Catch a Wave and You're Sitting All over the World

Tom Kendrick, PMP, MSEE, MBA
Sr. Project Manager, Visa Inc. (Hewlett-Packard retiree)
Wolfgang Blickle, PMP, Dipl.-Inf.
Sr. Project Manager, Hewlett-Packard Company

Abstract

Managing the worldwide deployment of a complex information technology application is a massive undertaking. Coordinating the efforts of an ever-shifting program staff of roughly 200 global contributors over more than 5 years is difficult and complicated.

The authors of this paper were responsible for planning and release management for such a program, which is now substantially completed. This paper examines what worked well for the program, and outlines adjustments made during the program as a result of lessons learned. The focus is on the program management tactics that were employed in successful execution, including: cascading plans with increasing granularity of detail, scrupulous scope management, continuous attention to program communication and information storage, and rigorous program risk management.

Overall Program Summary

Objectives: In 2002, the Hewlett-Packard Company undertook a program to consolidate the overall management of all fee-for-service projects using a single, consistent system of processes and applications called COMPASS. The COMPASS implementation program was responsible for deploying this functionality to all countries and regions of the world where HP does business. It was also charged with integrating all pre-merger HP and Compaq engagements into a coherent global business operation. The implementation of these new systems and processes was scoped as a multiyear effort. The program had management responsibility for a budget of several million dollars per year and a shifting roster of about 200 contributors.

Timing: The overall program duration was initially planned to take roughly 4 to 5 years, starting with deployment of key functionality in the top-revenue countries for this multi-billion-dollar HP business. The key to success for this long program was decomposing the worldwide deployment into quarterly waves, each lasting roughly 8 to 9 months from the setting of initial requirements through system implementation. Following a European pilot country in mid-2003, the COMPASS program implemented roughly 4 to 6 additional countries each quarter. The system is currently in operation in more than 50 countries worldwide.

Governance: Within the overall COMPASS program, each deployment wave was a separate project that was independently planned, scoped, and managed, with the waves overlapping throughout the run of the total program. The program staff included two release managers who took turns managing the successive wave releases, working with the program planner, the worldwide and regional deployment managers, regional and functional specialists, IT development and support teams, and other members of the program team. Wolfgang Blickle managed the pilot release starting in summer 2002, has been one of the release managers for the entire program, and for about a year also took over some of the responsibilities of the IT program director. Tom Kendrick was the program planner from summer 2003 to spring 2006. Wave 13, at the end of 2006, will substantially complete the program. A steering committee composed of executive-level sponsors from the relevant HP businesses and regions met monthly to review progress, set priorities and deployment sequencing, and deal with significant program-level issues.

Managing the Program

The program staff: A small team of about a half-dozen people reported directly to the IT Director responsible for the overall program (see the striped area in **Figure 1**). As with many modern projects and programs, these program staff members were not in direct control of the large number of contributors ultimately responsible for the work.

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

Fortunately, we established a strong matrix for the COMPASS program early on that turned many of the managers who were responsible for the necessary functions and individuals into de facto members of the program staff.

These additional members of the team appeared on all program organization charts, were tightly involved in all planning and tracking, and attended all weekly and other program staff meetings. No distinction was made on the program team based on management reporting relationships, and a great deal of effort was put into ensuring that the primary loyalty for each team member was to the program, not elsewhere (Kendrick, 2006, pp. 29-56).

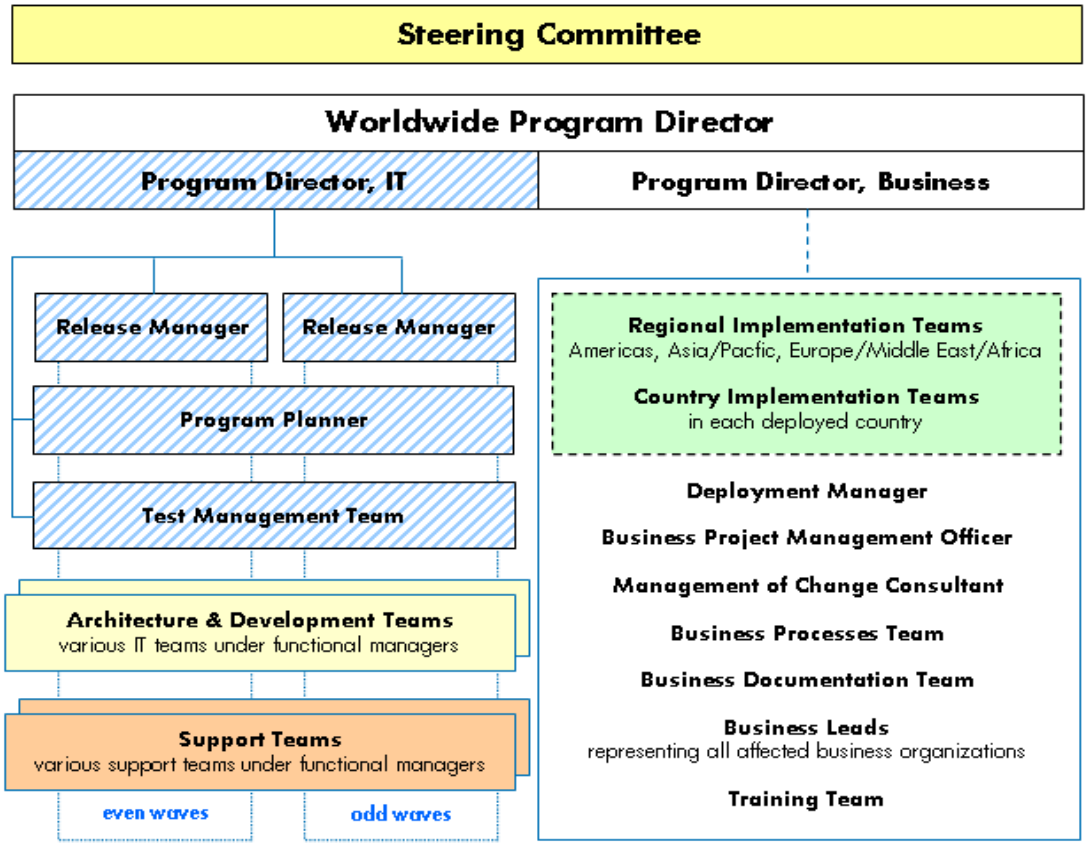

Figure 1. A high-level organizational chart of the active program participants

Planning summits: The COMPASS program team was distributed worldwide, and at various times had people located in just about every time zone. We held face-to-face planning summits about every six months throughout the program to foster trust and loyalty. They were rigorously planned, multiday meetings during which we accomplished a great deal of planning and decision making. All program staff members and a good number of others who were responsible for significant program deliverables participated, and over time scores of people attended one or more of these sessions.

In addition to the program-related work accomplished through pre-study, working sessions during the meeting, and follow up, the program benefited immeasurably from the team-building that resulted from putting names to faces. Joint meals, snacks provided by the participants, and events (such as the legendary bocce outing) ensured that the program seldom encountered problems common with “virtual teams.”

Weekly program staff meetings: Weekly 60-to-90-minute conference calls provided overall program continuity. Starting times varied during the program to accommodate distant participants with as little annoyance as possible. We distributed detailed meeting agendas in advance, and we sent meeting notes to all program staff promptly. We also posted staff notes to our program knowledge management system (see below), where all program contributors could review what was discussed. Standard agenda items included the status of the roughly three deployment waves that were active at any given time, with additions to resolve significant program issues. Meetings were kept on point,

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

and whenever possible they ended early. For the most part, everyone involved attended regularly and with little complaint.

Planning and Scheduling

Program planning was done at several levels: at the overall program level, the deployment wave level, and the detailed function level. These plans had increasing detail corresponding to the depth at which the plan was used, but all plans were synchronized throughout the COMPASS program.

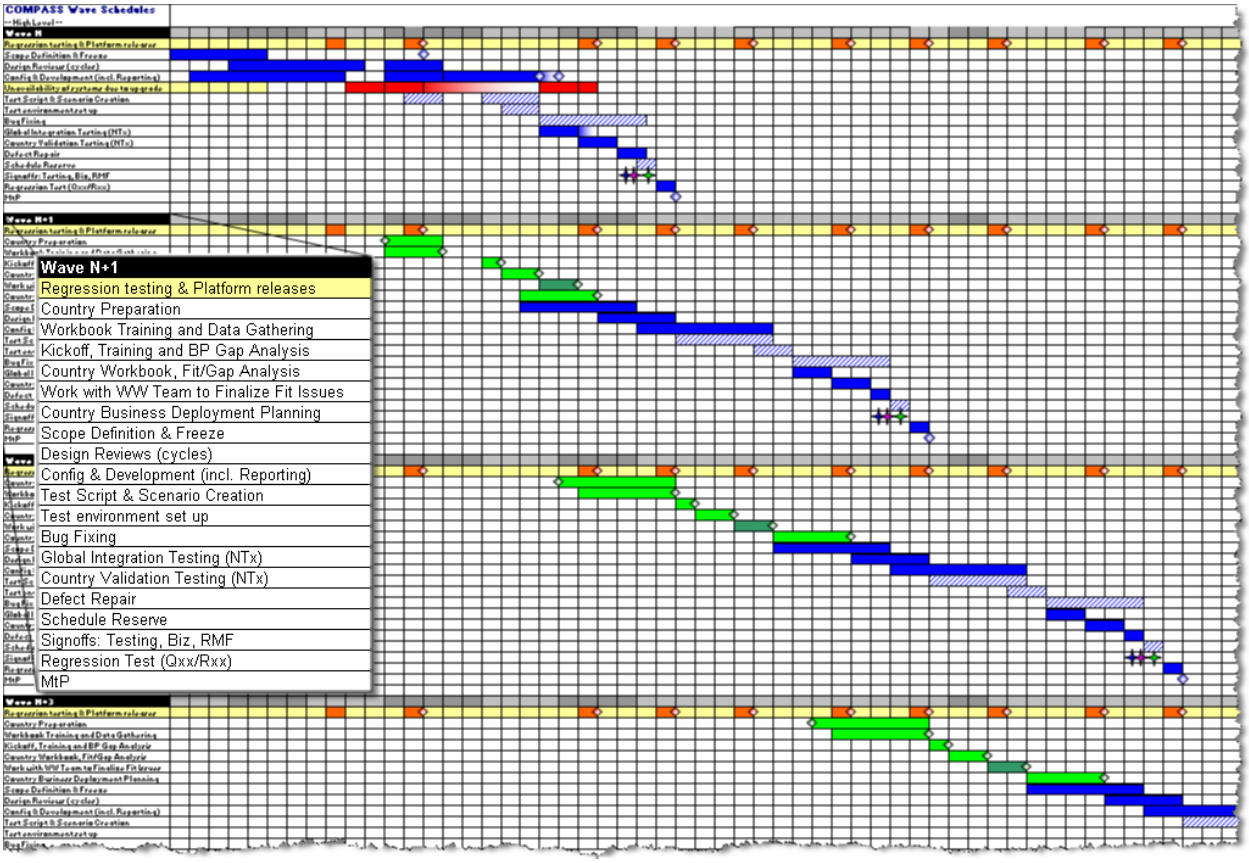

Figure 2. A simplified program-level multiwave plan

Program-level multiwave plans were created with Primavera TeamPlay, and they included sufficient detail to identify contention and other major timing issues that would arise from the phased execution of the overlapping deployment waves. For planning summits and for other discussions requiring high-level program information, we created extracts that we could distribute in Excel or PowerPoint, such as the somewhat simplified example in **Figure 2**. We also decorated large expanses of wall space in the meeting room with poster-sized versions of the charts for reference. Throughout the program, we created evolving versions of this chart, dropping completed waves and adding new ones in a moving time window extending about a year into the future. These high-level views were also stored online, where program contributors could easily view them.

Cross-functional wave deployment plans for the COMPASS pilot consisted of a roughly 1,000 line Gantt chart that was very heavy on IT development activities. This schedule was generally incomprehensible and not very useful, even to the planner who created it. To better address key activities from all the functions involved in the program and to make the overall wave plan easier to read, we developed a format for wave planning that grouped key decision points and milestones shared by all parts of the program team at the top of the plan, under the heading "Wave Key Milestones."

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

We identified milestones 2 to 3 weeks apart, and arranged them in sequence. Subsequent sections of the plan clustered summary-level activities for each part of the program team. Dependencies within each section were linked as usual, but cross-functional dependencies were all linked through the common set of major milestones at the top of the plan. Overall program timing was easy to see by expanding just the milestone section. Each part of the program team could expand their segment to see additional details that mattered most to them. The plan in **Figure 3** is a simplified example of the typical wave plan that we used, with the testing team's summary activities expanded. Although the overall wave plans still contained hundreds of activities, when only one section plus the program milestones were expanded, only a comprehensible few dozen lines were in view. Wave plans were stored online with all the sections collapsed, except for the milestones cascading across the wave's timeline at the top.

ID	WBS	Task Name	Duration	Start	Finish	% Complete
1	**1**	**Wave N Key Milestones**	**199 d**	**Jul 4**	**Apr 6**	**0%**
2	1.1	Wave N Participants finalized	0 d	Jul 4	Jul 4	100%
3	1.2	Wave N Participant Configurations Documented	0 d	Sep 5	Sep 5	100%
4	1.3	Wave N Requirements Complete	0 d	Oct 10	Oct 10	100%
5	1.4	Wave N Scope frozen	0 d	Oct 31	Oct 31	100%
6	1.5	Wave N Scope changes prohibited	0 d	Nov 21	Nov 21	100%
7	1.6	Wave N Design Complete	0 d	Dec 22	Dec 22	100%
8	1.7	Wave N Construction Complete	0 d	Jan 16	Jan 16	100%
9	1.8	Wave N System Tests Complete	0 d	Feb 6	Feb 6	0%
10	1.9	Wave N Participant Tests Complete	0 d	Feb 20	Feb 20	0%
11	1.10	Wave N sign off and release	0 d	Mar 9	Mar 9	0%
12	1.11	End	0 d	Apr 6	Apr 6	0%
13	**2**	**Program Staff activities for Wave N**	**191 d**	**Jul 4**	**Mar 27**	**76%**
34	**3**	**Business Process activities for Wave N**	**194 d**	**Jul 4**	**Mar 30**	**71%**
76	**4**	**Participant activities for Wave N**	**199 d**	**Jul 4**	**Apr 6**	**67%**
121	**5**	**System Development activities for Wave N**	**179 d**	**Jul 4**	**Mar 9**	**79%**
153	**6**	**Output developmnt for Wave N**	**180 d**	**Jul 4**	**Mar 10**	**77%**
170	**7**	**Finance activities for Wave N**	**180 d**	**Jul 4**	**Mar 10**	**80%**
179	**8**	**Testing activities for Wave N**	**95 d**	**Nov 1**	**Mar 13**	**62%**
180	8.1	Develop Wave N test plans	20 d	Nov 1	Nov 28	100%
181	8.2	Communicate participant testing requirements	20 d	Nov 1	Nov 28	100%
182	8.3	Develop Wave N test scenarios	15 d	Nov 29	Dec 19	100%
183	8.4	All test data loaded	16 d	Dec 23	Jan 13	100%
184	8.5	Test Plans for Wave N validated	16 d	Dec 23	Jan 13	100%
185	8.6	Conduct system tests	12 d	Jan 17	Feb 1	0%
186	8.7	Conduct Participant tests	13 d	Feb 2	Feb 20	0%
187	8.8	Retest, following defect correction	28 d	Feb 2	Mar 13	0%
188	**9**	**Training activities for Wave N**	**110 d**	**Nov 22**	**Apr 24**	**33%**
199	**10**	**Support activities for Wave N**	**140 d**	**Nov 1**	**May 15**	**39%**
213	**11**	**Release activities for Wave N**	**17 d**	**Feb 7**	**Mar 1**	**0%**

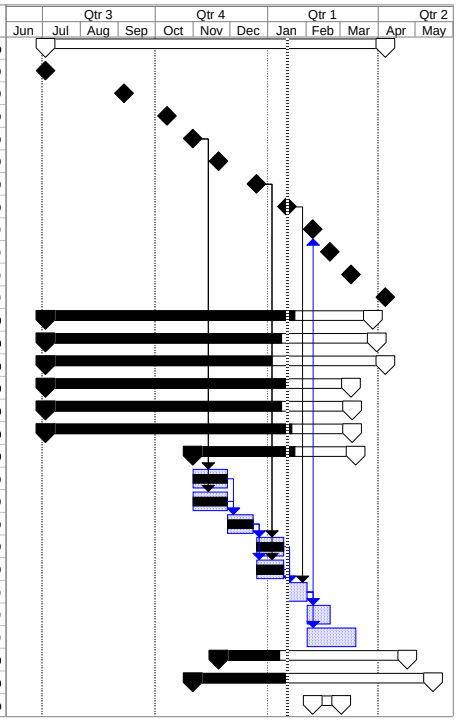

Figure 3. A simplified cross-functional wave deployment plan

Structuring the plan this way also simplified tracking, because the milestones gathered at the top would shift whenever anything in any section of the plan slipped enough to affect one of them. This would cause ripple effects throughout the plan and help us identify program-level issues that we had to manage.

Detailed functional plans: Consistent with good project planning practices, each functional team (development, testing, training, and so forth) was responsible for developing its own schedules, and many of these contained several hundred detailed activities. We aligned the higher-level plans with the task summaries from the functional plans and used them to identify overall program timing issues. We baselined each wave using information from each functional plan and we tracked overall weekly program status using summary-level status from the functional plans.

Scope Management

Scope control can be a challenge on any project, but on this multiyear program with frequent releases and many players with different interests, it was initially a nightmare. The keys to bringing the program under control were gaining buy-in for a well-documented scope management process, proactively and visibly assembling requirements for prioritization, setting and holding minimum date offsets for changes relative to release timing, and dogmatically removing items from scope that failed to meet key checkpoint requirements (Kendrick, 2004, pp. 109-110, 183-185, 189-90).

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

Process definition and program change management: We adopted a detailed process for scope management early in the program, using a thorough process description and explicit flowcharts to illustrate the steps and decision points. Clearly separating the overall process into two separate parts was a key to success:

- **Change control:** The submission and analysis of change requests. Virtually any program contributor could submit scope change requests at any time during the program. We listed requests and initiated analysis on them at least biweekly.
- **Scope control:** The decision to set (or change) program scope. All scoping decisions were the exclusive responsibility of a small number of key program staff members, and most program scope decisions were made only quarterly, roughly five months prior to each wave's release when plans were baselined. Subsequent to this, scope changes became a very big deal. Potential changes, particularly any additions, were major agenda items scheduled in advance and discussed during the weekly program staff meeting. The threshold for justification of a change was set high, and as a result changes were infrequent. Often, scope additions resulted in other items that we dropped.

Prior to this process change, program scoping was too fluid and difficult to manage. Development teams had to contend with unrealistic additions to scope early in the wave that created more work, only to see those additions dropped prior to release. While the process initially appeared to constrict the scope of each wave, the stability it encouraged actually resulted in significantly better throughput and delivery of functionality in the later program waves.

Collecting configuration data and other requirements: Another aspect of the program that supported our success was our early realization that we needed to begin collecting configuration and specific legal and other requirements for each country to be deployed quite early. Initially, we assumed that roughly six months prior to a release would be sufficient to finalize the list of countries in a deployment wave and to collect information for each release. For a program of this complexity, we quickly realized this was inadequate. Initial contact and establishment of a team to support deployment in each target country had to be nine months in advance to allow for training, assembly of key configuration data and specialized requirements, and gaining commitments from all the local stakeholders and sponsors who would be involved. We set up a series of teleconference meetings (at "convenient" times for the program management team such as 5 A.M. or 10 P.M.) about seven months prior to release to collect the information necessary for routine configuration and to document any proposed changes to the system that would be needed to meet regulatory or other local requirements. This timing also allowed us to uncover, document, and analyze any country-specific scoping requirements in sufficient time to incorporate them into the scope freeze for each deployment wave.

Scope freeze and scope lock: Freezing scope at about five months before each release improved throughput and predictability, but even this was not quite enough. Through our early deployment waves, late additions to scope persisted that could not be completed and ultimately had to be dropped. Solving this required an additional scope date: scope lock. About four months prior to each release, wave scope was "locked," which involved two things. First, all requirements that required a design review date and did not yet have one scheduled were removed from scope. Second, following the scope lock date, no further additions to scope could be made—the only changes after that date would be drops. This further improved our ability to deliver more functionality in each deployment release, meet commitments, and improve user satisfaction (because fewer items were ultimately dropped).

Design review and testing checkpoints: Also critical to scope management were the requirements associated with key checkpoints. No change request could be part of the official plan of record for a wave until a design review had been scheduled for it, and change requests that missed their design reviews or failed to get through them were promptly removed from scope and added back to the program queue for consideration in the next deployment wave. Similarly, change requests that lacked information needed for the testing that began about six weeks prior to release were also dropped. By putting teeth into the checkpoints and making it clear that we would drop all items with unmet requirements from the plan of record, program scoping problems dropped significantly.

Program Communications

Communication is important on any project; on large, complex programs it is essential to communicate well, or face chaos and disaster. Throughout the COMPASS program, we put a good deal of effort into program communications.

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

"All hands" conference calls: One good example of our communications program was our monthly "virtual" program team meetings. With roughly 200 program contributors from various functions and regions, it was impossible to hold any sort of regular, in-person meetings with everybody. What came close and helped keep everybody synchronized were our monthly all-hands meetings: The program leadership team (typically 2-3 people) presented the status of current waves and plans for the near- to mid-term future to everybody in two hour-long phone conferences (scheduled about 10 hours apart on the same day to accommodate global participants). Each month about 100 contributors participated in these calls. All of the presentation materials to be discussed on these calls were distributed in advance and stored on the program web site, as well as maintained in an archive for future reference and for new program team members. The format of each call's beginning was essentially a broadcast, but ample time was reserved at the end for questions from the participants.

Broadcast emails and training: Weekly meetings (such as the program staff meeting and other cross-functional meetings) always produced meeting minutes that were e-mailed to a subscription-based audience and posted in our knowledge management system for others. In addition, every release manager published and e-mailed a weekly status report of his or her wave that was the definitive source for up-to-date information concerning each wave.

Program training had a number of objectives:

- Early high-level training was essential for future end users to ensure that they understood the solution well enough to identify gaps and issues that had to be addressed in their waves. In each region, this was initially done by the worldwide team, but was quickly delegated to experienced business users on the regional teams.
- A dedicated training team developed much more detailed and practical end-user training for each wave and delivered it near to each release date with assistance from existing trainers worldwide. For each wave, they also summarized all new program functionality that would affect the installed base in their quarterly training bulletins.
- The program development team was responsible for even more technical training for the support team members, and conducted special quarterly knowledge transfer sessions.
- Finally, each functional team in the program conducted overall introductory training and coached their new team members on their specific part of the program.

Program Web site: A simple Web site was maintained for the program, listing high-level information of interest for many people, including those outside the program, with pointers to more detailed and specialized information stored online elsewhere.

Knowledge and Information Management

We stored most detailed program information online in a knowledge management system that provided all program team members with around-the-clock access. Nearly all program team members had write access to all areas of the workspace, and they were able to read and update program information as needed. All older versions of current program files were automatically retained (as earlier versions), providing historic project records that could be referenced whenever needed (Project Management Institute, 2004, p. 230). The only overall restriction on our knowledge management system was that we disabled "delete" access for most users to prevent inadvertent data loss.

Program plan of record: The plan of record (POR) was a high-level document that was maintained in a partitioned workspace where all could read it, but only a small number of people on the program staff could update it. It tracked a summary of the scope for all past and planned future waves, and it was updated with each program scope change. The POR provided a good summary of the program's overall achievements and plans.

Overall program information: We maintained general program-level information in a number of folders set up for program plans, information specific to each wave, program staff information, and other matters of interest to all program contributors. Access was wide open to every single contributor of the program, and these folders were the main information archive for the COMPASS program.

Program function detail: In addition to the general program folders, we also set up sections in the knowledge management system for each functional area and region, where this team-specific information could also be

centrally maintained and was available to everybody. While each team maintained their functional plans and status information independently, because it was all in one place and consistent we could assemble the weekly program progress reports directly from their data. This minimized aggravation and our annoying requests for program status. Functions such as testing established elaborate sub-structures that extended this idea and allowed nearly automated reporting based on detailed information posted by participants during each wave's testing.

Change requests and status: Change requests came from various sources, but each one had to be listed, stored (after verifying its contents), and organized. We used an online list within the knowledge management system to track all submissions, many hundreds over the course of the program. The list contained links that pointed at the detailed information for each proposed change, individual's names who were responsible for various aspects of the changes, status information (including the wave containing each change that was in scope), and other data. The list could be sorted, filtered, exported, and printed by any contributor. We used the list to assist our difficult prioritizing, analysis, and scope freeze decisions throughout the program.

Process documentation: We also centrally maintained all key program process descriptions in the knowledge management system. Whenever we determined the need to change a process, we thoroughly discussed it with all the affected functions and people, documented the modified process, communicated the change, and promptly updated the online process descriptions.

Program lessons learned and retrospective analysis: At the end of every wave (sometimes even within a wave at the end of a particularly bumpy phase), the release manager conducted a retrospective analysis by soliciting written feedback from all program contributors. We used a form with a number of high-level categories for documenting individual feedback and then held teleconference meetings to refine, augment, and prioritize this data. We "mind-mapped" the inputs into categories to determine what was most at issue (Bolwerk, 2006). We used these mind maps for presentation and discussion in our program staff meetings. After discussion, we identified the top three areas of improvement and took action on them. We also archived all the retrospective survey information in our knowledge management system for reference and to help us in identifying longer-term trends.

Risk Management

Risk management was another important success factor for the COMPASS program. By periodically considering risks and keeping them visible, we avoided quite a few problems and kept the program on track.

Program risk reviews: Starting from the very beginning of the program, we met monthly to consider exposures and potential problems. The program management team reviewed existing risks and added new ones based on analysis of current plans and changes external to the program. For each identified risk, we assessed both the likelihood and the severity of significant risks, implemented strategies for mitigation, and developed contingency plans for risk recovery.

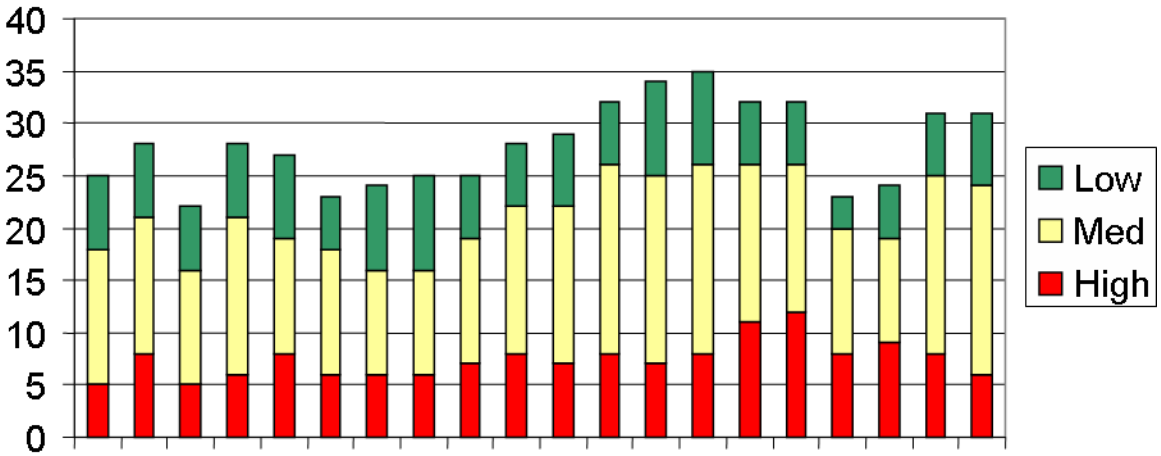

Figure 4. Program risks tracked monthly, by severity

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington

Program risk register and metrics: We maintained a program risk register in a simple table that was updated after each monthly risk meeting. The list was sorted based on overall risk severity, and stored where program contributors could review it. Changes to the risk register were tracked each month, and we communicated directly with all the people who were involved with new or imminent risks. We also tracked key program metrics (Kendrick, 2003, pp. 251-258). For example, **Figure 4** summarizes the content of the program risk register through the initial 6 waves showing the overall risk profile for the program. As the program progressed, we used our risk analysis and metrics to better manage potential future risks based on their similarity to past problems.

Conclusion

The COMPASS program has successfully completed its overall goals despite significant challenges, including reductions in staffing and funding, accepting substantial increases in functionality from its initial scope, and the turnover in staffing that is inevitable on lengthy programs. While our success relied substantially on dogmatic application of good project management practices, this alone would not have been nearly enough.

In large part, we succeeded by developing strong processes for managing scope and timing at all levels of the program, processes that were created with broad participation from the program's contributors and improved throughout the program using the lessons learned from each deployment wave. In the most recent waves, we successfully delivered substantially more functionality using smaller staffs; without the improved program processes that evolved, progress would have become sporadic and execution chaotic. Hierarchical decomposition and synchronization of plans made it possible (though never easy) to control the hundreds of interdependencies and myriad activities necessary to keep the overlapping wave schedules on track. Program management processes that were fine-tuned to our needs were essential to our accomplishments.

Looking back, however, the biggest success factor was the investment we made, early and often, in building strong relationships and trust throughout the program. While this was generally true across the whole program, it was especially important among the members of the program staff. We all placed the needs of the program well above the specific details of our assigned roles. There were never issues of coverage when people had to be absent from the program. Each individual's broad involvement in all aspects of the program was enormously important in uncovering risks, because all of us were aware of the whole program, and each of us could spot potential problems from our perspectives that may not have been visible to others. Blurring of roles was also common during times of stress (which were frequent), such as collecting configuration information simultaneously from a half-dozen countries to start a new wave, freezing scope for a wave while managing an avalanche of new change requests, and dealing with the late stages of release testing; it mattered little what our assigned roles were, we all pitched in and got things done. Trusting people, providing frequent, truthful, and timely status, and investing heavily in communication created the atmosphere of "one for all and all for one" that built the foundation for a successful program.

References

Bolwerk, T.G. (2006). *Mindjet MindManager: A Vital Solution for Improved Project Management*. Retrieved on July 30, 2006, from http://www.mindjet.com/pdf/us/Mindjet_Project_Management_Whitepaper.pdf.

Kendrick, T. (2003). *Identifying and Managing Project Risk*. New York, NY: AMACOM.

Kendrick, T. (2004). *The Project Management Toolkit*. New York, NY: AMACOM.

Kendrick, T. (2006). *Results without Authority*. New York, NY: AMACOM.

Project Management Institute. (2004) *The Guide to the Project Management Body of Knowledge, Third Edition (PMBOK® Guide)*. Newtown Square, PA: Project Management Institute.

Originally published as a part of 2006 PMI Global Congress Proceedings – Seattle Washington