

Collaboration: Development & Management Collaborating on Project Success

Collaborative management techniques bridge the needs of developers, project managers, and infrastructure operations staff to ensure the applications needed to support e-business can be built quickly and managed effectively. Here is a report on trends in collaborative management to support e-business from The Standish Group.

BY JIM JOHNSON, KAREN D. BOUCHER, KYLE CONNORS,
AND JAMES ROBINSON

[Collaborating on Project Success - Part II](#)

[Evolve: Collaborating at Peak Efficiency](#)

[Project Managers Find Tips, Tools, and Community at gantthead.com](#)

[With Help from Visible's Tool, OPT Converts Manufacturing App](#)

[Artemis Offers Multiproject Planning, Control, and Tracking Software Solutions](#)

[Collaboration Center for Software Development: speedDEV](#)

[Table: Project Management Tool Suppliers](#)

Project Management: The Criteria for Success

There's great news on the project management front, according to The Standish Group, West Yarmouth, Mass., a research firm that focuses on mission-critical project management applications. Its most recent findings indicate researchers and project managers alike are learning how to become more successful at IT project management. According to results of Standish's 1994 CHAOS study, a research report published annually since that year, only 28,000 application development projects met the criteria for success—completed on time, on budget, and with all the features and functions originally specified. Last year's results show that 78,000 U.S. projects were successful.

The reasons for the increase in successful projects vary. First, the average cost of a project has been more than cut in half. Better tools have been created to monitor and control progress, and more highly skilled project managers are using improved management processes. The fact that there *are* processes is significant in itself.

Most of these new projects are well within The Standish Group's criteria established in "Recipe for Success, 1998," which limits project duration to six months and project staff to six people. This article is based on information from the company's latest paper, "Extreme CHAOS 2001."

The Road to success

Success rates are up across the board, while cost and schedule overruns are declining. The CHAOS research timeline is evidence of steady improvement in IT project management. In 1994, only 16% of application development projects met the criteria for success—completed on time, on budget, and with all features/functions originally specified. In 2000, 28% of projects were in the successful column. (See the figure below.)

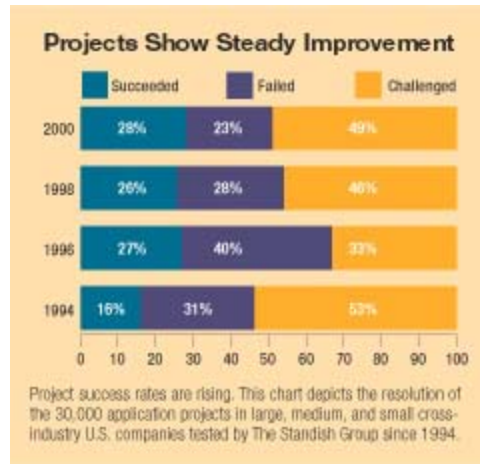
Standish categorizes projects into three resolution types:

Successful: The project is completed on time and on budget, with all features and functions originally specified.

Challenged: The project is completed and operational, but overbudget, late, and with fewer features and functions than initially specified.

Failed: The project is canceled before completion, or never implemented.

Tracking U.S. project outcomes showed that in 1994, 28,000 projects were successful, while in 2000, the number rose to 78,000—almost a threefold increase. Conversely, failed projects amounted to 54,000 in the 1994 study vs. 65,000 in the 2000 study. This was an 18% increase, while overall project growth exceeded 60%. Challenged projects grew at a rate of 62%, to equal 137,000 over the 1994 number of 93,000.



Cost overruns in 1994 equaled 189% over the original estimate. This was reduced from 69% in the 1998 study and down to 45% in the 2000 study. Time overruns dropped from 222% in 1994 to 63% in 2000. Another piece of good news is that in 1994 only 61% of the required features were delivered on challenged projects, compared with 67% in the 2000 study.

Overall, the outlook is good. Project success rates are up, and overruns are down. More importantly, although the number of projects is expected to double in 2001, the rate of failure is expected to decline substantially.

What ensures a project's success? The original CHAOS study, conducted in 1994, identified 10 success factors. Standish has updated the CHAOS Ten for 2000. Although no project requires all 10 factors to be successful, the more factors present in the project strategy, the higher the confidence level. (See Table 1, "Recipe for Success: CHAOS Ten," below.)

Recipe for Success: CHAOS Ten

Confidence Level	Success Factors
Executive support	18
User involvement	16
Experienced project manager	14
Clear business objectives	12
Minimized scope	10
Standard software infrastructure	8
Firm basic requirements	6
Formal methodology	6
Reliable estimates	5
Other criteria	5

Table 1. Each factor has been weighted according to its influence on project success. The more points, the lower the project risk.

1 Executive support. Traditionally, executive support occupied the No. 2 spot; however, it is now the No. 1 factor in project failure. Executive support influences a project's process and progress. Lack of executive input can jeopardize a project.

2 User involvement. Lack of user involvement traditionally has been the No. 1 reason for project failure. Conversely, it has been the leading contributor to project success. Even when delivered on time and on budget, a project can fail if it doesn't meet user needs or expectations. However, this year user involvement has moved to the No. 2 position. Despite how that may sound, user involvement hasn't decreased in importance; it's just that IT professionals have, in effect, solved this major problem.

3 Experienced project manager. Ninety-seven percent of successful projects have an experienced project manager at the helm.

4 Clear business objectives. This factor has moved down one spot because evidence shows that experienced project managers increase success rates.

5 Minimized scope. Wrapping up the top five is minimized scope. Time is the enemy of all projects, and since scope affects time, or project duration, they are linked. Clearly then, minimizing scope increases a project's chances of success. Minimized scope has replaced small milestones. While these two factors are similar, the act of minimizing scope leads to greater success than does creating small milestones. Concentrating on the top five will result in 70 success points.

6 Standard software infrastructure. Requirements are in a state of constant flux, but infrastructure needs stability. The Standish Group's research shows that 70% of application code is infrastructure. Some of this code is unique to the application; nonetheless, much of this code could be purchased from an infrastructure vendor.

By using standard infrastructure, the application development team can concentrate on business rules rather than on technology. Many application development projects fail not in stand-alone application development, but in existing application integration. Standard infrastructures can shortcut application integration.

7 Firm basic requirements. The word "basic" refers to base-level requirements. Creating minimal, obtainable base requirements and then developing those features will reduce the effect of change. Delivering minimal features allows users and executive sponsors to see quick results. As a result, project managers are better prepared to articulate the needs and priorities of the next project phase.

8 Formal methodology. This provides a realistic picture of the project and resources committed to it. And it results in steps and procedures the team can reproduce and reuse. It also enables the team to maximize consistency. And it incorporates lessons learned into active projects. The process encourages a go or no-go decision checkpoint. It also helps the project team proceed with a higher level of confidence, or halt or alter steps to fit changing requirements. CHAOS research shows that 46% of successful projects use a formal project management methodology, compared with 30% of challenged and failed projects. So, this factor should increase success rates by about 16%.

9 Reliable estimates. Systematic project estimating must be approached realistically, because estimating is just plain hard. Then add to that the difficulty of developing, purchasing, and integrating components into existing and packaged applications, and outside services. IT managers must use all their collective knowledge and experience to come up with estimates that reflect the true effort required.

10 Other criteria. In last place is a collection of other factors. These factors include small milestones, proper planning, competent staff, and ownership. In the past, each of these factors was given its own category.

The CHAOS Ten success factors continue to be valuable for assessing project potential. While nothing can guarantee project success, adhering to the CHAOS Ten will increase your odds of putting together a winning project.

The Standish Group predicts that most projects started and resolved this year will be microprojects: ones lasting no more than four months and run by four people. CHAOS research shows minimization as a key factor in successful projects. The microproject is the ultimate in minimization. Many last only three to four weeks. Don't confuse microprojects with milestones—microprojects live and die on usable deliverables.

The Web and standard infrastructure have made the microproject a viable entity. New application versions can be brought up quickly, and bugs can be found, corrected online, and benefits realized immediately. There's no need for release dates, shipping codes, or drawn-out user training. One of Standish's Fortune 500 clients mandated microprojects. The company's president must approve any project estimated to cost more than \$100,000.

However, the advent of the microproject has made it harder to manage resources, and has turned code version management into a nightmare. While microprojects are deliverables by themselves, they rarely stand alone. Therefore, changes in one project can affect the collection of objects, individual programs, interfaces, and databases. These entities could be in the thousands, while teams and developers span time zones. To address these issues requires an asset-based change management tool.

An asset-based tool that supports configuration management over multiple technologies and programming styles is just the ante. The tool must support versions and relationships of source files, directories, test cases, and configurations. The tool must also be able to communicate the project to the development community to prevent duplication and make developers aware of the project's use and reuse. It should control concurrent and parallel versions and use roles-based access control. This type of tool can go a long way in maintaining and developing successful applications.

Continue to [Part II](#)

Jim Johnson (jim@standishgroup.com) is chairman of The Standish Group, West Yarmouth, Mass. Karen D. Boucher (karen@standishgroup.com) is executive vice president. Kyle Connors (kyle@standishgroup.com) is a research adviser at Standish. James Robinson (james@standishgroup.com) is also a research adviser. Call The Standish Group at 508-760-3600 or visit <http://web.archive.org/web/20010304015530/http://www.standishgroup.com/>.

For more information on this topic in the future, register [Here](#).

[\[Editorial Focus\]](#) [\[The 500\]](#) [\[Newsletter\]](#) [\[Subscribe\]](#)

[\[Current Issue\]](#) [\[Next Issue\]](#) [\[Past Issues\]](#) [\[Advertising Info\]](#)

[\[More Info\]](#) [\[Contact Us\]](#) [\[Write for Us\]](#) [\[Feedback\]](#)

[\[Home\]](#)

This page was updated February 26, 2001