

Is “Process” getting in the way of process?

Part 2 of 2 -- Survey Results

How does your organization handle process?

By Harry Kitchen, CSQE
hkitchen@columbus.rr.com

At the April 9, 2002 meeting of ASQ Section 801, the author gave a presentation on the effect of compromise in retarding continuous process improvement for software development and business in general.

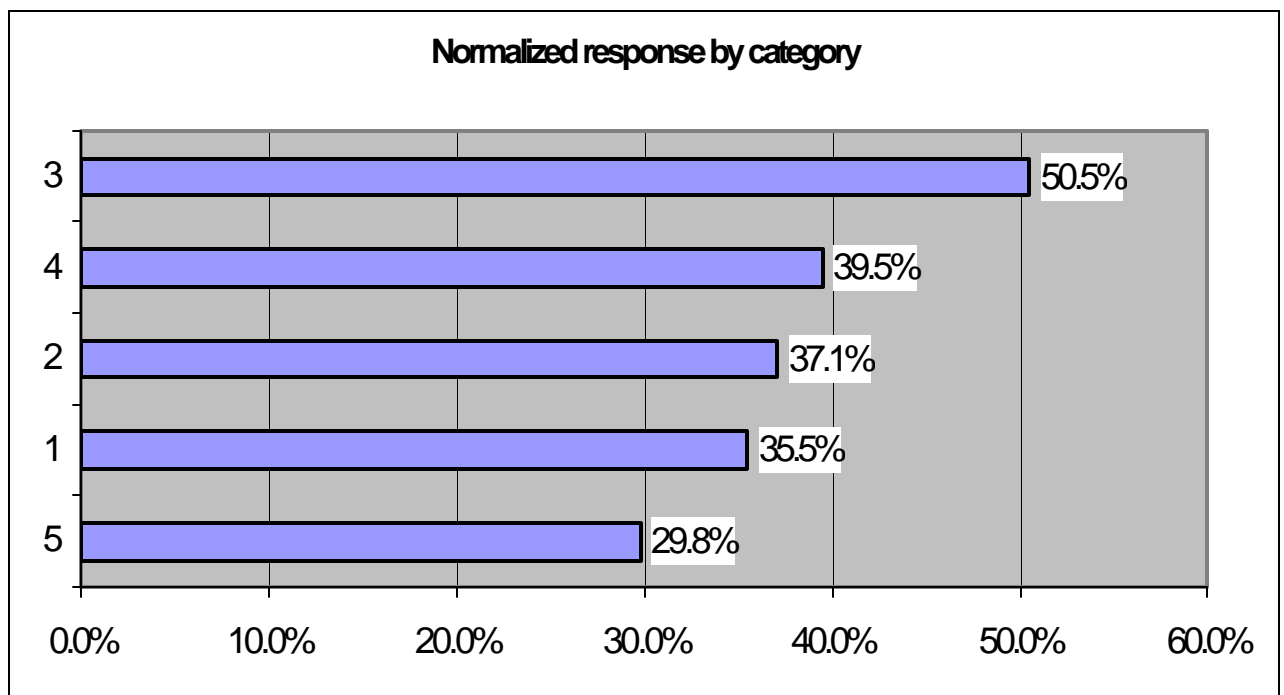
The talk was based on one of the same title given at the Michigan Quality 2001 Conference, to an audience of software quality professionals. The April 9 audience was from a broader quality background, comprising both quality professionals and users of quality. The presentation was tailored to remove most of the material specifically related to software development.

While the presentation was in progress, a survey was given to the attendees, and the results are tabulated here. The response was heavy. Of 47 in attendance, 31 completed responses were returned, or **66%**.

Responses by category

Respondents were asked to check whether their organization met each of eighteen criteria in five categories:

1. Phantom process improvement, whereby an organization just pretends to follow a standard
2. Stalled process improvement, in which a process exists, but is retarded by compromise
3. Impairment in the availability of requirements, where the project goals are described inadequately, vaguely, or tardily
4. Project planning and scheduling issues, characterized by informal or obsolete methodologies
5. Process and product review defects that obstruct the benefit of objective review



Of the five categories presented, the chart shows the relative responses per category. They have been normalized to show the percentage of total possible responses for the category.

Category 3, Availability of requirements, is clearly in the lead at **50.5%**, and is worthy of focus. It is no coincidence that in SEI's Capability Maturity Model [1], the first Key Process Area is Requirements Management. If an organization can get its act together in documenting and reviewing its requirements, it has taken a major step in learning how to become more mature and process-aware in all the other areas.

Responses by question

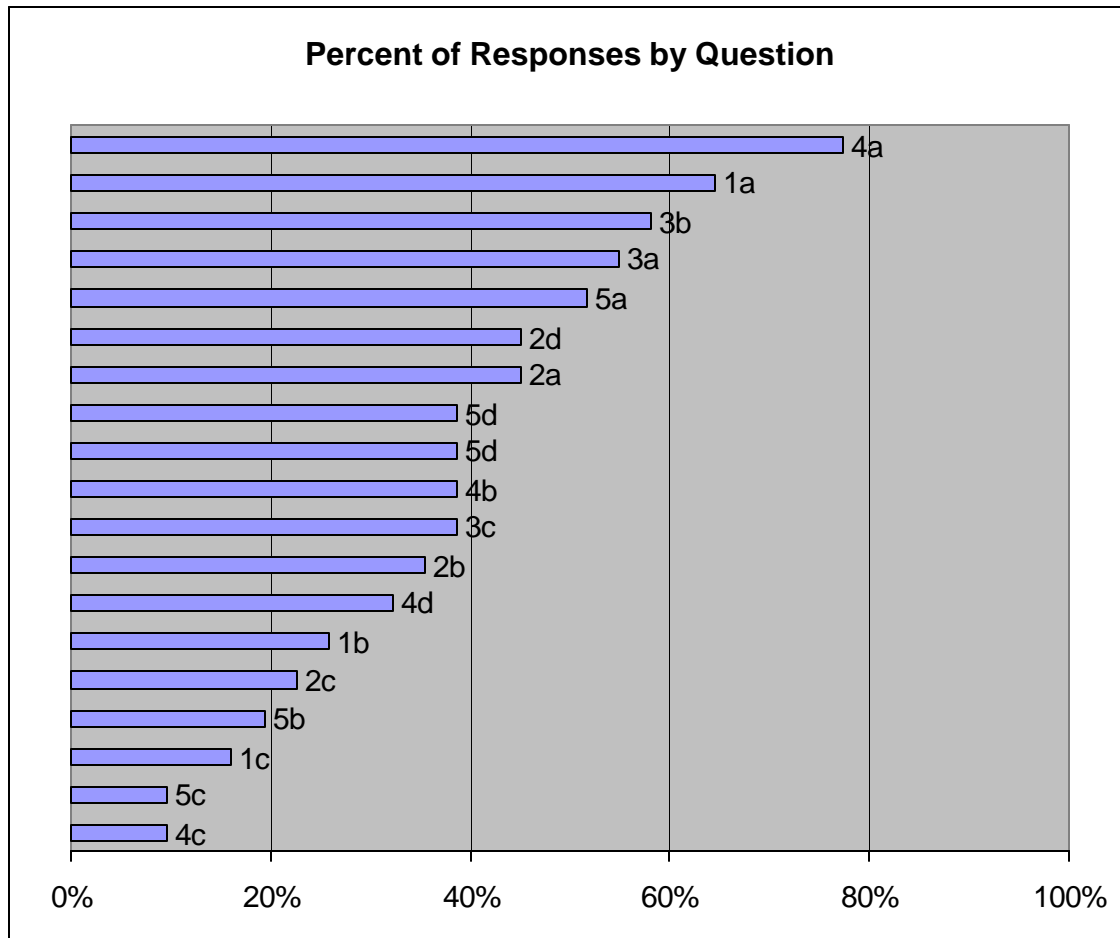
The number of responses per question ranged from 10% to 77%. The Pareto chart on the next page shows the percentage of responses, and the table at the end of this article lists the text of the questions.

Here are the top six responses of the eighteen questions:

Schedules are based on need more than the reality of development cycles (Category 4). More than three-quarters (**77%**) of respondents checked this issue. This means that the ideal of a two-way dialog between customers and developers has yet to be achieved in most organizations. We need to train our internal and external customers to disclose the cost of slippage, so we can make reasonable trades and adjust the resources applied to a particular project. One of the best seminars I ever attended was "How to Develop Products in Half the Time" by Don Reinertsen [2]. He challenges developers and their customers to start a quantifiable dialog about the cost-vs.-revenue basis for scheduling.

Process documents do not reflect changing priorities or practices (Category 1). Nearly two-thirds (**65%**) checked this one. This is often a consequence of creating a process standard "for show", then putting it away until the auditor arrives. A simple fix is as follows: the next time someone asks you, in conversation, to change the way something is done, make it clear that you will be happy to rewrite a paragraph of the process manual to reflect the change. That will not only improve the documentation, but force reflection on the consequences of a particular change. If your organization resists, there is a wonderful book by Price Waterhouse [3] on organizational change.

Specifications are vague and indefinite (Category 3). This was checked by **58%** of respondents. We have all been in a situation where the developmental goals are not clear at the outset, and the chickens don't come home to roost until someone needs to be blamed for project slippage. When faced with such ill-defined requirements, write questions, circulate them to your customers (internal or external), and don't be afraid to challenge the vagueness. It is in the customer's best interest to have the developer know his or her expectations before committing to a development schedule. My own practice is to keep a metric of unresolved requirements (say TBDs per thousand words), and someday I hope to prove that it correlates with late delivery.



Requirements or specifications arrive over a long period of time (Also Category 3). This was checked by **55%**. Closely related to the one above, this one is more difficult to overcome at first. If we all knew the exact outcome of every development project, there would be no need for development projects. We (as quality professionals or users of quality) and our customers need to be honest with ourselves – the purpose of a development project is to answer the question “How can such-and-such be done?” An iterative development plan is one solution, because it honestly admits that at each iteration, more will be known about the outcome. These valuable results can benefit both the customer and the developer as they become available, but both need to decry omniscience at the outset for an iterative plan to be accepted. Change is inevitable – recognizing that fact is a sign of organizational maturity.

Review standards and policy are largely undocumented or are applied on an ad hoc basis (Category 5). More than half (**52%**) checked this. It may show a lack of training in the conduct of reviews, or it may show ordinary fear among staff. The solution to insufficient training is more training at all levels. The solution to fear is full acceptance of genuine process improvement among top management. You have just given me the topic for next year’s presentation: “How to train your boss.” Thank you.

Process improvement activities are regarded as too costly in dollars or effort and It is too easy to exclude projects from process improvement (Both Category 2). These were from the same category, and tied at **45%** of respondents. In this author's opinion, humble or otherwise, both of these are a manifestation of the belief that process improvement is a pure-cost item, made to satisfy auditors and no one else, and should be kept at bay. The truth is simply that quality pays. We can measure the benefit of any improvement program, and if it is not making a profit, it needs to be rewritten or practiced better. Our goal is to make the developers run to us for advice, rather than run and hide when we approach them for an audit.

Correlations

Note that the top six responses came from all five categories. In all respects, the responses were remarkably independent in a statistical sense.

The strongest correlation (**56%**) was between Schedules are based on need more than the reality of development cycles and Review standards and policy are largely undocumented or are applied on an *ad hoc* basis. In the grand scheme of things, 56% is not a very strong correlation, although it is reasonable to expect that an *ad hoc* organization will use obsolete scheduling criteria. See comments accompanying both responses in the previous section.

There were no significant inverse correlations, the greatest being only **31%**.

Conclusion

The original premise was that compromise can interfere with a process improvement effort to the extent that it makes improvement appear more costly and less beneficial than it is capable of being. The survey has borne this out, and has given us a set of priorities for our own efforts at process improvement. If in our own organizations we apply resources in approximate proportion to the above responses, our lives will be better and someone might even thank us.

Methods

The survey was intended to find a hierarchy of needs among those interested in process improvement. Percentage responses should not be construed as representing the general population. Potential source of bias for the survey include:

- The attendees were not drawn from a random population, but one in which the topic was promoted, and no controls were applied
- The presentation was highlighting the very questions that were on the survey, with the author's own slant on them
- All questions were phrased to illustrate an unfavorable outcome
- No attempt was made to randomize the order of questions.

References

1. Mark C. Paulk et al., The Capability Maturity Model: Guidelines for Improving the Software Process. An overview of CMM may also be seen on Software Engineering Institute's Web page, <http://www.sei.cmu.edu/cmm/cmm.html> .
2. Don Reinertsen, <http://www.reinertsenassociates.com> .
3. Price Waterhouse, Better Change: Best Practices for Transforming your Organization.

Appendix - Survey questions and responses

Pct.	No.	Question
77	4a	Schedules are based on need more than the reality of development cycles
65	1a	Process documents do not reflect changing priorities or practices
58	3b	Specifications are vague and indefinite
55	3a	Requirements or specifications arrive over a long period of time
52	5a	Review standards and policy are largely undocumented or are applied on an <i>ad hoc</i> basis
45	2a	Process improvement activities are regarded as too costly in dollars or effort
45	2d	It is too easy to exclude projects from process improvement
39	3c	Specifications contain many "TBDs" ("to be determined")
39	4b	Estimates of development effort are made with unknown methods
39	5d	The opinions expressed in reviews are inhibited or affected by fear
35	2b	Process improvement decisions are made with little or no measurement of cost or effort
32	4d	There is inadequate mid-project review or revision of project plans
26	1b	Process documents have little to do with the process
23	2c	Process improvement is overruled by "the company way"
19	5b	Review teams or process groups are dominated by managers
16	1c	Process documents discourage process improvement
10	4c	(Software development) Effort estimates are not based on product size
10	5c	Review team members are selected to affect the outcome

About the author

Harry E. Kitchen, CSQE, is a software developer and management consultant in private practice. He invites comments and questions about this paper at hkitchen@columbus.rr.com , and you are invited to visit his Web page at <http://www.hkitchen.com> .