

## When Politics Overshadow Software Quality

**Scott Stribrny and Fran Boehme Mackin**

**O**n the road to software quality, life happens. Even for someone who's really passionate about software quality, it's hard to focus on quality for long because other things intervene. As software industry expert Fred Brooks said about software becoming late one day at a time, our focus on software quality also dissipates one day at a time.



The following scenario is familiar to most of you: management, feeling pressure from corporate headquarters and, more important, from the marketplace, dictates that a product will be released ahead of the agreed-upon schedule. This is an example of people not doing what they're supposed to do—keep to the schedule. It's also an all-too-common example of a company's political climate in which influential people base their desire for success more on personal agendas than on quality.

This scenario illustrates two key issues: the frustrations involved in producing quality products and the conflicts between self and team. How would you balance the quality and

time-to-market equation? Consider that software quality is subject to the politics of

- *control*, when one person can tell others how things will be done;
- *position*, when an individual's rank in the corporate structure influences outcomes; and
- *power*, when one individual has power over another, as with setting salaries.

In turn, the politics of control, position, and power affect what questions are asked, who may ask them, and what the “right” questions are. And the question asked can control outcomes. When all these political factors—control, position, power, and questions—converge, the politics of communications and marketing come into play. These determine what people hear and how interaction styles affect outcomes.

Everyone in the corporate hierarchy must consider these political realities. They aren't just for management.

### Scenario

As each department's management team contemplated the schedule compression's impact, self-interest was often the primary consideration. When telling the team about this new mandate, for example, the vice president of software engineering indicated how the organization would accomplish the task and that software quality would not be at the top of their list of priorities. Everyone processed this new directive and thought about the impact it would have on them and their teams. No one wanted to be the

first to say they couldn't complete this task. No one questioned why the president was making such a demand, nor asked questions such as what level of quality this first shipment would require.

The quality assurance manager, in a defensive position, began quoting a stream of "rational" data from the weekly reports about inadequate equipment and staff availability. Surely these facts would persuade the other managers, who seemed focused on merely getting the product out rather than getting a quality product to market. In the past, the QA manager had a patron to report to, an executive who understood the customer impact of not completing the testing cycle. When the QA manager had reservations about software quality, this executive listened. But ever since this executive changed jobs and the QA manager reported to someone else, much had changed.

If only the QA manager could logically demonstrate that the company could improve the bottom line by improving product quality, first by documenting system features so that QA could write test plans prior to receiving code, then by taking time to test the code. Why didn't executive management, or for that matter all levels of management, understand poor software quality's effect on the staff, before the final product release? Not wanting to take an adversarial position with the other departments, especially engineering, the QA manager outlined how this new initiative would impact the QA staff's ability to support other initiatives.

In this scenario, external forces and the lure of a quick climb up the corporate ladder seemed to influence most of the staff. But what of the individuals who saw the road to customer satisfaction as a quality process pathway? Why did it seem that those who focused on recognition for themselves rather than the production of quality software were the ones getting ahead?

**Principles**

Principles are general truths founded on what others have proven or learned via experience; you can use them as guides to inform your actions. Looking

beyond the politics, what principles could the QA manager have employed to ensure that quality remained an important objective?

The first principle is to accept that all organizational life is political. This political game, however, isn't like other games with rules and one winner and one loser. This game has many "truths," and not all are based on what QA managers value: rational thinking, logic, and facts. Rather, there are many vantage points to an issue.

Another principle is to build relationships and partnerships. You're part of an ever-changing organization, and you work with others to promote the company's and your agenda. Look around at successful coworkers. What are their relationship patterns? Whom do they interact with on a daily and weekly basis? Most likely, they have a patron who offers them advice, insider information, and a type of beneficial partnership that has developed over time. In our scenario, the QA manager neglected to replace the patron. Not only did frustration set in, but also the QA manager lost the ability to influence others through the patron.

Yet another principle is to not try solving problems immediately, but to try looking at situations first through others' perspectives as a way of overcoming our own bias. This is where the QA manager's ability to frame the issues can pay off. For example, did the manager consider why time-to-market was driving a reduction in testing time? Did the manager ask to be involved in the requirements phase so that his team

could begin developing testing scripts then and complete testing in parallel with development?

**Surviving the journey when life happens**

Are you willing to take responsibility for defining your organization's software quality agenda? It's not a one-time event, nor an easy task amidst all the pressures that can slowly erode the organization's ability to create quality software. To do this well, you must be clear about what you want, where you want to go, and how far you want to go.

How can you take responsibility for defining your organization's software quality agenda? In *The Empowered Manager* (Jossey-Bass, 1991), author Peter Block argues, "The process of organizational politics as we know it works against people's taking responsibility. We empower ourselves by discovering a positive way of being political. The line between positive and negative politics is a tightrope we have to walk." Amid all the pressures that slowly erode the goal of creating quality software, choose a positive, principled way of being political.

As we've said elsewhere ("IT Politics: Strategies for Personal Survival," *Cutter IT Journal*, vol. 8, no. 4, 2005), you must also be clear about the extent to which your current organization is at the right stage for you to act out this practice. It's also about having the courage to follow your convictions, even when confronted by opposition. Politics is all about pitting one set of values against another. But unlike pitting technologies against each other, unambiguous benchmarks and objective tests don't exist. There's only your ability to persuade and influence others' actions.

As with any other approach to getting things done by influencing people, you can't prepare for all the challenges you might encounter by reading articles, papers, and books. Try putting these principles into practice on your own road to software quality. ☺

**Scott Stribny** is president and managing director of Group Atlantic. Contact him at [scott.stribny@groupatlantic.com](mailto:scott.stribny@groupatlantic.com).

**Fran Boehme Mackin** is the founder and president of Goals Etc. Contact her at [goalsetc@comcast.net](mailto:goalsetc@comcast.net).

**Not all "truths" are based on what QA managers value: rational thinking, logic, and facts.**