Effective Zero-Cost Help Desk Software

Dan R. Herrick  
Colorado State University  
Engineering Network Services  
Fort Collins, CO 80523-1301  
970-491-3131  
Dan.Herrick@Colostate.edu

Lisa Metz  
Colorado State University  
Engineering Network Services  
Fort Collins, CO 80523-1301  
970-491-3465  
Lisa.Metz@Colostate.edu

Andrew Crane  
Colorado State University  
Engineering Network Services  
Fort Collins, CO 80523-1301  
970-491-2465  
Andrew.Crane@Colostate.edu

ABSTRACT
The College of Engineering at Colorado State University provides computing support for a growing population with a shrinking budget. Help desk communications with our clients, such as problem reports and resolutions, was inconsistent. We had difficulty tracking trouble tickets, our staff collaboration tools were insufficient, and our response times were poor due to the lack of proper software tools. In an effort to provide quality customer service and support, we had to find a solid help desk software solution at no cost.

We implemented Trellis Desk in August 2008 to replace our limited-capability open source help desk system. Since that time, we have achieved positive results. Response times to our clients have decreased, employee collaboration has grown, and client satisfaction has increased. We have better management tools including reporting capability, an overhead view of issues, and logical grouping of tickets. We will explain how an effective help desk software tool gives us the ability to better manage support for our clients by using the “right tool for the job”.

Categories and Subject Descriptors
H.3.4 [Information Storage and Retrieval]: Systems and Software – Performance evaluation (efficiency and effectiveness), Question-answering (fact retrieval) systems.
H.5.2. [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces – Graphical user interfaces (GUI), Training, help, and documentation.
H.5.3. [Information Interfaces and Presentation (e.g., HCI)]: Group and Organization Interfaces – Computer-supported cooperative work.

General Terms
Management, Documentation, Human Factors, Standardization.

Keywords
Client Service, Desktop Support, Help Desk, Software

1. INTRODUCTION
Colorado State University (CSU) has a highly decentralized IT environment. CSU has a central IT group, which provides basic infrastructure for all administrative and academic units, including internet backbone, basic email and calendar support and course management software. Each college and department must provide additional support based on the needs of its faculty, staff and students. Desktop computing support most often falls to the individual college or department.

The Engineering Network Services (ENS) group provides computing support for the College of Engineering (COE). The COE has 2,474 students and 498 faculty and staff and occupies three main campus and eight off-campus buildings. ENS employs eight permanent, full-time staff and 25-30 temporary, part-time student employees.

2. BACKGROUND
2.1 History
Historically, ENS employee roles were divided into about half client services (including labs and desktop support), and half network and systems infrastructure. Each department of the college was assigned one Support Analyst who was their primary point of contact and who primarily performed all the required work. Eventually, restructuring and budget cuts reduced the number of Support Analysts to below the number of departments, which necessitated a change in our (ENS’) service strategy. Work was divided among the pool of Support Analysts, although individual Support Analysts were still the primary point of contact for departments.

2.2 Centralization and Consolidation
In 2007, we saw that we would need a better way to manage our help requests. Our one-analyst-per-area model was becoming increasingly cumbersome and, as a result, internal and external (client) communication was less than ideal. At the same time that our college grew in faculty lines, our student enrollment increased. Support became more costly from a time and resource perspective. One of our full-time Support Analysts departed, after which we lost one FTE completely. We knew at that point that we would need a different, more cohesive way to support the College without extra cost.

We developed two approaches to changing our service strategy: We would centralize all desktop support work and create one central point of contact for our entire IT group.
2.3 Centralizing Desktop Support Work
To support the first approach, we ceased our practice of assigning one Support Analyst to one or more departments, and instead gave all desktop support work to a pool of available Support Analysts. After we lost one FTE, we were able to use some of our budget to increase our student labor force, which in turn required even more coordination of tasks and facilitation of information sharing. The current help desk, established in November 2006 to serve the needs of the student computer labs, was expanded into a full help desk in November 2007 that served all of the college’s users (students, faculty and staff). Overall responsibility for and staffing of the help desk formally transitioned from the labs team to the analyst team.

The second approach required even more information sharing among multiple sub-groups within our IT group: The analysts group, the labs group (which supports the computer labs), and the systems group (which supports network and systems infrastructure at the college level).

Our first tactic to support this strategic change was to find a Help Desk software suite that would not only help us manage workflow, but improve communication with our clients and staff. Our hope was to implement Help Desk software, which would allow us to communicate internally (staff only) through a specific trouble ticket and allow the client to ask questions of our staff without the need to call a specific staff member. We also thought that a new system would allow for more sharing of information amongst our group that would help expedite problem solving.

3. GENERAL NEED FOR A HELP DESK TOOL
In general, help desk software allows IT organizations to improve customer support, become more efficient at handling customer requests, prioritize customer requests, and generate reports. It has long been recognized that help desks, and in particular help desk software, add value to an organization.

A help desk software tool provides a standard, central communications method between staff and clients (internal or external), and among staff. For example, staff may need to collaborate to solve a problem, and they may communicate within the help “ticket” about the problem. That way, troubleshooting steps and eventual resolution are documented within the same ticket where the original problem is noted. The ticket provides a way for staff to ask clarifying questions and to communicate the status of an issue, and allows clients to provide further information about a problem. It also allows a client to query for the status of an open issue without involving staff.

Organizations further gain efficiency by being able to assign a priority to tickets and respond to tickets in a centrally defined prioritized order. Clients appreciate having a central point of contact, which takes the guesswork out of who to call.

4. OUR SPECIFIC NEEDS
Our existing help desk tool worked well for internal communications, and tracking systems or labs-related issues. However, it was not an ideal candidate for tracking desktop support issues and it was certainly not a tool that we could transition to a fully functional, client-friendly help desk system.

Losing one FTE forced us to take a good look at our existing model and find a way to more easily handle help requests.

Our next step was to determine what was most desirable in a Help Desk solution. We created a list of important components: cost, ability to assign tickets, staff-only notes, searchability, and customization.

4.1 Cost
Given our budgetary issues, cost was clearly the number one factor in helping us to determine which solutions we could look at. Our IT Director gave us a budget of “a few hundred dollars,” which was to include technical support. This seemed to limit us to free, open-source software tools.

4.2 Assignation
Because we have several different functional groups, whose area a ticket needs to be directed to, we required the ability to assign a ticket to an entire group or to an individual.

4.3 Staff-Only Notes
At times we may need to discuss the technical issues of a ticket internally to the IT group, without the client hearing the “chatter” of the work-in-progress of the support issue. We desired a mechanism to attach technical or staff notes to a ticket that the client did not see.

4.4 Searchability
As part of an increasingly complex knowledge base of support information, we needed to be able to quickly and efficiently search through solved and unsolved tickets for keywords, client or computer names, and similar information to help more quickly resolve repeat support issues.

4.5 Customization
Our group has always been inclined to use open source software because we are comfortable modifying existing software, and often see the need to enhance the performance of the software and tune it to our specific needs. The “free” price tag certainly helps.

We also have developed our own software tools to query databases of software tools like a help desk tool to extract relevant data and generate our own reports. It is often easier to manipulate data directly than to go through a software tool’s built-in reporting features.

5. THE FIRST ATTEMPT WAS A “WREQ”
Prior to this season of change, we had been using a web-based tool called wreq as our help desk software. According to the wreq web site, “Wreq is designed to be a distributed request/problem tracking system with builtin [sic] knowledge database to help systems personnel to stay on top of requests and to prompt knowledge sharing among all local support groups.”

The wreq tool is highly customizable using Perl, and has very minimal system requirements, but only runs on UNIX-based web servers. It is offered as free open source software.

Although the wreq tool can be adapted to serve the IT department’s clients, as we did, it was primarily developed as an
6. SELECTING NEW HELP DESK SOFTWARE

After exploring various options, we selected three software tools to review in detail: Best Practical RT: Request Tracker (RT)\(^5\), OTRS (Open Technology Real Services)\(^6\) and ACCORDS\(^5\) Trellis Desk (Trellis)\(^5\). We established a test server to allow our group to test features, create tickets and evaluate the pros and cons of each program. Once we felt primary testing was complete we held a meeting and addressed all possible concerns regarding each piece of software.

In this process, it was helpful to us to acknowledge that the final solution would not have to be a perfect one -- just better than the rest. We documented the limitations of each and asked ourselves: “Will this limitation prevent us from using the product effectively?”

6.1 Costs and Support

All three software tools had no cost to download and use the software. Best Practical provides custom support for RT for a monthly fee. OTRS provides value-enhanced customized versions of the software and custom support for a fee. All three had a variety of community support options. RT and OTRS are open source software. Trellis is open source, but the license agreement allows for full access to and customization of the code.

6.2 Formally Evaluated Components

All three tools fulfilled our requirements for assigning tickets. Trellis gives the ability to assign a ticket to a group separately from an individual, so that tickets may be assigned to a group and not to an individual, thus allowing greater flexibility.

All three tools allow for some type of staff-only note or comment. Trellis allows for two types: A general “Notes” section for each ticket, which is not incorporated into the timeline of the ticket; and a “staff-only reply”, which logs a reply within the timeline, which only staff can see.

All three tools had built-in search functions. RT seemed the most complex and robust, allowing a SQL-like syntax for searching tickets. Trellis provides search by predefined fields, and filters for viewing tickets.

Each of the three tools provided some ability for customization, either through the provided interfaces or administrative tools, or direct modification of the code. Trellis uses PHP for its operations and interface, and MySQL for the backend database, both of which we are familiar and comfortable with. RT and OTRS both use Perl and support MySQL, PostgreSQL and Oracle database options. (In addition, OTRS supports Microsoft SQL Server.)

In general, we found RT to be complex in setup, configuration and customization. OTRS seemed to be more straightforward in setup, partially because it provided less customization capability.

Trellis seemed to outperform the others in the flexibility of components, speed of operation, and ease of use.

6.3 Other Factors Affecting the Evaluation

Although we did not initially decide to elect usability as a specific evaluation point, we had to acknowledge its effect on the final outcome. We had to look at usability from a client standpoint, and a staff standpoint.

For clients, the usability had to be very simple: a web-based and an email interface that provided the same information. Clients can email a central email address to create a ticket, and create a ticket from the web-based tool. We also added a custom form to our web site, which allowed users to create a ticket without entering the Trellis web interface. Later, we also added functionality so that email replies to a ticket appended to that ticket rather than creating a new ticket.

For staff, we acknowledged that some staff are not as technical as others. A senior level UNIX systems administrator is not going to approach a user interface the same way as a student lab consultant. Also, the typical help desk worker is going to work within the tool much more often than a systems administrator, but the tool needs to accommodate both.

In appearance and operation, RT was too reminiscent of the “barebones”, text-oriented wreq tool for our liking. OTRS, while much more streamlined and modern, had a too-high level of complexity in everyday operation, requiring, for example, several clicks to accomplish a simple task like viewing all responses to a ticket.

7. IMPLEMENTATION

We tested Trellis for only two months (concurrent with our old help desk software) before we were satisfied enough to go ahead with a full implementation, in August 2008.

Trellis implementation was relatively straightforward as it is built on PHP and uses MySQL for the database. One of our systems administrators was able to easily install the software and get the basics running for us with no issues. We defined our departmental headings and added our staff members to the Staff Group with appropriate permissions. Trellis Desk is very customizable so we
created fields for Academic/Administrative Department, Building, Room Number, Operating System and Computer Type.

Trellis allows us to define a group (department), which a ticket can be assigned to, as well as individual assignments. We selected the following departments: Systems, Analysts, Labs, and Help Desk. Later, we added an Admin group to cover general questions like chargebacks for accounts and software licensing. All tickets are initially assigned to a generic “ENS” group, which prompts us to manually assign it to the correct group. Our help desk workers are tasked with answering common questions, and asking clarifying questions before moving the ticket to the appropriate group.

8. FEATURES OF TRELLIS

Trellis Desk allows us to group tickets logically. We can view them by internal department, user or owner (who has been assigned a ticket). When our employees arrive for their shifts, their first task is to check tickets assigned to them for follow up. They will then go through the remainder of all tickets in their department to ensure nothing gets missed. Trellis Desk makes this easy and we’re able to use the Internal “Staff Only” replies to communicate with each other on issues without the client being aware of what we’re doing. We also make use of “canned replies” to easily enter frequently used text into a ticket.

Because the Trellis database is hosted on our MySQL server, we can manipulate the data directly for fixing data issues, and export the data for reporting purposes.

Trellis provides a “System Overview” dashboard view, which shows a customizable snapshot of items like tickets awaiting action, last few actions, and system status. The Ticket Control view allows for fast sorting and filtering of tickets by priority, department, last reply and other criteria. A staff member can quickly see what tickets are assigned to them personally or to their department. Tickets can be bulk-managed (for example, sorted into a department or closed.)

Email integration is excellent. When a user sends an email to our help desk email address, a new ticket is automatically created and a confirmation response sent to the user. A direct link to the user’s ticket is included in the email so that they may refer to it for updates. Future responses to the ticket are also sent to the user’s email address. A user may click on the provided link and respond to the ticket through our Trellis web site, or reply to the email to update the ticket.

9. LIMITATIONS OF TRELLIS

Every piece of software comes with pros and cons and Trellis was no exception. We noticed that the initial version didn’t handle attachments well and there was no asset management component. These issues were not enough to deter us from using Trellis; however, and the attachment issue was addressed in a more recent revision.

Reporting capability is minimal but it wasn’t high on our list of priorities. While we can look at some statistics through the Trellis management portal, we find it most beneficial to export relevant data and use our own tracking and reporting methods.

We recently re-evaluated our help desk software tool, with the intention of finding one that incorporated asset management. We decided to stay with Trellis as our help desk software and implement a separate asset management tool, because we could not find one within our price range that performed noticeably better.

10. BENEFITS

We are able to tell that over half of our help tickets have been help desk or desktop support related so we are certainly taking advantage of this resource. The other groups, like the systems and lab groups, have benefitted from a centralized ticket tracking system, since they do not have to handle many of the “frequently asked questions”, as well as try to determine what staff member is the best equipped to handle a problem.

This software tool has certainly improved staff collaboration, as we are able to easily converse with each other within the context of the ticket about the issue at hand, without resorting to email. When the staff-only response is used, rather than two staff members emailing each other, other staff are informed of the issue and frequently (and voluntarily!) add their own helpful responses. Response times have increased due to the efficiency of the tool.

Our internal policy is to respond to every ticket within 24 hours, but in practice, we usually respond to each ticket within minutes during business hours. The web-based tool is easily accessible and some staff members choose to respond after hours (a practice we do not encourage but also do not forbid, because it indicates high involvement with work).

Most of all, our customers are happier. We do not formally track customer satisfaction, but in surveys and informal feedback, our users very much appreciate having a clear, concise route to asking for help. Our clients now have four methods of contacting our help desk: An email to our help address, a web form, a telephone number and a visit in person to our help desk. Each contact opens a help ticket in the same place, so they can check the status of their issue at any time.

Looking ahead, Trellis is preparing to come out with a new major version and will convert to a GNU General Public (open source) license. Community involvement with the software is at an all-time high, and we are excited to see what other developments will result from this new energy.

11. REFERENCES


