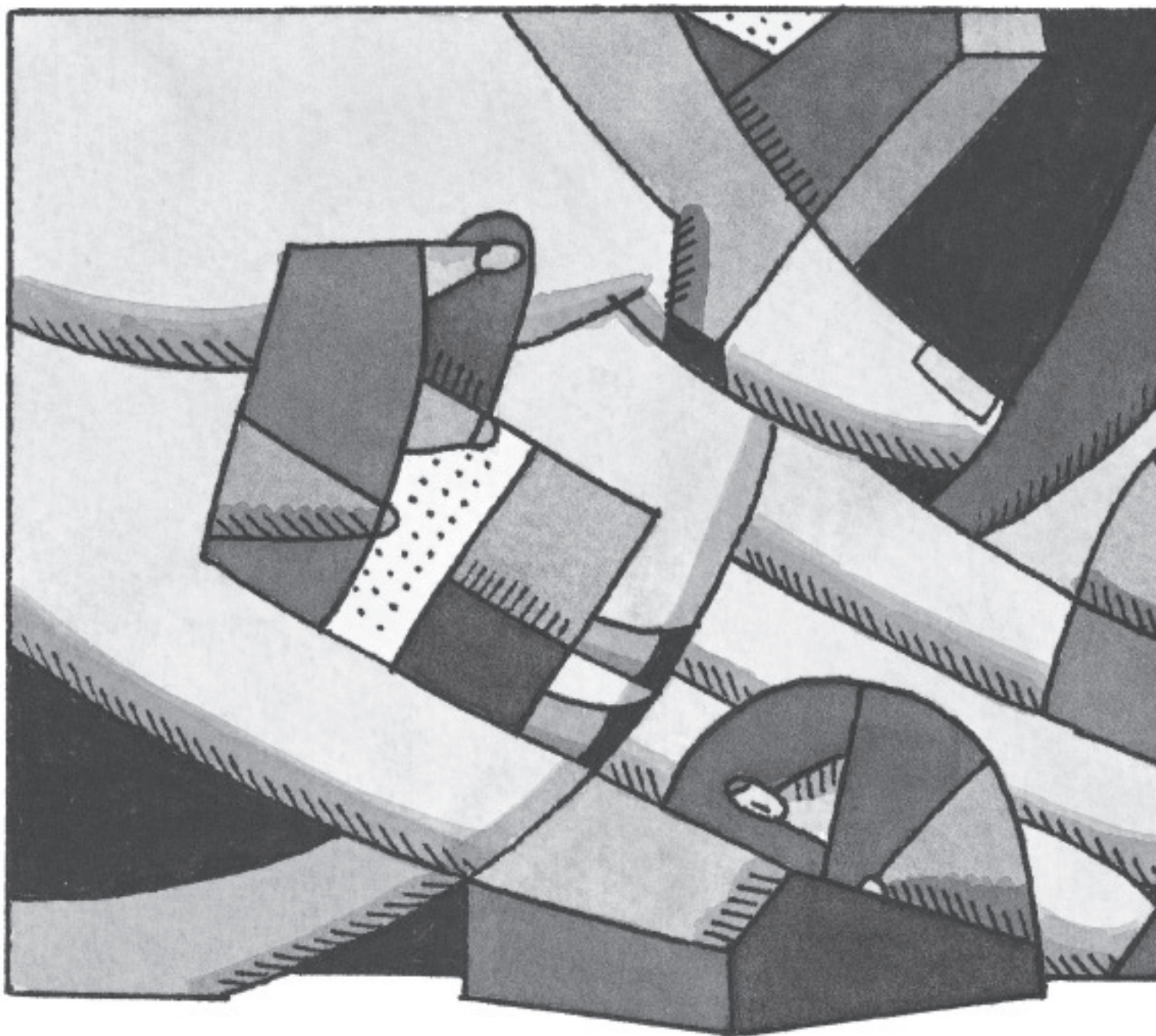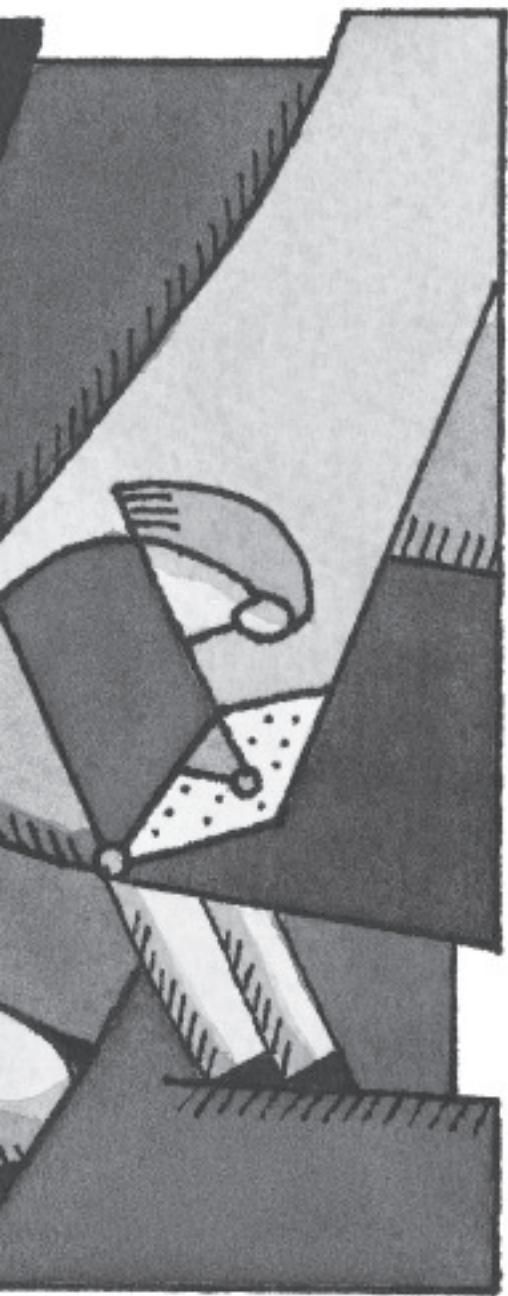# COLLABORATIVE DEVELOPMENT:

A New Culture Affects an Old

Organization

Using a collaborative model for developing IT projects delivered high-quality results with collateral benefits to the organization

By **Jim Phelps** and **Terry Ruzicka**

At the University of Wisconsin–Madison, the Registrar's Office and the Division of Information Technology (DoIT) apply a collaborative development process to joint projects. This model differs from a "waterfall" model in that technical and functional staff work closely to develop requirements, prototypes, and the product throughout its life cycle. The goal is to craft a product closely aligned with service needs and design objectives—and with few bugs.

In this article, we describe our methodology, the collaborative tools, and the organizational structure we used to implement the process. A case study highlights the lessons learned, which we continue to apply as we engage in new initiatives and advocate for the use of this methodology at UW–Madison. We hope that leadership at other institutions will recognize the value of applying this new approach to development of their own IT systems.

Methodology: The New Collaborative Process

Our previous projects employed a waterfall-type development process, in which requirements were gathered by a functional team and then passed to a technical team. The technical group generated cost and time estimates. Once terms were agreed upon, the technical staff would design, build, test, and deliver a product to the functional personnel, with limited communication between the two groups along the way. This methodology pushed the majority of testing and design review to the end of the process, when bugs are most difficult to fix and design changes are most expensive. Compared to collaborative development, waterfall processes delay key communication and decision making until costs—and tensions—are highest.

The collaborative process we now employ is based in part on the management concepts of agile development and model-driven design. *Agile development* values face-to-face collaboration between developers and users in negotiating agreements. It also emphasizes rapid development of prototypes that spiral toward the final product. *Model-driven design* emphasizes the use of standard models (usually in

Unified Modeling Language) to document requirements. By leveraging aspects from both of these processes, we collaboratively develop the models and requirements and begin prototyping the product as a way of finalizing its design and functions.

Continuous process improvement emphasizes regular check-ins with team members where we ask, "How is the project going?" and "How are you doing?" and realignment of the project structure. Improvement comes from frequently evaluating the effectiveness of the team structure and communication models, adjusting both to fill gaps and reduce redundancy in the process. This works well with agile development's use of self-organizing teams, where the project members decide to start up subteams that focus on specific issues. (See the Further Reading sidebar for more information about these concepts.)

A variety of tools can facilitate the collaborative communication built into this methodology, including e-mail lists, an issue-tracking system, a communications portal, a document repository, online calendars, and time-tracking software. Organizational tools include a physical project room for meetings and long-term posting of artifacts, weekly team meetings, and various ad hoc presentations to and by stakeholders, sponsors, and others.

The project leaders found that this new methodology affects the organization in unexpected ways. The following case study explains our experience with these changes and with applying the new, collaborative development process.

Needed: An Online Transcript Ordering System

In spring 2005, UW–Madison began a large project to build a system for ordering official transcripts online. At the time, the Registrar's Office generated approximately 150,000 official transcripts each year. The old transcript request fulfillment system was a labor-intensive, manual process. To streamline the process and allow for online payment, we needed a new ordering system. Because the majority of requests require paper copies of our official transcript and because of the unique format and process for producing the transcript, we decided a solution built in-house was best.

The project scope involved building an online store for ordering official transcripts, with an automated back-end for tracking and printing the transcripts and cover pages. The new transcript system would include options for order entry and tracking by customer service reps, the ability for customers to track their orders, payment by credit card or check (or any payment method in-person), the ability to send transcripts to multiple recipients, inclusion of attachments (such as LSAT applications and special forms for applying to other institutions), and delivery by USPS, UPS Express, or pick-up at a service window.

The transcript system project touched numerous areas in university administration. Key representatives from each office were invited to help design the new system. Similarly, the system's breadth required the expertise of technical personnel from DoIT's middleware, application development, architecture, and security departments. In all, 45 people were involved in the project at some point. This may seem like a lot of people, but because of the nature of the initiative, people flowed in and out of active participation as needed.

The organizational structure created for this project consisted of a leadership group (the Leads Team) and a network of subteams. The project was cosponsored by Joanne Berg, vice provost for enrollment management and registrar, and Diane Mann, DoIT's director of application development and integration. The Leads Team included the leaders of each subteam, plus resource managers from the Registrar's Office and DoIT. The 11 subteams were Accounting, Business Process, Operating Procedures, Authentication/Authorization, Cost of Transcript, Payment Processing, Printing and Distribution, Storefront, Tracking, Development Environment, and System Testing. Each subteam had a mix of functional and technical members, and memberships often overlapped.

Further Reading

Agile Development Process:

Scott Ambler, http://www.ambysoft.com/onlineWritings.html

Martin Fowler, http://martinfowler.com/articles/newMethodology.html

Wikipedia, http://en.wikipedia.org/wiki/Agile_software_development

Model-Driven Design:

Object Management Group, http://www.omg.org./mda/

Scott Ambler, http://www.agilemodeling.com/essays/agileMDA.htm

Wikipedia, http://en.wikipedia.org/wiki/Model-driven_architecture

Object Management Group, Unified Modeling Language Resource Page, http://www.uml.org/

Object Management Group, Business Process Management Initiative, Business Process Modeling Notation (BPMN) Information, http://www.bpmn.org/

Continuous Process Improvement:

Wikipedia, http://en.wikipedia.org/wiki/W._Edwards_Deming

Joiner, Brian L., *Fourth Generation Management—The New Business Consciousness* (New York: McGraw-Hill, 1994)

Scherkenbach, William W., Deming's Road to Continual Improvement (SPC Press, 1991)
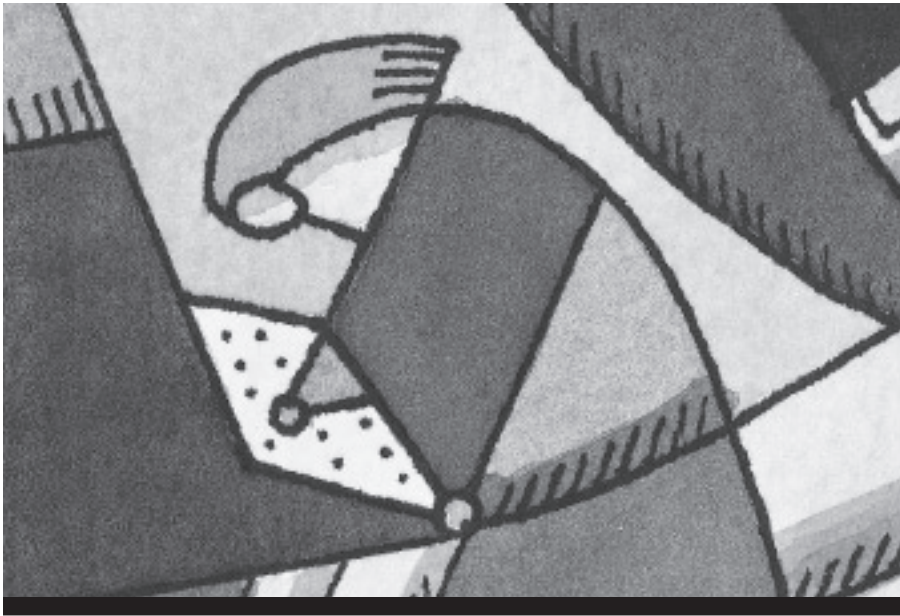
Staff have an opportunity to learn new skills and develop professionally, and working closely together builds lasting communication bridges that benefit the organization

Impacts of the Collaborative Process

We believe the collaborative process improves over the old waterfall process in several ways. Most importantly, it brings the functional and technical people together, which leads to sharing issues and jointly solving problems. The two areas gain a mutual understanding of each other's constraints and capabilities. In addition, staff have an opportunity to learn new skills and develop professionally, and working closely together builds lasting communication bridges that benefit the organization. Further, the transcript delivery product launched successfully on a tight timeline despite competing projects. Finally, a key indicator of the project's success was the effective response to a critical bug that showed up after the new system went live—people involved in the project had a deep understanding of the application, communication flowed easily, and the problem was fixed quickly.

During the implementation of this new collaborative process, we ran into some surprises and learned some useful lessons. The first thing we learned was that the impacts of the collaborative process reach deep within the organization. Issues of culture and communication must be carefully managed. We identified a series of corrective steps, listed as "Fixes" below. These issues and fixes fall into four broad categories: culture change, communications, decision making, and expectation management.

Culture Change

According to Marilyn McIntyre, director of Information Services in the Registrar's Office and functional project resource manager on the Leads Team, "We took significantly different cultures and mixed them into a whole new culture." This shift is not an incidental or negligible result of the methodology—it is a major benefit.

Staff have a strong sense of their jobs and their daily activities. They know what to expect and what is expected of them. When you ask them to step outside of their known space, they often feel uncomfortable. When teams consist of people all outside their known space, then the whole team can feel at a loss.

Most people tend to "go quiet" in new situations. When you are the new person in a meeting, you might lis-

ten for the first session or two until you feel comfortable with the people around the table and you understand where you can contribute safely. In the collaborative process, the team leaders initiate conversations that encourage people to settle into their new roles and the new situations.

Establishing open, collaborative communication in the teams is critical. Constant attention must be paid to not just allowing but *encouraging* people to express their ideas and opinions. The goal of the process is to put ideas on the table from both the functional and technical groups and to come to a decision that balances all aspects of the design. This is difficult to do if people aren't comfortable in their new roles. To quote one team member, "I was walking on egg shells. I didn't know what I could say or to whom."

Joanne Berg recalled, "I kept hearing that the…transcript service staff were feeling that everything was happening around them. Things were really changing and they weren't in control. There was a huge sense of uncertainty."

The evening before a scheduled meeting, she continued,

I was at home wondering what I would do about this group. I turn to music when I need inspiration. I was listening to Joyful Noise's version of *Pachelbel's Canon*. This version has surprising twists and a bit of discord in it. The next day, I had the group sit down and listen to this piece. It was a long piece of music. The team was anxious and they were twitchy. They all wanted something to happen. They weren't really sure what this was about. After the piece ended, I asked them, "Why did I make you listen to this music?" They all started talking about their discomfort and anxieties. I filled my white board with notes. They then understood that discord and angst [are] a natural part of the process.

A new understanding of the team's source of anxiety and their changing roles came out of this meeting. Later, one team member said, "Joyful Noise

really represented the angst that was present in the office." Their discomfort came from feeling that the way they had done their jobs for years was no longer important. They felt marginalized.

In the IT arena, the collaborative process also precipitated dramatic changes, and the intense teamwork was not initially seen as desirable. The technical staff wasn't used to, or comfortable with, the continuous feedback in the design phase. They struggled with the functional staff's roles on teams in this collaborative development process. As one technologist remarked, "I wished we could have gone back to having a single designer."

Over time both the functional and technical staff came to understand that their participation was critical to the quality of the overall process. They began to appreciate the value of their new roles and realized that they were moving to a new and important way of working. Once they came to grips with their discomfort and better understood their new roles and the value each group brought to the endeavor, they became more engaged in the overall process. This helped the project teams coalesce.

Team members at all levels need to fully understand what is expected of them. Think of it as training personnel for new positions—you should provide the same level of management, nurturing, and communication. "We took people whose job was to do one thing they knew well, and we said go over here and do something completely different," explained McIntyre. "That made them very uncomfortable and unsure about what was expected."

Fix: Have lots of conversations at every level—one-on-one, small teams, and larger work groups—to discuss the process, people's roles, how they might feel unsure, and how this is really like starting a new job. Help the team members become comfortable in their discomfort.

Fix: Check-in regularly with team members to assess who might need attention or mentoring at any given time. Use this as

an opportunity to help staff learn new skills and develop professionally.

Staff need to feel secure to participate fully and to talk through their discomfort. Encourage a culture in which participants understand the importance of speaking up and raising issues. As one person said, "The participants are obliged to raise issues in the process." Leaders need to instill the "anyone can stop the assembly line" mentality.

Communications

Team members expressed their initial confusion and frustration thus:

There were many times that I was in a meeting and I was struggling with why I was there.

I had this constant barrage of e-mail, most of which I had nothing to do with.

The disruption of workers from their everyday culture means, among other things, that their communication channels are redefined. Their instincts about whom to talk to are no longer valid. This lack of a known communication structure led us to two well-intentioned but mistaken approaches. The first involved team membership, and the second involved communication channels.

Some of our teams had too many members, which led to more confusion about roles and responsibilities. We had put some people on teams as a way of making sure they were in the communication loop. For many people, attending meetings does not add value to the team, nor does it guarantee their participation. They need to hear about key decisions and have input on issues where their expertise is required. To them, the meetings became a "time sink" and led to frustration. Their disengagement can hamper the progress of the team and affect morale.

Other participants, especially the technologists, view any meeting as a distraction from their "real" work. For them, the key is to get them to understand the value of their participation. The balancing act is between participa-

tion and communication. Some don't need to participate but do need be in the communication loop. Others need to be convinced that their participation is critical—just being informed is not enough.

Fix: Think carefully about the role of each subteam and who is a necessary participant. Don't use team membership in place of a communication channel.

*Fix: Communicate each member's value and role on each team. Make sure they understand that this is real work too.*

Since communication is key to collaboration, we assumed that more was better. We learned that "more" can soon lead to "ignore," especially with e-mail lists. Many people were on multiple teams, so as each team began sending messages out to each list, sometimes copying multiple lists, team members were drowned in redundant and unfocused e-mail. The e-mail lists should be as dynamic as the teams and the project. Removing people from e-mail lists is as important as adding new people.

Fix: Monitor communication channels and tools, especially e-mail lists. Evaluate their effectiveness often. Add or remove members or lists as needed. Check with people, especially those on many subteams. What makes sense for a communication structure at the start may not align with activities later in the project.

Fix: Structure e-mail lists around topics of interest, not just team memberships.

Fix: Each team needs to have one person responsible for communicating to others. These people need to act as filters, stopping unnecessary noise while channeling issues to the right people. Their role needs to be well defined and understood by all.

Another communication aspect has to do with individual personalities. Some projects require the presence of a person who has specific expertise but not necessarily the interpersonal skills desired. One person's quip can spark unnecessary panic in others. Rumors
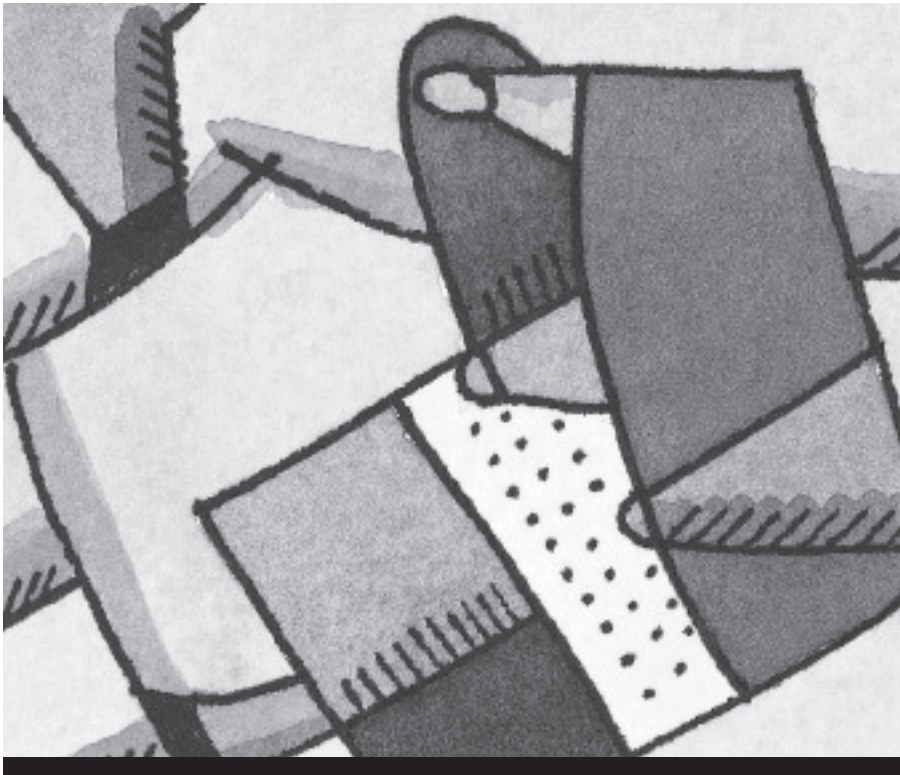
Achieving the optimal balance of communication and participation in a project is challenging both individually and situationally

and misunderstandings tend to take on a life of their own, especially at times of high stress.

According to Bob Mayville, technical resource manager for the project, "A person might be seen as flippant by some. It might also be that they thought we were underselling a problem, and this was their way of getting it on the table."

There should be communication about personality quirks that might cause challenges and appropriate ways to mitigate misunderstandings. It is also important to seek the underlying truth in a message even if delivered in a less-than-tactful manner.

Fix: Have a channel for fact checking: "If you hear a statement or rumor, go to Sally and she will verify the statement."

Fix: At the leadership level (team leads, project leads, and sponsors), have a frank discussion about the people participating.

Keep the comments as kind as possible, but relay the message that issues might come up. For example, "Chris likes to pop off just to stir the pot. If Chris says something, check with me and I'll verify what was said." There is a delicate balance here between poisoning the process for a person and being proactive to avoid problems.

Fix: A trusted facilitator is critical to the process. "Mostly, I just listened and let them vent their angst," said one facilitator. "Occasionally, I would escalate issues to the sponsors so that a longer-term strategy could be developed to deal with the issues."

Achieving the optimal balance of communication and participation in a project is challenging both individually and situationally. In the planning stage, think carefully about what kind of involvement is needed, from whom, and at what points in the project. Be

clear about who is responsible for what aspects of the communication flow, and when. Staff also may need a trusted third party to whom they can speak frankly. This third party should listen, document problems, and escalate issues as needed. Further, expect that your communication channels, teams, and team memberships will be fluid throughout the project. The communication framework is not static; it needs to be watched, nurtured, and realigned throughout the project life cycle.

Decision Making

One goal of a collaborative process is to resolve issues and test decisions early. You want to get ideas on the table, identify possible solutions, do a S.W.O.T. (Strengths, Weaknesses, Opportunities, Threats) analysis, and take the information back to the appropriate teams. The teams then decide which solution(s) to prototype. If the prototype fails, at least it failed early in the process, when changes are easy—which is much more desirable than pushing all of the design decisions until late in the process. If decisions are delayed, there may not be time or resources to do a good analysis of the solutions or to prototype various options.

In our project, the teams were at times uncertain of their roles in the decision-making process. To paraphrase a common issue that we heard in the post-project review, "We would discuss the same issue in multiple team meetings with no one taking ownership." People hadn't been through the process before, so they didn't understand which issues fell under their team's responsibility. Because teams were unsure, they discussed and documented the issue but then passed it off to another team. The next team would do the same, and we ended up with issues swirling throughout the various teams.

Each subteam should know its role in the overall project. The group must understand the issues for which it is responsible and take ownership for suggesting solutions. Team leads must realize that they will be held responsible for ensuring follow-through.

*Fix: Make sure to charge your teams and subteams with the areas for which they are responsible. Develop charters and define roles and responsibilities. **Verify** these frequently throughout the project.*

Fix: Make sure the teams understand that they are empowered and expected to make decisions. Ensure that team leads understand their responsibility in this process.

A lesson learned is to capture issues and assign decision-making responsibility for the issue to a specific team. Create a "trap" to capture issues floating between teams and a method for handing the responsibility to a given subteam.

Fix: Document issues in an issue log and revisit them regularly. Watch for issues that languish. Follow up with the subteam responsible for the decision.

We also found that our Leads meetings were being used for reporting status rather than dealing with issues and making decisions. This distracted the team from its main goal of capturing, routing, and resolving the issues that the subteams hadn't, or couldn't, deal with.

Fix: Have regular, written status reports, and reserve Leads' meeting time for dealing with issues, interventions, and decisions.

Similar to the work needed to bring people into the process, you must work with the subteams for them to understand the process. The discomfort around the role of subteams in the project and the overall decision-making process echoes the discomfort of individuals taking on new roles in the project. Each subteam must understand its role and must be nurtured through the angst and uncertainty.

Expectation Management

Another significant issue that arose was the need for expectation management. In a project of this reach, where brainstorming and collaboration are key, many people will propose good ideas that won't make it into version 1.0 or even the final product. For example, we spoke with the transcript service staff about printing solutions. One solution involved an elaborate automated distribution system. It turned out that the cost of such a system was well beyond the budget for the project. For some staff, we raised false hopes just by discussing the possibility.

Also, a project of this size will have iterations of implementation. Version 1.0 is not the "final" version. Expectations around what will be delivered in 1.0 must be carefully managed, and the reasons behind the decisions must be clearly communicated. A lot of assumptions will need to be addressed. "Leadership has to work to surface these assumptions throughout the whole project life cycle," commented Carol Gosenheimer, project lead and communication manager. "They also have to be able to have the difficult conversations about why something is not being delivered."

Fix: Set expectations up front. Talk about the fact that good ideas may not make it into the product for a variety of reasons. Explain that talking about a solution is different from promising delivery of the solution.

Fix: Explain that there are phases of implementation and that no 1.0 product is ever perfect or final. Explain, up front, the decision-making process for inclusion of features in the product.

The cornerstones of the collaborative process are brainstorming, openness, and inclusion. The methodology brings functional and technical personnel together to find the best solution to the problem at hand. The key ideas that come up when you talk about the collaborative process are involvement, outreach, creativity, and exploration of possibilities. This process can be exciting and energizing for many participants. Remember, though, that many of the team members are still in a new "discomfort zone," although they may have overcome their angst and put their ideas on the table.

People become emotionally invested in the process and their ideas. The choice not to include their suggestions can confirm their anxieties or doubts about the process.

Fix: Encourage staff to get all ideas on the table. Manage expectations by explaining that compromises likely will be needed among the many possibilities.

Both functional and technical staff bring their own assumptions to the project. People will assume their ideas are within the project's scope unless they are told otherwise. They will assume that everything discussed will be part of the product. Establish early on that you are looking at a product with a life cycle. Some functions will be delivered in version 1.0 and others will be delivered later or not at all.

Fix: Change the discussion to one about a product roadmap and product life cycle.

Fix: Communicate the methods used for putting things on the roadmap and ranking them for delivery.

This problem occurred in the transcript project around authentication of former UW–Madison students: Current students can authenticate with a NetID and password or CampusID and PIN, but former students do not have a similar authentication path. We looked at several prototypes for authentication based on biographic data. These features were not included in version 1.0 of the product for a variety of reasons. This instance of expectation management worked well because everyone understood the decision-making process and the rationale behind the decision. Each participant could come to the same decision given the data. The key is to have a defined process for making decisions and to have open communications about them.

Fix: Define the process by which features will be determined to be in or out of scope for the product.

Fix: Communicate the reasons for decisions as well as the framework for the analysis followed.

Other features may fall out due to failures in testing or in migration to production. Some things might work well in the test but not scale to a production environment. Other features may not adapt to meet accessibility issues or other requirements. People who see these features in an early prototype might believe they are now in the product even though testing and migration can move features off the roadmap. You need to have a well-understood process for testing the product and for communicating impacts of testing on the product's scope.

Fix: Establish functional testing protocols and rules for success early.

Why We Believe in This Process

A bug showed up in the production system soon after launch. Because we had used this collaborative process to design and build the application, the bridges between the groups already existed. Communications flowed easily and the bug was squashed quickly. As Leah Meicher, technical lead for the Registrar's Office, put it, "When a bug hit in October [2006], we had learned our lessons about how to collaborate. We really came together to solve the problem."

When an issue becomes apparent, the deeper understanding on both the functional and technical sides about the application, who is responsible for which parts, and how the application design was chosen will accelerate the bug-repair process. The functional staff will know whom to call and how to communicate about the system problems. The technical staff will understand how the system should work and why it should work that way. These connections and understandings smooth the repair process greatly.

Another important change is that people now view participation in projects as something they want to do rather than a burden that distracts from real work. We are well into phase two on this project, and staff continue to be actively involved. People participate because they want to be part of the decision-making process and to work closely with colleagues from different areas of the organization. They have developed a high level of trust that their input will be heard. These factors add to the overall value and experience of being part of the team.

The functional personnel also have a better understanding of the technical trade-offs. Training the functional staff in the new system is much easier because they were present in the design and testing phases. They understand the reasons behind product decisions because they participated in the discussions.

Finally, the technical people have a deeper understanding of the business processes they are trying to improve. They understand the current pain and the hopeful gain they are making. This adds value to their work. Rather than building some black box and then questioning change requests, they build an application to help their teammates and understand the reasons for changes. Being part of the discussions and decision-making process lets them make better decisions while designing the application.

The Registrar's Office and DoIT are currently collaborating on a new project, to which we are applying the successes and lessons learned from the transcript project. From the outset we put the issues on the table, and we continue to communicate with the team members about the process and what to expect. We have stressed the importance of their participation and open collaborative communication. A particular benefit was the strength of the relationships built by using this methodology. These relationships carry over into all other aspects of the organization.

We believe the collaborative process is a significant improvement over a waterfall process. It raises new challenges with culture change and communication, but the benefits to the institution are well worth the effort. Collaboration yields a better product

while benefiting the organization and the individual team members. The collaborative process has become the new standard for projects between the Registrar's Office and DoIT. *e*

Jim Phelps is Senior IT Architect in the Division of Information Technology, and Terry Ruzicka is Special Assistant to the Vice Provost for Enrollment Management, at the University of Wisconsin–Madison.