

Firefighting, or We'll Figure It Out Later

By Steve Mowry

I first heard the term “firefighting” used to describe chaotic and/or panic type product development activity from my supervisor within the transducer R&D group at BOSE Corp. in 1995. As a “freshy,” I did not understand and therefore was not concerned about this phenomenon at that time; however, in retrospect, I cannot recall any serious attempts to control and/or eliminate firefighting. It seemed as though the bigger the unforeseen problem, the more people along with their respective corporate stature (title/rank) would attend the “meeting.”

I can even recall a colleague attending such a meeting in 1997 wearing a classic red fireman's helmet! He left the company shortly after that. Now a decade later, I will discuss firefighting within the context of my New Product Development exploits and recommend ways to avoid or minimize firefighting.

Firefighting is defined in the Product Development Management Association's (PDMA) glossary of terms (www.pdma.org/library/glossary.html) as an unplanned diversion of scarce resources and the reassignment of some of them to fix problems discovered late in a product's development cycle. This metaphoric terminology also implies an element of unexpectedness or surprise unplanned tasks.

CONSEQUENCES

Unfortunately, firefighting imposes numerous costs on the New Product Development project(s). Introduction dates are often pushed back; engineers and managers sometimes work extraordinarily long hours leading to fatigue, burnout, turnover, and increasing the chance of further errors; and additional people are often added to the project, thus requiring additional development expense. Despite these costs, firefighting regularly occurs.

Developing new products is a funda-

mentally uncertain task, often involving numerous unproven technologies and processes. My experiences in Asia indicate that firefighting, rather than being isolated to specific projects, is the accepted mode of practice for developing new products in most companies. The existence of firefighting as a steady state rather than a temporary phenomenon limits the ability of these organizations to properly execute NPD processes in a number of ways.

Perhaps most important, successful projects almost universally point to the added value of careful planning and investing resources early in the product development process. Organizations that engage in firefighting find it difficult to make such commitments, thus making it impossible to follow the development processes described in the literature. Similarly, organizations that dedicate large portions of their available development resources to fixing unanticipated problems also find it difficult to successfully implement NPD tools, methodology, and processes.

Almost everybody agrees that firefighting is detrimental to performance, yet paradoxically it persists. Firefighting can be a self-reinforcing phenomenon. Once it starts in one project, it is likely to spread to others, permanently degrading the core capability of the development system.

Invariably in technical companies there are more problems and opportunities than time or people to deal with them. There are more problems than the organization should work on than it can realistically do so. This ultimately leads to situations in which some problems are ignored or patched. Too often, it leads to firefighting “syndrome” or what I will take the liberty to refer to as the “Asian Way.”

In firefighting syndrome, engineers, managers, and other knowledge workers rush from task to task, not completing one before another interrupts



them. The latest fire continually interrupts things that are merely “important” but not “urgent,” such as long-range development. Some managers evaluate employees simply by a review of time records; staff is expected to work until at least 8:00PM in many companies.

Although the most urgent tasks receive attention, productivity suffers. Some jobs are never completed. Although a task may appear to be complete, it may need to be redone or revisited later, perhaps even in another form, because it was not completely solved the first time around. Management is in a constant juggling act of deciding where to allocate overworked people and which problem to ignore for now.

In the sometimes confusing and multidimensional world of modern multimedia product development companies within Asia, the chaotic behavior referred to as firefighting persists with limited efforts to avoid it. Firefighting consumes resources and saps morale. In some companies, it seems pervasive and culturally embedded, yet it has been seldom studied and discussed. Remember that firefighting is a drag on performance even when the organization can keep it under temporary control. Both the efficiency and effectiveness of problem solving are reduced as more resources go into various forms of firefighting.

Once firefighting establishes itself, it is likely to worsen. At some point senior management will face unpleasant but impending choices. Only drastic action may suffice, such as outsourcing a large fraction of the work, shutting down and “starting over” in some fashion, or bringing in a massive infusion of outside help, all while considering whether to fire the engineer(s) “responsible” for the problems. Such drastic gyrations are a major drain

on money and management's time. In essence these are problems (fires) for senior and middle managers.

PUT OUT THE FIRE

Given the self-perpetuating and vicious circle nature of firefighting, getting out of it once you are in it requires determination. An incremental approach may not work, because a large adjustment in resource allocation is required to get the backlog of work down to where the organization no longer feels pressured to firefight. Any improvement smaller than this can make things better, but probably only temporarily.

The core competency of a company's development team(s) influences the likelihood of firefighting. The lower the standards, the more likely firefighting related to capability to effectively solve problems will be practiced. Recruit and keep good people. I cannot see how an engineer who is asked to be innovative and diligent would not be rewarded with some kind of wage increase or financial incentive for a three-year period or more. I have encountered this complaint from junior engineers on more than one occasion and it raises the question, "Do engineers in Asia under-perform because they are under-compensated, or are they under-compensated because they under-perform?"

Consultants can be of help with not only problem solving and technology transfer, but also training of staff members. Turnover and the abrupt loss of critical human resources can send a NPD project into turmoil and thus initiate firefighting activity. Freshies are less adept at problems and may not be as low a cost skilled labor as they seem.

There are companies in the US, Canada, Europe, Australia, along with Japan and Singapore (the Asian exceptions) which have found many ways to avoid or break out of firefighting. I will break them into three categories: tactical, strategic, and cultural.

Tactical methods can be put into ef-

fect quickly, during current projects, without high-level policy changes. Strategic methods take a deeper commitment and more time to implement. Cultural changes require shifts in the culture of the whole company, including senior management. In fact, cultural changes are an essential step in permanently getting rid of firefighting. Otherwise, no matter how much improvement is achieved by tactical and strategic methods, a few bad financial quarters will likely lead to pressure to improve short run performance at the expense of the long run, setting off a new round of firefighting. Management must commit to avoiding firefighting.

TACTICAL SOLUTIONS

There are many actions a company can take immediately that will ameliorate or in some cases solve the problem of firefighting in ongoing projects. Some of them are culturally difficult, while others are just a matter of recognizing the problem and applying resources to it.

A good short-run solution in situations where the rate of problem arrivals

has suddenly increased is to bring in temporary assistance or an independent consultant. In many high-tech projects, companies have learned

to send development engineers from the US or Europe directly to Asia whenever a new product starts in the factory. Not only do these visitors provide more troubleshooters, they also have special expertise because they may have seen related problems during prototyping. Furthermore, this has good long-term incentive effects because the Western engineers know that patching problems in development will lead to more problems during ramp-up, leaving them stuck in China for a longer time when ramp-up comes. Even individual engineers know that if their part of the design is still iffy, they must stay longer than their colleagues.

There are potential drawbacks to this

solution, of course. First, it only works when the excess workload is cyclical rather than chronic. Second, if you pull knowledge workers from other parts of the organization, you risk setting off firefighting in those areas. Third, if you use knowledge workers who come from outside, they will be unfamiliar with the situation, and you must deploy them in a way that they can take over more routine or independent tasks that require less familiarity.

Development companies sometimes decide to push out a release date to give more time for problem solving and avoid firefighting. This gives more time for the available resources to solve the same number of problems. A variant of this is to chop off a number of features from the planned release, and release them in subsequent models.

A more proactive way to reduce a backlog of work is to prevent it from growing in the first place. When the number of open problems becomes too large, shut down until they are solved. Only allow a new problem to go into the queue when an old one is removed. Most non-firefighting companies do this instinctively.

A more controlled way to limit the problem queue size is to deliberately do what will happen anyway, admitting that some problems will not get solved. Rather than let problems queue up indefinitely or work their way through the queue only to be worked on sporadically and then either patched or dropped, determine to commit resources to a problem when it first comes up. This technique is organizationally difficult, because it is much easier to tell someone, "We'll get to your problem as soon as we can" and then assign it to an overworked employee than to tell them, "We have looked into the problem and decided the situation is not critical enough. If you want that problem solved you will have to deal with it yourself."

This is a tough stand; however, if it is most likely that resources are not going to get to that problem, at least that engineer is free to think of other ways to work around the difficulty. Too often in firefighting situations, the rule "the urgent drives out the important" may be applied. Another

"Cultural changes are essential in getting rid of firefighting."

popular rule is “the urgency of the problem is driven by the rank or stature of the requestor.” These forms of prioritizing resources are both symptoms and causes of firefighting.

Avoid patching problems. In the short run, patching almost always appears to be faster than solving problems systematically. Therefore, it is typically one of the first responses by engineers who are being pressured to work faster. Whatever they were taught about systematic problem solving goes out the window and in the long run, patching fails to actually solve the problem and may produce new ones. In most settings, it is better to solve a few important problems well than to patch a number of problems hastily. The net losses due to bad patching will outweigh any gains from lucky guesses. This is a controversial principle.

One counterargument from some managers is that engineers work harder under pressure. The implication is that harder is equivalent to better, so managers should demand ambitious results, then get out of the way and let the engineers choose their own way of working. This is true but only up to a point. Once the pressure becomes too great, engineers start taking shortcuts, not just working more hours. Junior engineers have told me, “I just don’t have time to do simulation.”

There are a few situations in which modest patching is justified, although this approach is not recommended. Most problems can be solved at different levels of depth. A hasty or superficial solution is both faster and less useful than a careful or deep one.

The key determining factor is the benefit/cost ratio, where cost is measured in hours of problem solving time. If a problem is of very low importance and/or it affects only a few end users, then it could be best to invest zero time on the problem. If the problem is a serious one, or a hasty solution could have very adverse side effects, it is best to find the root cause and solve the problem thoroughly.

STRATEGIC SOLUTIONS

Strategic approaches to firefighting take longer to implement, but pay off

across a broader range of projects and over longer time periods. Even if they are not sufficient to avoid firefighting, they will increase the number of problems that are solved.

The last decade has provided new insights into how to design and develop products. For example, companies have had great success in increasing the commonality of designs from one generation to the next, and across products. Not only does this reduce the number of design problems that must be solved within and across product generations, it also reduces the changes and therefore the problems in manufacturing. Commonality can be further enhanced by modular designs, which allow improvement of one section of the product without much change to the others.

One of the reported successes of the Total Quality Management (TQM) movement was that in some companies, non-engineers were trained to solve problems. Although they are not as fast or as broad problem solvers as engineers, by handling many of the more mundane problems, technicians and workers can free up resources for the more difficult problems. However, this advantage can be eaten up if problem solvers are led or encouraged by management to merely patch problems.

Better simulation tools and an increased understanding of problem solving as a task make it possible to improve the problem solving speed and accuracy of individual knowledge workers. Higher-level methods, such as structured design methodologies, can also be quite effective. However, there is a potential trap here: the introduction of improvements can permanently worsen performance. While new methods are being introduced, they inevitably reduce engineering productivity at first, because it takes time to learn to effectively use the new tools. If the introduction is too rapid, the effect can be enough to tip the company into more firefighting.

CULTURAL SOLUTIONS

Even in a company that has implemented tactical and/or strategic methods, there will still be occasions

when extra work arrives and produces an unexpected backlog. Pressure to resume firefighting then occurs. It is almost always possible to make some temporary and/or apparent gain in development performance by firefighting methods, such as concentrating resources on short run payback at the expense of important problems, patching in the hope that nothing will happen right away, and so on. This is where much of the confusion is manifested in a company culture that is conducive to firefighting.


If managers are too far removed from the problems to see the consequences, and if the reward system favors firefighters, then the vicious circle will begin again and the system will again go into firefighting mode. Avoiding the firefighting syndrome depends on the culture and mindset of middle and senior managers.

Don’t reward firefighting. In many companies, the hero is the one who puts out the most visible fires. But this leaves the questions: “Where were they when the problem started? Why didn’t they intervene sooner, before it became this large?” Instead, reward managers and engineers who don’t have visible fires, who plan and allocate reasonably, and who practice long-term problem solving.

Don’t push to meet deadlines at all costs. Such incentives favor firefighting. Evaluate development projects by how few problems are encountered and how many problems remain or reappear afterwards.

Under-funded and/or understaffed projects inherently encourage firefighting. Be reasonable. I was asked by a client in Malaysia in 2005, “How long will it take to develop a “new” transducer?” My reply was, in this case and based on the support capability and available funding, “no less than one year.” The general manager then replied, “We can do it in two weeks.”

Be realistic when planning and budgeting resources for NPD projects. Build a problem solving mentality within the company. In today’s highly dynamic business environments, innovation, improvement, and dealing with the unexpected are key tasks. The unexpected can take the form of

problems, which, when solved, open the door to innovation and improvement. Does this then imply that firefighting limits or constricts innovation and improvement? I sincerely believe this to be the case. 

Steve Mowry is president of S. M. Audio Engineering. He has a BS degree in Business Administration from Bryant College and BS and MS Degrees in Electrical Engineering with highest distinction from the University of Rhode Island. He has worked in R&D at BOSE, TC Sounds, EASTECH, and P.Audio. Steve is currently an independent researcher, lecturer, and consultant in transducer/loudspeaker system design and new product development. His website is www.s-m-audio.com.

REFERENCES

1. R. E. Bohn, *Firefighting by Knowledge Workers*, March 2000, <http://isic.ucsd.edu/pdf/Firefighting.pdf>.
2. N. P. Repenning, *Understanding Fire Fighting in New Product Development*, March 2001, http://web.mit.edu/nelsonr/www/Tilting_v40-web.pdf.