ITOnTime
23679 Calabasas Rd. #170
Calabasas, CA 91302
Phone : (866) 210-5205
Fax: (818) 888-5587
www.it-ontime.com

Request Management For Improved IT Service

November 2004

Summary

By far, the largest part of any organization's software budget is maintenance. Many Information Technology people including the most sophisticated managers believe that most of this expense goes to fixing bugs and technology transitions. However, academic research shows this to be wrong.

Because of this misconception, IT focuses on only a subset of tools needed to successfully do its job; bug tracking, software asset management and software configuration. This leaves the 65% that research shows is spent on software "perfective maintenance" outside of IT's core focus.

Perfective maintenance is the work done in response to users' requests for changes. These may be as small as changing the title of a report or as large as changing application logic in response to business process changes. Yet, despite the large sums spent in this area, few organizations have adopted special purpose systems.

This paper proposes that special purpose systems with distinctive features are required to meet this need. Users need an easy yet controllable way to make requests. Users need to see the progress and disposition of their requests. IT organizations need ways to control and monitor requests.

To achieve this, the software solution must have four characteristics: it must use standard desktop tools for creating requests; it must be easily accessible; it must have tracking and notification features and it must have workflow.

The Problem

Users complain loudly and frequently that Information Technology organizations are not responsive. IT generally focuses its energy on design tools; project management and reporting systems; defect tracking system and asset management systems. These support IT's perceived basic mission. However, data collected over the years by academic studies paint a different picture.

Software Maintenance Cost to Total Software Budget

Year Published	Percentage of Total Software Budget	Researcher
2000	90	Erlikh
1990	70 to 80	Huff
1979	67	Zelkowitz *et al.*

This might not be significant if the software and programming budget were not the largest part of the IT budget. IT understands this and has taken action by using defect tracking. However, this captures the smaller part of the expense. Academic studies show that 65% to 75% of software costs are perfective.[1]

For perfective maintenance or perfective software development, most organizations adopt one of three solutions. Some use structured email submissions to a central clearing house. Others use help desk solutions. Many organizations try to adapt bug-tracking systems. These fall short for many reasons. Important ones are:

> These methods are primarily black boxes from the user's viewpoint. The user puts things in but does not receive regular feedback about the disposition of requests.
> Structured email is fundamentally an open-loop process. There is no guarantee of delivery and if delivered no way to tell if someone acted upon the request without then posting it to a second system.
> Help desk systems usually have three states: we received the request; we put the request in a work queue; we completed the requests. Perfective requests are more complex and the processes for fulfilling them require workflows that are more complex then the simple help desk workflow.
> Bug tracking systems are programmers' tools. They frequently lack features required of a good system for perfective maintenance such as customized forms for different projects and different workflows for different types of requests within the same project.

Web sites are the clearest demonstration of this. After initial development, end users control the content of the Web site. Since these sites change frequently, users regularly request service. This forced many IT organizations to look at request management. There were two reactions. Some IT organizations advocated content management solutions. While appealing on the surface, they turned out not to solve the problem. Many organizations soon discovered that end users know their content creation tools well. Since they use these for the non-Web site part of their jobs (most of their day), they resist learning the specialized tools in many content managers. They also discovered that users are not designers so that content as it comes from the creator needs reworking. In the end, the expensive content management systems wound up being nothing more than a very costly Web servers and document repositories.

IT has seen "end-user computing" as the magic bullet for getting out of the request business. While appealing, two things are fatally flawed with this approach. First, end-users have other responsibilities. This limits their training and attention to the type of detail required to build quality applications and Web site content. Second, once users go off on their own, they start running around with a hammer fixing

[1] About 65% of maintenance was found to be perfective by Lientz & Swanson (1981). About 75% of maintenance costs are spent for providing enhancements (in the form of adaptive and perfective maintenance) (Martin, 1983; Nosek & Palvia, 1990; van Vliet, 2000).

every problem in sight. For many years, spreadsheets stored more mission-critical and sensitive corporate data than well-designed and well-managed databases.

End-User Requests and IT

IT is methodology and process rich and responsiveness poor. This is the view help by most end users. Users often see IT as focused on its internal needs at the expense of users' needs to operate the business. In the past two years, a backlash against increased IT spending has not only stopped spending growth but in many organizations resulted in spending reductions.

The reasons behind this are simple. User perception of IT continues at historically low levels. After the IT spending binge of the late 1990's and 2000, management is asking where the ROI is resulting from these investments. While IT makes presentations about new and expanded infrastructure to support upgraded application capabilities, users continue to complain that IT continues to have difficulty meeting their most basic needs. When senior management looks at this conflict between reported results from IT and complaints from users, they most often conclude that something is wrong with IT.

IT continues to have difficulty making users believe that they care about their day-to-day needs. Help Desks and user-centric design methodologies have not changed users' perceptions. IT wanders around asking why. Users continually tell them that it is the lack of openness and difficulty in getting simple things done that causes this problem.

Examining the key systems and process that IT uses to manage itself clearly shows why this occurs. IT sees their mission as maintaining a secure, robust and reliable infrastructure, offering applications that support and enhance the business and maintaining control over these activities. There are many operational tools that IT uses and they are not of interest in this paper. The interesting cluster is what end users see. These are the Help Desk, Asset Management, Application Requirements Definition and Project Management.

The common thread is their focus on providing most of the benefits from their use to IT. Help Desk is a good example of this. In the common Help Desk system, the technicians responsible have access to the system. Other than receiving an acknowledgement that their request is now in process, this is the last communication until the ticket is closed out. Meanwhile, the technicians have full view of the ticket's progress from assignment through diagnosis to closure. IT knows exactly what is occurring while the user is outside looking at a black box. Similarly, all the other systems mentioned are black boxes until IT shares information with users.

Because of the irregular nature of these updates, users fall back to the natural human reaction of assuming if they know of no progress, no one is working on their request. This lack of regular communication from IT reinforces their prior experience and leads them to conclude that IT does not really care about them. Meanwhile, IT is surprised when the users ask if their request or project is moving

forward. Since IT may have no idea where the request really is, they provide an answer that has a much chance of being wrong as it has of being right. This frustrates users even more and leads to a perception that IT incompetent.

Short development cycle, simple to moderately complex end-user requests suffer even more from this lack of communication. Commonly, the request system is e-mail or telephone. Both are inefficient at communications requiring feedback, as they are open-loop systems. Since users only see the result when it appears as a completed product, they generally assume that their request is at the bottom of the pile.

IT departments respond to this by doing one of several things. The most common effort is to add a request form to the e-mail system, send it to a request mailbox and then issue an auto-reply confirmation. The next most common is to hastily build something as a knee jerk reaction. Sometimes it is built around one of the existing general workflow management systems. In all cases, the result is less than optimal.

Formalized Request Management as the Solution

Instead of hiding from end users' requests, IT should embrace them as a way to improve both actual and perceived service levels. While formal structures such as ITIL do not contain an end-user request module, the need for one is quite clear.

Formalizing the request process brings benefits to both users and IT. Users have a clear way to make requests below the project level and outside items requiring immediate Help Desk resolution. Rather than being uncertain of the process or person for doing this, they can go to a centralized system. If the system has the correct features, they will do this rather than call the IT person who always helps to get things done. They may not agree with priorities or with progress on their requests. However, they will be informed and can make their case based on facts rather than rumors or hearsay. In most cases, users will abide by decisions communicated to them if the system is clear and easy to use.

IT benefits by having a standardized and systematic way to receive requests. Rather than having to track down emails, phone calls, memos and notes and then compile them, all requests will arrive in the same way. This allows for easy review, prioritization and tracking. No longer will weekly status meetings begin with a discovery of where things are but with a review of bottlenecks and schedules. This will shorten meetings and make everyone more productive.

Request quality will improve. Instead of users selecting the information they feel is important, IT will have a Request Creation form. This will enable IT to collect all relevant information easing implementation. By opening up the request process with a tool geared to end users, IT can improve service levels. End users will gladly give up their IT careers if they believe they can get service from IT.

A Proposed Solution

As a provider of Web site design and development services, IT OnTime has first-hand experience in fulfilling users' requests. Early project experience showed that for offshore outsourcing to work correctly, IT OnTime needed an easy-to-use but

comprehensive solution to collect requests. As most organizations discover, IT OnTime found that e-mail and voice mail are unreliable for collecting requests.

Here are the five items that IT OnTime sees as required for a successful Request Management solution.

1. Users must recognize all tools. They must be able to use them with little or no training.
2. The system must have integrated email notification capabilities.
3. The system must have complete workflow.
4. The implementation must not be a big project and get in the way of other priorities.
5. Once in operation, changes to forms and workflows must be quick and simple without disrupting operations.

The problems with the most common ways of collecting requests from users cause the following:

Black box systems such as the Help Desk – the original requestor has no idea where their request is in the queue. Help desk systems contain little or no workflow that involves the user. Most lack update and user inquiry functions.

Open loop systems such as Email forms – after the original confirmation, the user has no idea what is going on.

Homegrown systems – The features required to do good request management are difficult to build and maintain. Flexibility is essential and many of these applications are inflexible. The cost of developing a flexible system is very high and requires much time, knowledge and skill. Most in-house developed systems are rigid. Since processes change, this makes them difficult to maintain and so they are constantly in development and debug mode.

After surveying available solutions, IT OnTime decided to develop The Request Management System (rm^s).

rm^s combines three important elements to make the request process easy for users and valuable to IT. rm^s provides a form-based input screen for user input of requests. An administrator can customize the fields to reflect the project and users' requirements. Users can upload documents, graphics and any other files containing materials that support the request. rm^s presents this using a tool much like the Microsoft Outlook File Attachment feature. rm^s contains a workflow engine that moves requests from inception to completion. An administrator defines the workflow including branches; common forms; return loops and exceptions with a point-and-click tool. Uniquely, the rm^s workflow model is an active model. Once the workflow is specified, rm^s does not require compilation or any other steps to begin using the workflow. rm^s provides reporting so that users and system managers can observe requests flowing through the system and measure performance.

The result is greater user satisfaction coupled with increased control over the request process. Since the request flow is standardized, all valid requests are in one

system. Users can find their requests and using the request number, contact IT if fulfillment is slow. IT can view the request queue and plan resources required to manage the number of requests in the queue. Using "Time to Fulfill" reports, users and IT can see if request fulfillment is meeting everyone's needs.

Conclusion

A comprehensive request process will increase user satisfaction with IT. To achieve this, an automated request management system is essential. IT OnTime experienced this when working with its customers. In response, the company developed rm[s]. This system incorporates the features targeted specifically at end users' requests. By having a centralized request system, IT can better plan fulfillment of these requests, increase user satisfaction and controlling the cost of the largest part of the software expense budget.

About the author:

Max Rosenblatt President and CEO of IT OnTime, Inc. He is a senior business executive with extensive experience with IT organizations and within the software and computer services industry. He has significant experience in e-commerce, recently installing a distribution facility in Thailand for a Web-based vendor and developing a Website for a major US toy manufacturer. He is a pioneer of the Decision Support/Data Warehousing industry. He has consulted to Fortune 500 clients in all aspects of application development and has held executive positions with three software development companies. He has also been CIO of four companies in manufacturing and telemarketing and fulfillment. His functional area responsibilities have included Finance, Marketing and sales, customer service and Call Center operations.