DEVELOPING NEW PRODUCTS IN HALF THE TIME

Rok Ursic, Instrumentation Technologies, Srebrnicev trg 4a, SI-5250 Solkan, Slovenia

Abstract

Accelerator devices are in general systems consisting of hardware and software components. From the customer (operator, user, machine physicist, ...) point of view they are products (systems) that should offer excellent performance and reliability. In this article we present reasons and advantages of developing products faster and present the opportunities for shortening the development cycle.

1 INTRODUCTION

1.1 Why Developing Products Faster

The motivation for rapid product development is different for each organization. On the one side we have business organizations. Their motivation can be to increase sales, beat the competition to market, be responsive to changing markets, styles, technologies or maintain the market leadership. Industry offers some outstanding examples. Hewlett-Packard, which used to require 54 months to complete a major new computer printer project, reduced the interval for its first DeskJet to 22 months and then to 10 months for the first color one, the DeskJet 500C. Intel has reduced its development cycle for motherboards from 12 months to 6 months, then to 3 months [1].

Motivation for shortening development cycle is certainly different in the field of particle accelerators. In general, budget cuts force new project proposes to shrink time frame for building a new machine (accelerator) in order to make the project (politically) appealing and get the approval. As a consequence all the subsystems have to be operational in a shorter time frame. An other reason for shortening development cycle may be a sudden and urgent need for performance improvement an accelerator (sub)system. Sometimes we want to respond to changing technologies. There may be other reasons. Even though are the reasons for shortening the development cycle different for industry and particle accelerator, one issues stands out in both cases. How to maintain or even improve product's quality given a shorter and shorter development cycles.

1.2 The Project Management Boundaries

Successful project management is in its strict sense defined as having achieved the project objectives:

- within time
- within cost
- at the desired performance/technology level
- while utilizing assigned resources effectively and efficiently

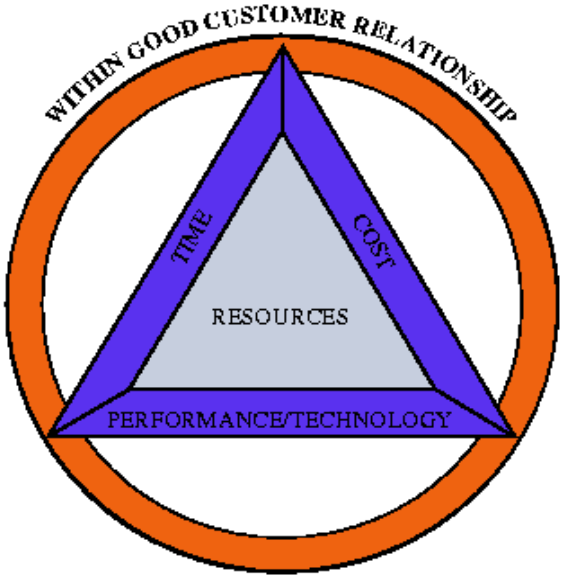

Figure 1: A pictorial overview of project management.

Too many times people connect accelerated product development with project management. This perspective is too narrow, because it ignores the product-specific elements and the characteristics unique to product development projects. Also, all too often "project management" fails because its key element is ignored: management does not provide its project manager with the authority or support necessary to reach the objective.

1.3 No Recipes

In our experience, the universal solutions are not effective, because each firm has different requirements and constraints. Furthermore, in order to create the appropriate environment, each solution must draw upon that organization's strengths - there are cultural, geographical, skill, knowledge and moral, to mentioned just few of them, differences between them.

In this article we present some opportunities for shortening development cycle. The idea that we advocate in is to probe the pros and cons of various approaches, respecting the complexity of the problem.

2 THE OPPORTUNITIES

2.1 The Fuzzy Front End

The opportunities for shortening the development cycle, large and small, appear throughout the development process. There is, however, one place where we found the least expensive opportunities to achieve large improvements in development cycle. Let us call it the fuzzy front end. It is the fuzzy zone between when the need for a new product is known and when we mount a serious effort on the development project. An interesting fact is that our memories of the fuzzy front end tend to be especially fuzzy. We incorrectly remember both the time when the opportunity was available and the time when a full development team was started.

The fuzzy front end is by definition on the critical path of the project. Every delay in this phase goes at the expense of other development activities or, even worse, delays the product delivery.

If we analyze the front-end processes, we will discover that they are composed of bursts of activity followed by long stretches of inactivity. One way to avoid this is to establish a spirit of urgency already at this stage. It can be done by explicitly assigning somebody responsible for this portion of the development cycle, by providing adequate resources and by setting clear deadlines.

An other tool we suggest is to subdivide the initial planning activities and overlap them with design. Do we really need to complete all our planning activities before we begin any of our design activities? By overlapping these two processes, planning remains on the critical path of the project for the shortest amount of time possible.

2.2 The System Architecture

The right product architecture can be a secret weapon in creating shorter development cycles. In general, system engineers design products to meet customer requirements. But doing so doesn't tap the full potential of product architecture, which can be powerful weapon to achieve many other objectives, especially shorter product development cycles. A poor architecture can easily double the length of a development cycle by adding scope, forcing sequential design and increasing rework.

Product architecture can act in three ways to shorten development cycles. First, it's a key control point for project scope - an excessive scope is one of the greatest causes of long development cycles. Second, good product architectures dramatically increase design concurrency by allowing many pieces of the system to be designed simultaneously. Third, good architectures reduce both the frequency and the magnitude of rework in the design process.

Controlling the scope of the project is one of the key factors in reducing development cycle. In our practice we found that designers can use architecture to reduce system scope in three ways. First, by carefully choosing the system boundaries. Next, by using modular product structures. Finally, by making careful make/buy decisions for subsystems.

2.3 The Technology

We normally do not have the luxury of developing technology as a part of our development effort. This would add to much uncertainty and delay to the program. In other words, we should never put a new technology on a critical path of a development project. Instead of including it as a part of the product development, the technology must be developed outside of the program. This means that we have to anticipate technology needs before they are required and start developing this technology early. These needs should be carefully planned at the group or even higher organization levels.

Development team should use incremental innovation when implementing new technologies. Think of incremental innovation as a way of doing cheap experiments in product development. A final and more general advantage of incremental innovation is the way it accelerates the learning process in organization. We learn a bit more about the behavior of the product, about the technology, and about our process of developing products with each additional development program. The more times we repeat this cycle in a given time period, the more chances we have to learn and to reinforce this learning.

2.4 The Team

A group that develops new products is often called the development team. This term probably grew out of a recognition that a more cohesive group than usual is needed to get a new product. As the product development quickens even more is demanded of the development group. The word team is in our opinion heavily overused. It is highly ranked on almost every manager list or vocabulary.

A lot has been written about team staffing and teams in general. In this paragraph we discuss three issues we consider important.

First, fast product development is a lot of work and requires a high level of commitment. Being on such a team is neither easy, safe, nor predictable. The reason people would want to be involved in such an undertaking is that it offers some things not otherwise available in structured environments: excitement, a chance to learn new skills, or an opportunity to put one's name on a specific product. The only way to obtain a required level of commitment to an accelerated development project is for participants to make a conscious decision to be fully involved in it. The recruiter's job is to identify each prospect's motivations and present the virtues of the team membership accordingly but honestly.

Second, anyone developing a highly technical product will need technical specialist on the team. This is especially true when introducing new technology or pushing on the state of the art in it. Having a technical specialist on a team should be thought of as the exception, however. Specialists tend to fragment the project focusing only on their narrow area and not paying much attention to the project as a whole. For this reason, generalists should be assigned to the team whenever possible. Doing so helps the team work more effectively in two ways. Most important, it keeps the work within control of the team members responsible for maintaining the schedule. The second advantage is that generalists give the team much more flexibility in keeping its members productively occupied.

Third, development teams by their nature tend to be heavily staffed with engineers, who will have studied lots of math but probably not much psychology. Engineers are trained to construct logically tight arguments and filter out apparently extraneous emotional factors. The problem with this approach is that success in product development depends less on scientific purity than on such human emotions as enthusiasm, satisfaction, frustration, boredom, and depression. The astute team leader thus should pay attention to these factors rather than filtering them out.

2.5 The Process

Process has become more important topic in recent years, as companies have delayered their organizations, thus de-emphasizing the role of organizational form and emphasizing process in its place. ISO9000 has encouraged the shift to an established development process.

In our opinion there is no perfect recipe for a rapid development. Process is certainly an important element, but not the only one. The problem with the process is that it is effective in a particular organization to the extent that it has been adapted to the specific needs and strengths of that organization. Borrowing someone else's process is not the way to fast development. The good news here is that once we have an effective process, it is a valuable asset of our organization. The bad news is that we have to put effort into building our own process. Another aspect of the process is that organizations that are good at it continually work at it, modifying and honoring their process to suit ever-changing conditions. Due to limited scope of this article we will cover only two areas we found particularly fruitful.

The product development process used in many companies and some laboratories was originally patterned after the phased project planning (PPP). In this process the project passes through checkpoints sequentially to ensure that all the items required by that checkpoint are in good order. Any problems are corrected and the project proceeds sequentially to the next checkpoint. The PPP project is wonderful for catching items that have been forgotten and for assuring that organization's fund are not spent without justification. We suggest, if appropriate, to avoid the PPP process or at least minimize the number of checkpoints and use other means, some of them suggested in this article, to accelerate the development process.

Overlapping - working on multiple activities simultaneously - is an other very important and effective tool of rapid product development. To gain a perspective on this issue we should think of product development as a process of gradually building up a body of information until it eventually provides a complete formula for building and integrating a product. Overlapping influences the pattern of accumulating this knowledge. In the traditional, more comfortable approach, information about a topic builds up until it is virtually complete, then is transferred to next activity, where it is used to build the body of information needed for that task. One major opportunity and challenge is to overlap hardware and software development and testing. Such overlapping is not natural and will not occur without some specific attention to it. The opportunities to overlap hardware and software activities often relate to testing, where complex scheduling linkages often exist. The software people need hardware to test their code and the hardware people need some software to even make their hardware work. Potential solutions include:

- Using modern requirements-driven software development methods, which defer the need for a testbed for he software and reduce tail-end-debugging effort when the testbed is available.
- Planning the hardware and software testing early, so that scheduling challenges can be identified while there is flexibility left to resolve them.
- Employing a simulator as a software testbed; this allows some early software testing and later provides a tool to isolate hardware-software interaction problems in the real hardware.

3 CONCLUSION

Most of us think of a development project as delivering only one item: the product itself. We are pleased if we get that as planned. But there is a second, valuable deliverable available: learning about the process that was used. Organizations at the higher levels of development-process maturity expect both deliverables. They do not consider a project complete until both are delivered. Expecting both is a learned style, a new habit. Just like other habits, you acquire it by practicing it.

REFERENCES

[1] P.G. Smith, D.G Reinertsen, "Developing Products in Half the Time", Van Nostrand Reinhold