of targeted applications;
- Insufficient time is the consequence of reduced time to market;
- Insufficient resources are the consequence of reduced budgets.

Complexity, time to market and costs issues together with development process are analyzed in the sections below:

COMPLEXITY

In recent years, complexity of embedded systems has continuously increased and is foreseen to augment more exponentially, as our lives are becoming integrated to a vast and complex ecosystem of embedded devices and part of an interconnected world. For example in the automotive domain, in a car, the number of communicating on-board computers or Engine Control Units (ECU), according to the car model, may be more than 100 in number. Maintaining connectivity between present ECUs is thus becoming an essential aspect in automotive domain, and this has led to more complex applications being developed, new user experiences for UI being created and new protocols being managed.

Of course, the overall design strategy has to also take into account issues such as related to quality, security, safety and, overall design costs.

TIME TO MARKET

Time to market for car model manufacturers (but this applies also to majority of application domains) forces them to stress reusability across cars variant. Reusability equally implies modularity in the solutions (both hardware and software), software portability on different platforms, adoption of Customized of the Shelf Components (COTS), customization of features, configurability, etc.

COSTS

Budget constraints also force to implement the same features with less effort. This can be managed by adopting a more structured way to operate, such as: enforcing reusability, adopting all-inclusive process development tools, adopting process development models in order to obtain better processes and better product quality and training personnel to better achieve their objectives.

All of these critical factors need to be managed together. Solving only one does not bring sensible advantages, but surely a structured (formalized) way to behave during the development can enforce a proper way to proceed. This is the reason that we introduce, subsequently in this chapter, the development process as the key factor to support a cost effective and quality based approach to development of embedded systems.

DEVELOPMENT PROCESS

In software production the process has been analyzed many times leading to different solutions and specific best practices. This is part of the history and the intent of this chapter is not to choose which is the best or create new process ideas.

In embedded systems the development process has always had a special role: what are the reasons for this? It is certain that certification constraints