

Managing a Product Development Team: Part II

Growing the Team

Larry Barrett • Ken Lehtonen

As noted in Part I, one of the original constraints of the Hubble Space Telescope (HST) project was to use a legacy software staff. Despite our doubts about the technical currency of this team, they embodied the Hubble domain knowledge that was both critical and necessary to development of the new control center. Since a decision had been made to develop the control center using object oriented (OO) technology targeting a Unix® environment, the challenge became one of “converting” as many legacy programmers as possible to the object management technology (OMT) methodology. One of the greatest obstacles was convincing the team of both the personal and programmatic benefit to transitioning their design skills. This the management team did through a series of technical briefings that demonstrated the additional capabilities and flexibility of the OO technologies. These briefings served to convince senior members of the legacy staff that the HST project would benefit from employing modern software design principles, such as OO programming, in order to develop a system that had to last at least another decade.

At this point, the green light was given by HST senior management for a hiring binge to acquire additional staff with key OO and C++/Java skills. A major objective was to use these new team members to bring the legacy staff up to the necessary level of technical capability. This was accomplished through the following multi-faceted training approach.

In Part I of this article (Defense AT&L, March-April 2004), the authors presented several of the management issues hampering the effective startup of a project to re-engineer the aging Hubble Space Telescope (HST) ground system. The initial challenges facing the management team were significant. The schedule was aggressive and non-negotiable. The team had to maximize use of existing maintenance personnel to undertake new development, while at the same time creating “new and better ways of doing business” that required discarding business processes that were ingrained in the user community. The primary project management goal became to eliminate the sources of inefficiency on the project by building a culture that fostered an atmosphere of cooperation and that was success-oriented. Several of the actions taken to overcome these issues were presented in the previous article. Part II builds upon these, discusses the methods used to build a cohesive, synergistic team environment, and presents several implementation strategies that were used successfully on the project.

We initiated a massive, just-in-time training effort for the whole development team (at this point the architecture of the new Hubble control center system was just about completed). We brought in house some of the top OO trainers in the nation to provide targeted training. The traditional training approach was reversed by first training the team in the C++ language specifics (they were already familiar with FORTRAN and in some cases the C language) and then providing on-site training courses in generalized OO analysis and OO design. This approach worked better because the staff were more comfortable with implementation technologies from which they could then abstract the methodological underpinnings.

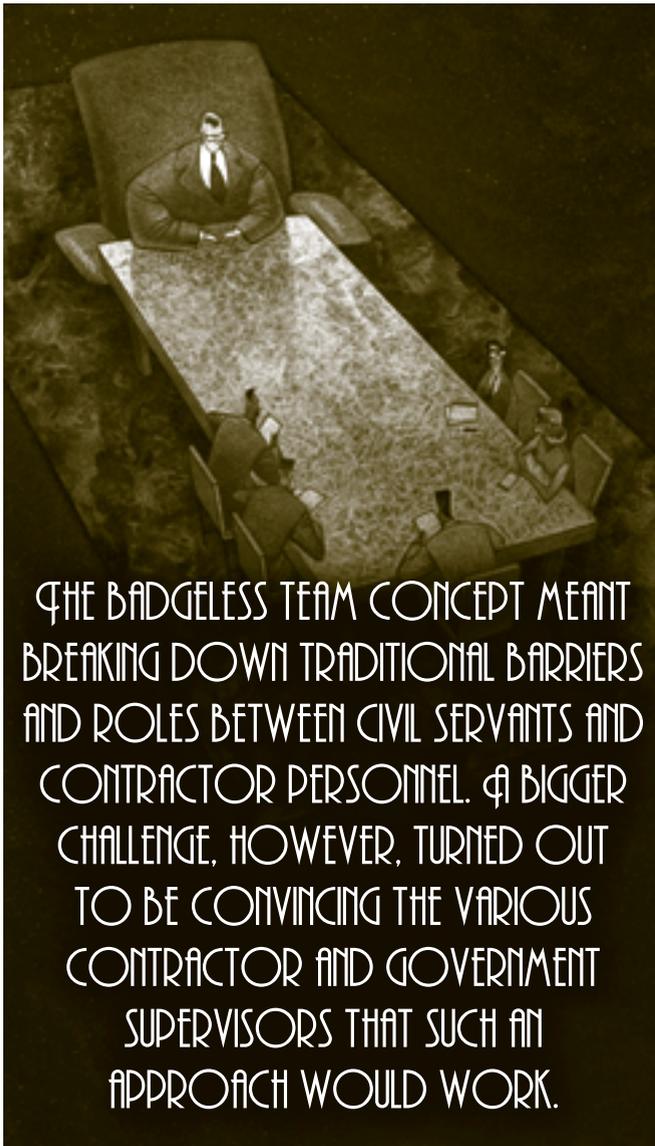
The vendors of the major commercial off-the-shelf (COTS) products that were selected into the architecture of the new system were willing to train the team in the specifics of their products. To supplement the standard classroom training, technical consultants (see next page) were brought in; they not only mentored the team, but

were exemplar software developers in their own right.

To improve our contacts with outside industry, the staff were encouraged to attend technical conferences and to present papers or provide demonstrations of the Control Center System (CCS) technologies under development.

Internal technical demonstrations of mature software were scheduled not only for the CCS staff, but for Goddard senior management as well. This was not only a

Barrett is the chief systems engineer for the HST control center system. He is primarily responsible for system architecture oversight and technical risk management. Lehtonen has over 35 years of experience in software engineering. His experience includes designing, implementing, testing, and managing a wide variety of mission software applications for NASA.



THE BADGELESS TEAM CONCEPT MEANT BREAKING DOWN TRADITIONAL BARRIERS AND ROLES BETWEEN CIVIL SERVANTS AND CONTRACTOR PERSONNEL. A BIGGER CHALLENGE, HOWEVER, TURNED OUT TO BE CONVINCING THE VARIOUS CONTRACTOR AND GOVERNMENT SUPERVISORS THAT SUCH AN APPROACH WOULD WORK.

morale boost for the presenters, but provided another means for communicating technical information throughout the development team and to the stakeholders back at Goddard. A side benefit of the demonstrations was that they helped identify specific technical skills of project teammates to the rest of the staff.

In spite of the focused training effort, it became apparent during our design activities that we needed to sprinkle our emerging OO team with some experienced on-site OO and C++ expertise. With senior management's approval and as part of the CCS management philosophy to engage outside expertise, we contracted with an organization expert in OO development to provide a small number of on-site consultants. To avoid the traditional (often contentious) consultant-client relationship, the management team decided to assimilate these consultants rapidly into our own evolving culture and make them an active part of the CCS team. For their part, the consultants provided mentoring services on a one-to-one basis

in analysis and design, C++ language skills, and software debugging. We also made them an integral part of the development team by assigning them key pieces of application software to design and code. (One of the consultants was tasked to lead the Middleware team until a permanent replacement could be found.) The consultants were instrumental to the successful, on-time delivery of the Release 1 system and provided a significant return on investment for their services.

Developing a Cohesive, Cooperative Culture: the Badgeless Team

By being physically separate from the Goddard main-stream, the product development team (PDT) was able to develop its own unique management culture and style to fit the environment and its goals. One of the first management goals was to replace the typical atmosphere of competition and animosity between contractors and the customer with a more universally cooperative environment. This transformation was effected by restructuring the team dynamics to implement and expand upon the concept of a "badgeless team."

The badgeless team concept meant breaking down traditional barriers and roles—often contractual—between civil servants (HST being a government-run project) and contractor personnel, as well as among a variety of support contractors, since there were eventually over a dozen different companies represented on this PDT. A bigger challenge, however, turned out to be convincing the various contractor and government supervisors that such an approach would work. In practice, there were civil servants reporting to contractor personnel; contractors reporting to contractors of the same company; and contractors reporting to contractors of different companies. This represented a radical departure from what senior NASA management viewed as the way civil servants and contractors were supposed to relate to each other. It should be noted that this was an evolutionary process, since not everyone on the CCS PDT was comfortable with this new management philosophy, and some chose to leave the project.

In retrospect, this management strategy became one of the main reasons the team was so successful. The emphasis on technical achievements and shared vision, along with a tight focus on the CCS goals (rather than on which particular company should get the credit for the work accomplished), created a unique situation. The ultimate goal was to erase from people's mindset the process of going through "channels." Everything you needed to get your job done was resident at the collocation facility (lovingly referred to by the staff simply as "Colo"). Again, the relative physical isolation enabled the staff to significantly reduce, but not entirely eliminate, traditional corporate politics and jurisdictional disputes that had previously hindered close, tech-



TO ACHIEVE A COHESIVE,
COOPERATIVE CULTURE,
THE MANAGEMENT TEAM
RECOGNIZED EARLY ON THAT
THE INTERNAL NAYSAYERS
NEEDED EITHER TO BE
CONVERTED OR TO BE
STRONGLY ENCOURAGED
TO LEAVE THE PROJECT.

nical exchanges and cooperation between different companies working on the same project.

To achieve this cohesive, cooperative culture, the management team recognized early on that the internal naysayers needed either to be converted or to be strongly encouraged to leave the project. Teamwork and the free exchange of ideas were to be the hallmarks of this project. Over time, these radical ideas bore fruit as team members freely circulated around the building and became comfortable creating ad hoc teams in the hallways. There was a high measure of trust between management and team personnel as well as between and among the individual teams. The overall collegial, community atmosphere allowed all members of the CCS project to excel and exceed expectations from both a technical and personal perspective.

Management Principles: Implementation Strategies

The following management principles (listed in no significant order) served to sustain the high productivity environment. Some of these principles are obvious, some are espoused in current management science texts, and some will work only in a collocated environment.

Use integrated product teams to provide short-term results

A meta-goal of every project is to make good design decisions and to develop the corresponding products as quickly as possible. The CCS PDT management selected from the entire organization those persons who could

best produce a particular product; assembled them into a small team; gave them the authority to make the necessary decisions; and when the product was completed, returned them to their core technical teams.

Use the 80/20 rule

As is the case with most projects, the CCS PDT existed in a very dynamic environment where technology was rapidly evolving, and user requirements were negotiable. Recognizing this, a decision was made to expedite the decision-making process and to avoid “paralysis by analysis” by employing the 80/20 rule. For example, if a COTS product could be found that satisfied at least 80 percent of the target user requirements, then feedback from the users would be solicited to determine if this was adequate. The process was driven by the understanding that not all user requirements are equal, and thus, implementation of the least important 20 percent can often be deferred, sometimes indefinitely. This process also served to keep the user community involved in critical design decisions so they remained part of the solution.

Establish proof of concept and/or prototyping teams

Early on, the core technical teams were tasked with performing risk-mitigation activities while the final architecture of the control center was being hammered out. (Remember that the team was originally front-loaded with a legacy software staff.) The proof-of-concept (POC) team was instrumental in identifying and demonstrating promising new technologies, such as Java applets, collaborative tools, and COTS packages. The results were fed back to

the top-down architecture team to help justify and substantiate the proposed control center architecture. This served as an excellent risk-mitigation activity by introducing the staff to a significant number of new (and sometimes unproven) technologies. One of the PDT's primary objectives was to leverage COTS hardware and software solutions as much as feasible; and thus, many of the teams worked to prototype these packages in an environment as close as possible to that envisioned for the actual control center. Out of these prototyping activities emerged a suite of commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) solutions that was later integrated into the control center design, with the added benefit of reducing both risk and implementation time.

Implement a "rewards and awards" program

After each successful delivery of a control center system release, the project lead acknowledged each individual who contributed to that release with a KUDOS® Brand candy bar. These informal rewards were so well received that team members came to expect a visit right after each software delivery. It was one strategy that cost so little but paid out with immeasurable returns. The upper management team at Goddard was also very supportive of

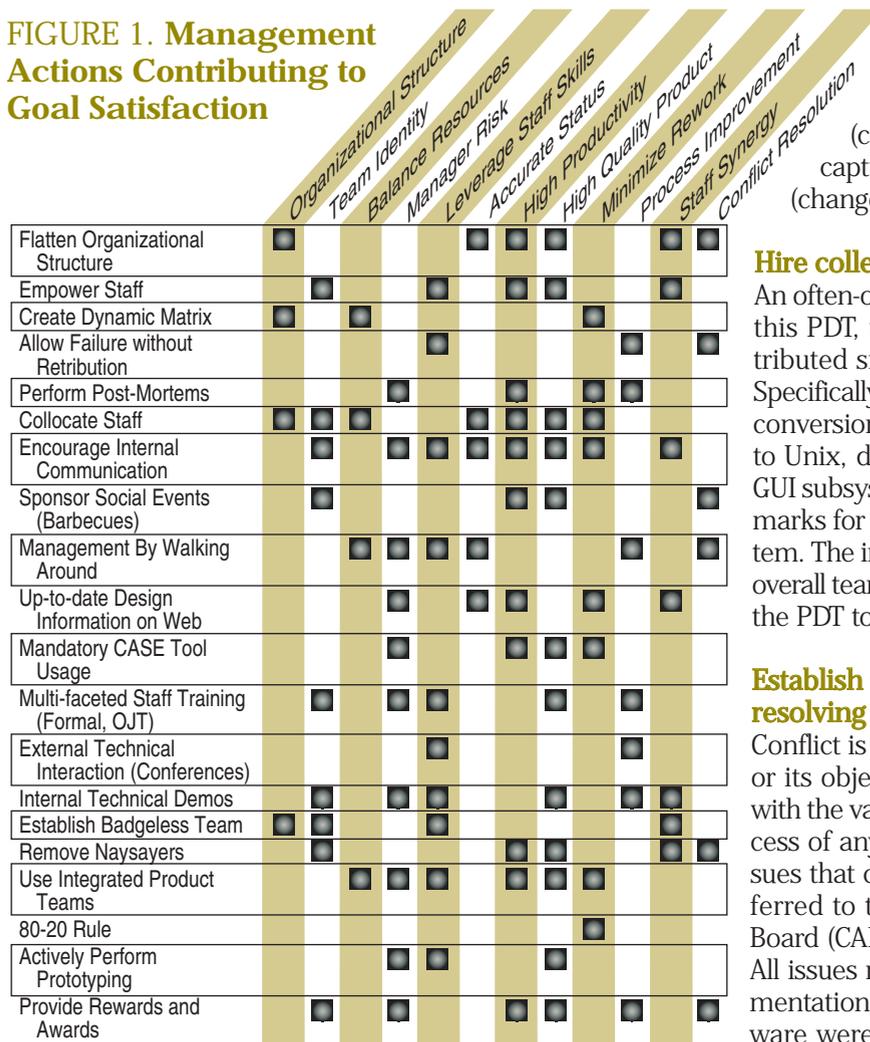
both individual and team efforts. Instead of just funding the prime contractor's award fee, management funded an incentive program that rewarded the Hubble control center team members with bonus checks upon a successful software delivery

Integrate and elevate traditionally background activities into the main software development cycle

The PDT recognized the importance and value of traditional support functions to the successful development and deployment of the control center system. Four examples illustrate this:

- The infrastructure team provided the systems administration, networking, and hardware expertise necessary to define the overall system topology and operations concept.
- The quality assurance team was responsible for ensuring that processes were followed and that design and coding standards were adhered to during all phases of development.
- The methodology team was responsible for tailoring and maintaining the CASE tool used to capture all the design information for the developers.
 - The configuration and change management team developed the electronic tools necessary to support our software baseline control process (configuration management) and the rapid capture and dissemination of problem reports (change management).

FIGURE 1. Management Actions Contributing to Goal Satisfaction



Hire college students for the summer

An often-overlooked area that paid big dividends for this PDT, three summer-hire college interns contributed significantly to the development process. Specifically, these summer interns contributed to the conversion of the command subsystem from VMS to Unix, developed and tested Java applets for the GUI subsystem, and developed performance benchmarks for a newly procured tape-based archive system. The interns were treated as full members of the overall team, were challenged technically, and helped the PDT to maintain an optimal skill mix.

Establish a mechanism for detecting and resolving conflict as quickly as possible

Conflict is inevitable no matter the size of the team or its objectives. Establishing mechanisms to deal with the various forms of conflict is critical to the success of any team. In this case, specific technical issues that cut across core team boundaries were referred to the Control Center System Architecture Board (CAB), chaired by the lead systems engineer. All issues related to the architecture, design, implementation, and correction of the control center software were also referred to the CAB for resolution.

DESPITE THE PROGRESS MADE OVER THE LAST 25 YEARS IN ADVANCING THE STATE OF SYSTEM AND SOFTWARE ENGINEERING PRACTICE, PROJECT SUCCESS STILL COMES DOWN TO PEOPLE.



Intra-team conflicts were expected to be resolved within the specific core team boundaries. At any time, a member of a core team could refer unresolved non-technical conflicts directly to the PMT. In such cases, the staff member's company supervisor could be included in the process to ensure a timely and equitable resolution.

Final Remarks

Because of the relative isolation from its predecessor culture, the Hubble Control Center System PDT management team was granted a great degree of latitude in applying unconventional management techniques. The goals of the management team were no different from those of most systems development projects:

- To establish an organizational structure that provides the right level of control without impeding progress

- To establish and maintain a high level of morale that fosters a team identity
- To allocate project resources in a balanced manner
- To intelligently manage technical and non-technical (e.g., schedule, cost) risk
- To leverage the existing skill set of the staff while continuing to build up weaker areas
- To acquire accurate and timely status of the overall project as well as each sub-element
- To meet or exceed expected productivity estimates
- To develop and deliver a high quality product to the customer
- To empower the staff to make timely and accurate design decisions to minimize rework
- To institute a method of achieving internal process improvement
- To enable synergy and a spirit of cooperation within the project
- To detect and resolve internal conflict quickly.

Figure 1 (page 39) summarizes this information. Each column represents one of the management goals itemized in the previous list. The rows identify key management actions presented throughout the main body of this paper. Marks in the table indicate those management actions that directly or indirectly contributed to the satisfaction of the corresponding goal. It should be noted that these marks represent the assessments of the authors and were not measured using any formal metrics.

In summary, despite the progress made over the last 25 years in advancing the state of system and software engineering practices—including improved methodologies, new languages, visual tools, online debuggers, lightning-fast PCs, and CASE tools—project success still comes down to people. Management still needs to find the best people available or be willing to invest the time and training dollars in the current staff. Once an exceptional staff is in place, it's necessary to keep the team focused on the technical milestones (eliminating the politics if possible) and to provide means of recognition from something as simple as a candy bar to a full-scale incentive bonus.

The Hubble control center PDT management team undertook all of these actions and was rewarded with a highly skilled, productive, cohesive, and communicative staff with an attrition rate that was significantly less than industry norms of the time. However, like all good things acquired, there is an upkeep cost: people need technical challenges, opportunities for additional training and professional growth, and a little TLC and recognition every now and then. But the results are well worth it—and besides, you can't be successful without them!

Editor's note: The authors welcome comments and questions. Barrett can be reached at lbarrett@hst.nasa.gov and Lehtonen kenneth.e.lehtonen@nasa.gov.