IT is Listening, Can You Hear IT?

**By Gordon Bleil, Lincoln Farnum,
Chad Hodge, and Amy Rubin**

**Introduction to Medical Informatics
403-DL Section 51
Summer 2009**

Problem Statement and Purpose

You use a clinical information system daily at the hospital and while it requires a few accommodations in workflow and process on your part, for the bulk of your work it functions pretty well. If only it could do a few little things better, like the Y-axis scale on the temperature graph, it doesn't need to start at 20 degrees Celsius. Also, if there were an admission order set for your specialty, it could reduce by 20 minutes daily the time you spend ordering. Is there any easy way to get more information about this patient's research protocols? And why aren't blood transfusion consents scanned as pdfs so I can see them before transfusing?

It seems like there are a thousand little things one wonders about, small changes that would create great efficiencies, and ideas, good ideas, that I don't know who to tell about. I could call the Help Desk but these aren't Help Desk issues, are they? I know I could make the system better though, because I use it. Who do I talk to?

Does the above sound at all familiar? It does to me. It sounds like an average user of our clinical information system, and we have about 4000 average users. That is 4000 users who don't know how to communicate their small needs and ideas for improvements to our system. It sounds distinctly as though we're ignoring a wonderful resource.

Clinical information systems are extremely complex assemblages of software consisting of multiple modules or many different interfaced applications addressing hundreds of user types and thousands of users. Information technologists responsible for programming these systems, in spite of knowledge of industry best practices and clinical processes, cannot know at an atomic level how various configuration options are used and embraced across the spectrum of users without significant usability testing and active monitoring of actual system use.

This proposal involves building a simple, convenient, and very fast tool to capture our users' responses to our system at the moment they experience them. Whether they are ideas, complaints, questions, or even just words of advice all that is required is to select an icon, identify the nature of the communication from a drop-down list, and type into the textbox. If they'd like to be contacted, they'll simply select that radio button and then click submit. The message files in our database, sends an email to the help desk system, and an email acknowledgement to the user thanking her for her input.

Users will take 20-30 seconds to communicate ideas to the information technology (IT) department, and if the IT department listens to those ideas and turns them into a conversation, the user will experience something unusual, satisfaction and an understanding that IT actually cares about the EHR. The IT department will experience something unusual as well, a window into the user experience at the transaction level, something that until now required shadowing individual users throughout their workday, a laborious one-on-one task.

So, how do we make this work? It sounds pretty simple but while it represents an unusual opportunity for success, the processes to support an ongoing conversation with users aren't necessarily endemic to most IT shops. In the following presentation, we'll discuss a strategy to build and operationalize this remarkable device. We'll show how it's different than current help systems and identify efficiencies not

available through other more conventional means. We'll discuss the back-end processes to support this conversation and describe a little more about what it means to the IT department and what it means to the users.

Vision Statement

Communication between system users and IT has never been easy. Different languages, different expectations, time constraints on both sides, and frustrations that lead more often to derision than partnership are situations common to all health care organizations. Our vision is one in which users can easily communicate with IT and IT can just as easily communicate back, one in which an ongoing conversation exists, continually exposing opportunities for system enhancement, specialized training, or analyzing and improving dysfunctional processes brought to light by the system.

These domains exist in an organizational gray area not served by current help systems. The kinds of messages we expect are comments and suggestions regarding system usability, logicality, and intuitiveness. We expect requests for small modifications, often too small to bother calling the help desk and waiting for an answer. We expect opinions on how things work, how they feel, and how they look, and we expect that in many cases our contributors will desire an answer, from a human, and this is where the ongoing conversation begins.

Interfaced with the current help system by email, user suggestions file and create a ticket categorized by the communication type chosen on the form. These tickets are specially routed to a triage analyst who reads them, sometimes between the lines, and assigns them to subject matter experts. Many times a comment about something being difficult in the system is actually a veiled request for education about other, easier methods of accomplishing the task, and the triage analyst will understand this. Users know and see system functionality through the eyes of actual use and virtually all users, given a voice, can impart knowledge of system limitations of which the IT department is unaware. We will cultivate that knowledge to improve the system and improve user satisfaction with the system.

Specific Objectives

To enhance usability, efficiency, and user satisfaction with our clinical information system though fostering communications between users and IT though the use of a simple input device allowing users to:

- Offer suggestions, comments, or questions at the moment they experience the wish to do so.
- Spend 20-30 seconds providing the feedback and then returning to their previous tasks.
- Direct comments to system subject matter experts and receive knowledgeable responses within a short period of time.
- Request minor changes to system functionality at the point of failure.

This will allow the Information Technology Department to:

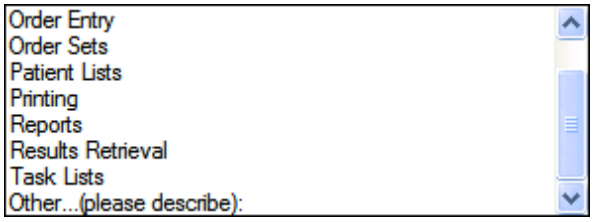
- File user input and create tickets addressed by subject matter experts.
- Respond to users with acknowledgement of their suggestions and either request more information or discuss next steps in fulfillment of them.

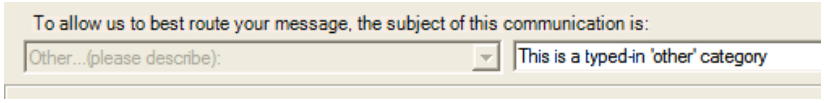
- Address unknown training deficits.
- Establish and maintain a conversational relationship with users willing to provide suggestions and input, an invaluable resource in system enhancement.
- Establish a knowledge base of suggestions usable for metrics and budgeting for user support and system enhancement.

Preferred Approach

Deployment of a user input device or form called a Suggestion Box within the system can be accomplished through numerous methods. Assuming the system supports it, it can be placed as an icon on the application's toolbar. It can also be placed on the desktop, or within an applicable menu structure. The critical success factor is availability. It must be available at all times and so easy to access that even if the user is significantly confused, he will be able to find it. Perhaps the image below will represent the suggestion box as a file box on the menu bar:

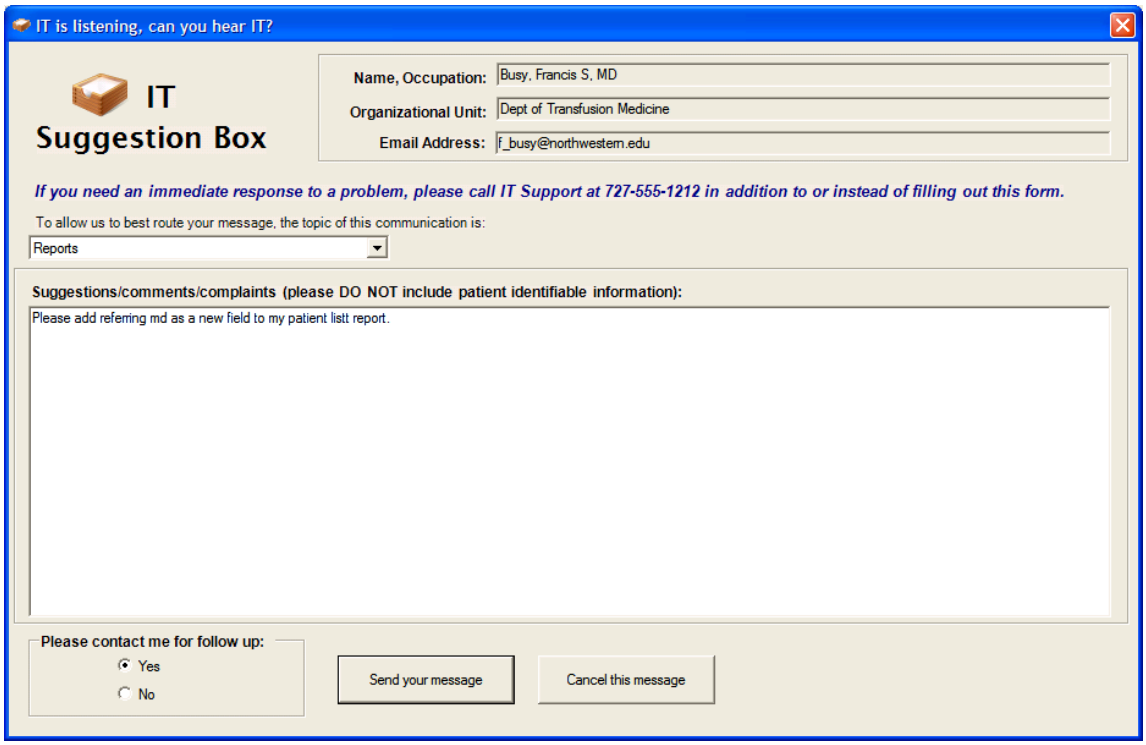
On the form, the critical success factor is ease of rapid use. According to factors published in Neilsen's Usability Heuristics, the design should remain as simple and as uncluttered with unnecessary information as is possible[1], though it may be useful to remind users to call the help desk for any real problems. The form should allow for capture of user name and credentials from the system login to alleviate the need for the user to type them in. The comment will be categorized by a method descriptive enough to allow triage to a subject matter expert and for retrospective analysis. The categories will pertain to system features and provide an 'Other' category which will expose a type-in field to the user as shown below.





The type-in field should allow for lengthy textual entries in the event that the user is so inspired, though entries are expected to be most often short and to the point. There will be a selection requesting personal contact and submit and cancel buttons. This design will serve to facilitate the action of data entry and avoid any possible user distraction. A prototype form is shown below:

[1] Nielsen, J, 1994 "Heuristics for user interface design ", viewed August 25, 2009, http://www.useit.com/papers/heuristic/heuristic_list.html



On selecting 'Send your message' the form fields will file in a custom table in the database and send emails via simple mail transfer protocol (SMTP) to the help system (e.g. Remedy), interested stakeholders in the IT department, and a note of acknowledgement to the submitting user thanking her for her comment and for helping to make the system better. If practical, the help system can send a ticket receipt to the user with the ticket number.

A simple database table created to host this data will be built with the following SQL DDL:

```
CREATE TABLE [dbo].[IT_SuggestionBox]
(
	[CreatedWhen] [datetime] NOT NULL,
	[UserName] [nvarchar](100) NOT NULL,
	[OrgUnit] [nvarchar](60) NULL,
	[Email] [nvarchar](75) NULL,
	[ContactMe] [bit] NULL,
	[Issue] [nvarchar](255) NULL,
	[Suggestions] [nvarchar](max) NULL
)
ON [PRIMARY]
```

This will be represented by the following table:

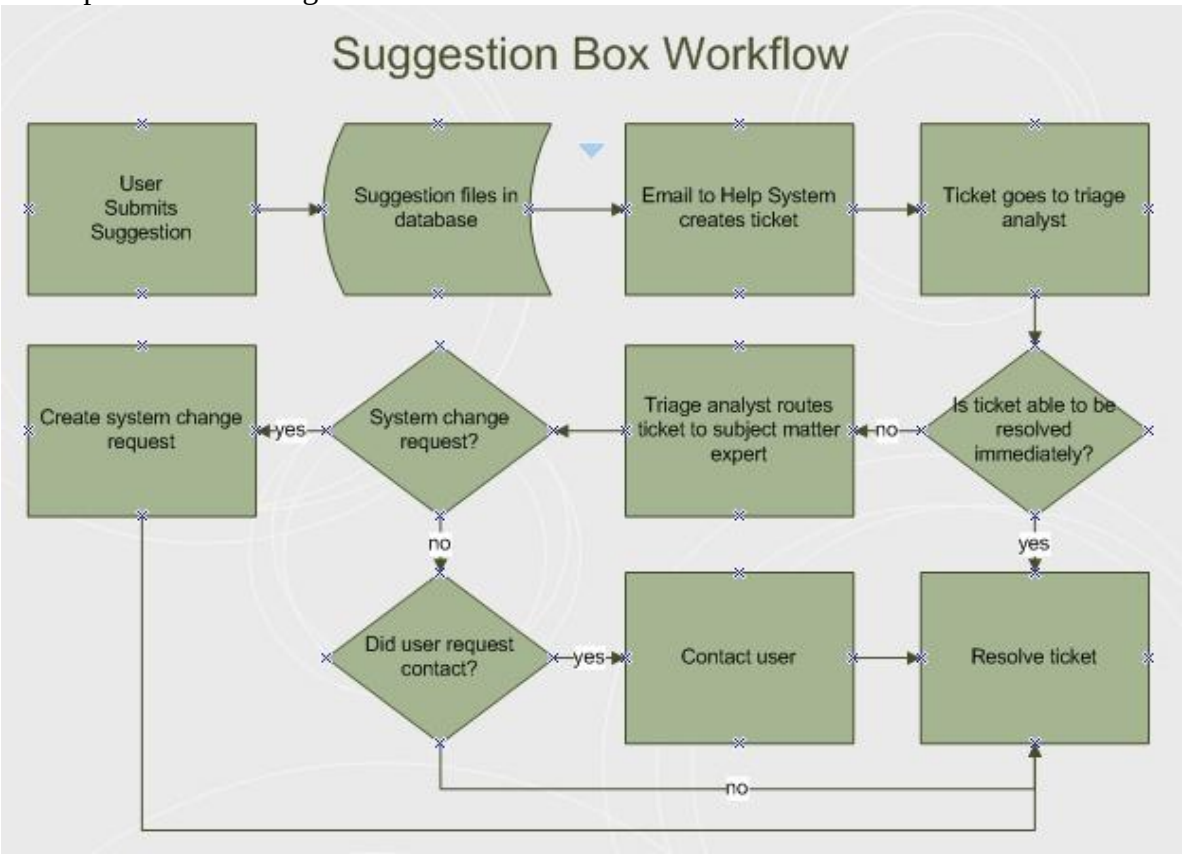
CreatedWhen	UserName	OrgUnit	Email	ContactMe	Issue	Suggestions
8/9/2009 13:24	fbusy	TransfusionMedicine	f_busy@northwestern.edu	1	Reports	Please add referring md as new field to my patient listt report

The database table will contain relevant information about suggestion submissions though, if there is an email link to the help system, ticket tracking by the help system will likely be the source of retrospective

metrics. Database tables will serve for trouble shooting in the event of errors and to support system processes in lieu of email submission functionality to the help system. In that event a back-end form could be built to support help desk personnel and to track contacts. This functionality is not currently anticipated and it therefore not illustrated.

Help desk software tickets will be directed to a triage analyst who will perform an initial interpretation of the suggestion and forward to the appropriate subject matter expert. If the suggestion is such that it can be addressed immediately, the analyst will do so and contact the submitting user to do so. If the suggestion is such that it requires more analysis, research, or other contacts, the analyst will contact the submitting user and communicate that information, providing a rough time period in which further processing will occur and when the user might expect to be informed of progress. If it is a request that is impractical or impossible, the analyst will so inform the user and close the ticket.

A sample workflow diagram is offered below:



Expected Benefits to Stakeholders

By utilizing the user input device described above, users of clinical information system can convey their thoughts and suggestions for improvement of the system. IT can use this information to assist, inform, and forward to higher levels based on the need or request of the user. This process will affect many stakeholders.

Stakeholders that would benefit from the implementation of the product within a healthcare setting include but are not limited to:

Major Stakeholders:

- Patient
- Providers/Clinical Information System Users
- Information Technology
    - Programmers
    - Testers
    - Support Staff
- Health Information Management
- Help Desk

Other affected stakeholders:

- Project Leader
- Team Member
- Patient families and personal representatives
- Ancillary providers (lab, radiology, pharmacy, etc.)
- EHR Coordinators, Trainers, Troubleshooters, Support Maintenance
- Organizations utilizing feedback system
- U.S. Healthcare System
- External Regulatory Agencies
- Payers
- Government
- Tax payers
- Staff Families
- Support Staff
    - Resource Managers
    - Line Managers

The Systems Development Life Cycle includes the integral phases of Planning, Analysis, Design, Implementation, Support, and Evaluation. The Support phase includes "maintaining and enhancing the system, and supporting the users"[2]. Historically, the Evaluation phase of an Information Technology system has been neglected without regard to the significant information it provides in reference to organizational outcomes and impact[3]. Evaluation of a Clinical Information System is instrumental in order to measure its effect on clinical outcomes and users' satisfaction. This information is beneficial for providing opportunities for continuous quality assurance and performance improvement measures, as well as justification for system changes and enhancements. Providing an electronic evaluation tool at the point of use may increase the users' participation in this process.

Additional benefits of full implementation and utilization of our user input system are far reaching. Enhancing communication between users (clinicians, support staff, etc) and information technology can allow for improvements in the usability and applicability of the clinical information system. This

[2] *Systems Analysis and Design in a Changing World,* Satzinger, Jackson, and Burd, 2009, p.48
[3] AHRQ, 2009

improvement will lead to efficiency of user input and evaluation of data by the user which leads to a cleaner database (resulting in fewer interface problems), more substantial clinical information collection and better transmission of public health data. Increasing the applicability of the system can increase the potential for clarity of understanding and effective assessment of results. The improved usability of the system leads to less compromise of the system due to commission (data being narrative or in an inappropriate filed), omission (skipping a step or leaving an empty field), abdication (failing responsibility to pass a new solution on, thereby decreasing the efficiency of others), or subversion (deliberate misuse and/or encouraging others to do so).

By facilitating better utilization and application of the clinical information systems by users, patient outcomes can be improved. Healthcare delivery and capture of critical information is improved which can lead to higher level of billing for the organization. Improved utilization during the clinical encounter due to improvements made as the results of enhancement requests and training provided will save time and thereby money and also allowing clinical providers more quality time to evaluate and assess patients.

Improved communications can allow for better allocation of resources. Training can be scheduled for the knowledge deficits identified by the users through the Suggestion Box thereby providing what is actually needed not what is thought to be needed. Clinicians will spend less time frustrated and troubleshooting the EHR and thereby can spend a greater amount of time performing patient care functions. This also leads to a more satisfied, productive user.

Requests for enhancements open a dialogue between users and IT and give IT the critical information for future developments to improve the system. This allows IT to focus their limited resources where they are needed most.

This user feedback system can be incorporated and customized not only into clinical information system but also into a wide variety of other systems as well. The underlying concept of reaching the users of a system and allowing for input at the time of need and thought will provide for improvement of usability and applicability of any system. Satisfaction of both end users and IT staff can be achieved. The users get what they need and IT knows what to give the end user.

Performance and Progress Measure

Performance improvement measures of the objectives can be measured through a Plan, Do, Study and Act methodology developed by Associates in Process Improvement.[4] By receiving input from users through the feedback system, IT can effectively plan projects, implement projects and then study the project for effectiveness and act to improve the process. The system itself can be assessed by this methodology to improve incorporating the input device to better fit the workflow. Surveys from users can be utilized to assess customer satisfaction with the product allowing IT to plan for improvements to the input system itself. Measures can be used to analyze the amount of time users wait for responses. Metrics should be developed at the onset to measure input device to see if system is providing improvement.

[4] http://www.ihi.org/IHI/Topics/Improvement/ImprovementMethods/HowToImprove/

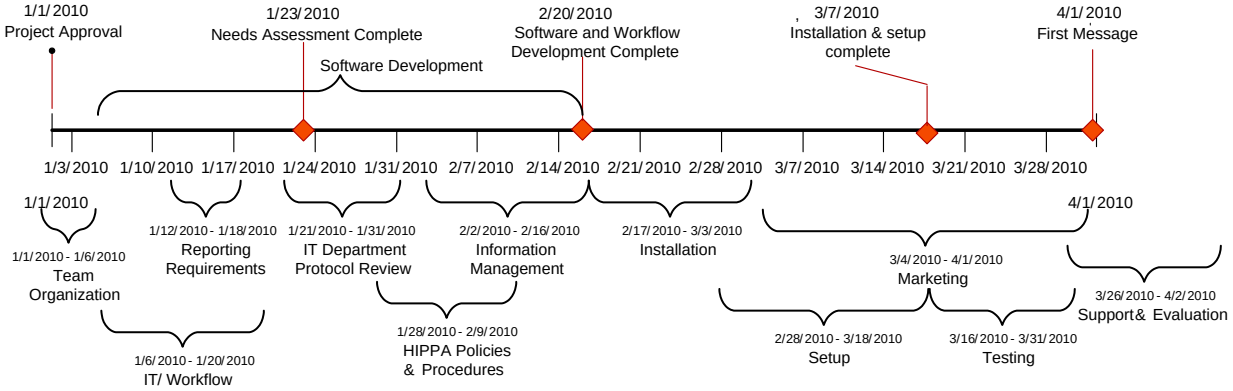
Studies into the requests presented through the Suggestion Box can be analyzed to provide areas for future developments. Areas of focus will become clear when information is easily collated and organized by task. Then by using PDSA or another performance improvement methodology, the focus areas can be recognized and the plan for improvement implemented. Analyzing the information will yield information for targeted areas for education or training, for areas that need re-development, and for areas to re-vitalize the workflow process.

Potential Risks of Implementation

Failure of user to begin utilizing the product after roll-out is a risk that will be present during implementation. Non-utilization by users will lead to a decrease in the benefits that may be realized by full utilization of the input device. Prior education and training to provide users with a look at the possible benefits obtained by utilizing a system will be a must. Emphasis on the goal of opening the communication channels between clinicians and IT should be given during training. By opening the channels, the system can and will be improved. By informing the consumers that they will receive a timely response, the chances of utilization by users is increased greatly.

Another potential risk lies in the staffing support of the help desk to incorporate the opening of the communication between user and IT. A major benefit of this system is the responsiveness from IT thus opening the communication channels. Some responses will be automated, however, more advance inquiries will need to be triaged and sent to the appropriate individuals. A delay in getting these responses to the user will cause a dissatisfaction and perceived failure of the Suggestion Box by the user. The staffing is a key issue. There must be sufficient staff to respond in an appropriate time. On the other hand, not too many staffing hours can be devoted to staffing as to undermine the project. The cost should not outweigh the benefits received.

Timeline for Implementation with Key Milestones



1) **IT / Workflow Assessment** - review IT infrastructure, internet service, software integration, workflow integration and any special integration needs. Also establish the roles and responsibilities of triage, intake, referral, evaluation, management and follow up personnel.

2) **Reporting Requirements** - understand the special needs required for reports from specialty areas within departments, software and workflow activities that are interdepartmental, and what information users find most helpful in feedback.

3) **IT Department Protocol Review** – review the roles and responsibilities related to these new tasks relative to these current tasks within the department.

6**) Information Management** – Development of assessment and reporting protocols within the existing administrative structure. Associated with this is the development of protocols to provide up to date information with appropriate levels of detail to the IT department in order to provide adequate feedback to the users.

7) **HIPPA Policy and Procedures** - updating or creating site policy and procedures regarding management of patient specific information in the event that particular records need to be utilized to evaluate the comments solicited.

8) **Installation, Set up and Testing** - IT support person to deploy necessary interfaces

9) **First Message** - Once the testing process is complete the desktop icon can be implemented with a script executed at sign-on. This may need to be flexible with limited departments or user groups being started initially in order to limit input while IT staff adjusts to the workflow.

10) **Support and Maintenance of the system** - After implementation, quality assurance and effectiveness evaluation programs are implemented, with reviews at 1 month, 3 months, 6 months and one year. This would include feedback from the users on their level of satisfaction with the system as a whole and the feedback system in particular.

11) **Marketing** - educate users about availability and how to use it. May also create custom marketing materials including a personalized emails based on whether users are registered and their activity level. Consider creating case studies after a few months of service are completed. Consider a news item on a regular but not too frequent basis to keep providers abreast of the latest changes in the overall IT system and the role that feedback plays.

Project Management and Key Leaders of the system

The development and implementation of the system would involve a relatively small team. It will be expected that this team will coordinate with appropriate IT support, software vendors and other departments to complete the necessary tasks. The most difficult aspect of the process is development of the workflow for IT, but there will be difficulties in integration of the add-on module to the existing Help Desk software. Evaluation of the system with users will be needed to assure that their workflow changes are not a barrier to usage. It will be necessary to assure that messages are accurately taken from

the desktop to Triage and capable of being accurately passed on to the Help Desk system or other locations as appropriate. Testing will be needed to assure that the screen elements work successfully and securely with the current desktop software. Finally, it will be important to assure that messages passing into and out of the desktop do not interfere with primary operations. These tasks, divided into stages as noted in the timeline above, will be completed by the following personnel.

Project Leader – Responsible to engage the stakeholders and assure completion of the project on time. It is equally important that the project not be implemented prior to having the processes in place to manage the information at all planned stages. Maintenance management of the feedback system would accomplished by the IT Department head working with the IT Triage & Referral leader and the IT Evaluation & Response Leader.

Software Developers – responsible to assure the development of a desktop icon and background software that will support the planned workflow. The software also needs to meet all the reporting, security, and management needs as they are determined.

IT Triage & Referral Leader – will be responsible to keep the user reports organized, referred, or otherwise appropriately responded to with a goal to keep system reporting timely, concise and complete.

IT Evaluation & Follow up Leader – After the input is sorted and referred, the information is evaluated by the appropriate department and the status of the follow up reported back to the user. It would be the responsibility of this person to assure that follow up is occurring and determine the barriers to response if this does not occur. Management would be notified if there is an administrative issue with follow up.

Users – for obvious reasons these are key leaders of the system. They will be helping to shape not only the character of the hardware and software computing environment, but the responsiveness and capabilities of the IT Department. It is important that they be engaged and educated early after implementation and encouraged to continue.

Quality Assurance Leader – During the development of the system, this leader will establish the measures of success and tools to be used. Subsequently periodic reports of quality and effectiveness will be provided for discussion and evaluation by management.

Alternatives considered

An important question to consider is why is this different than other help desk products that are available? If a system already exists to interact with my users, what makes this worthwhile?

The answer lies in understanding the difference in goals. A help desk is intended to help solve day to day problems and resolve bugs in the system. This project is designed to help IT actively listen to the user so as to better understand the user and what they are looking for. This is, in essence, an extension of the initial needs assessment into day to day activity. The goal is to help maintain the investment that the user and all the other stakeholders have in the information system.

Communication about change and its associated frustrations is key to success. As an example, IT practitioners using the ITIL (Information Technology Infrastructure Libraries) framework will painfully attest that even having world-class developers using world-class change and configuration management you can still have horrendous service levels because of poor coordination with IT operations.[5] The same is true if you don't have an adequate link between users and IT operations – it doesn't matter how well you in IT are if you don't coordinate with your users.

Although there are many Help Desk software packages available, they don't have modules that are designed to provide a broad range of timely feedback to the user, nor do they focus on managing long term development.[6] There is clearly overlap in these tasks, which is why the goal is to have a module that would take advantage of the currently available modules in the help desk software that exists, and integrate the interaction into the user's workflow to the greatest extent possible.

Modules commonly included in help desk software [such as Service Desk 7.5] include self – service pages and/or web portals with knowledge bases (FAQ's) and other limited interaction support tools.[7] They also have suggestion boxes via email request pages and custom request forms. They may offer simple standardized reporting of request types to IT, as well as more detailed reports to IT, but the user commonly only sees the statistics about how many requests they have made and whether they were responded to (not necessarily if they were answered). These systems will also keep track of business rules and notification rules which are helpful in managing information flow. Finally they include user surveys that are helpful with determining the effectiveness of their system, and API Integration to keep track of other pieces of the system (such as this project) that might be added on to the help desk software. Standard database methods are the most common, both in proprietary and open source forms.[8] [9] Other types of database organization are sometime used as well, such as the wiki format with ZwikiTracker™ and Trac™. These are all elements of the overall system described above, but the added effect of triaging interactions to be developmental, instructional, problems, or chatter about functionality will improve the system function in a way that is much different than a help desk alone.

Cost Estimate and Funding Sources

To create software of the complexity we have described for our suggestion system, it will take several weeks of development and testing time. A project of this level of complexity would be considered a small project. Determining costs for a software project requires an understanding of several variables. To come up with a reasonable estimate, one would need to understand how complex the project is, compared to others they have written before, determine how much domain specific knowledge is required to fully create and test the system, identify the length of the project, and determine if there are any specific requirements to use a particular programming language, database, or hardware setup.[10] For our project, we can assume we will be using the most mainstream technology, and thus the most easily

[5] http://itmanagement.earthweb.com/service/article.php/3295251
[6] http://www.helpdesks.com/chart.htm
[7] http://www.manageengine.com/products/service-desk/it-help-desk-software.html?gclid=CKqT0-L2q5wCFSQeDQodz2Ge7w
[8] http://www.helpdesks.com/chart.htm
[9] http://www.opensourcehelpdesklist.com/
[10] http://ssrn.com/abstract=569875

staffed. We can assume we will write the software in either C#, or Java, and we can use either a SQL Server database, or an Oracle database, with no special hardware requirements.

When estimating the cost of a software project, one needs to determine if work will be done with resources you already have, or if you will use a contractor, an offshore company, hire a new full time employee, or purchase existing software, or customize software you already have. Each of these options has their own inherit strengths and weaknesses, as well as their own costs. Off shoring costs an average of $10 to $30 an hour, which is potentially the cheapest option, but comes with its own downsides.[11]Using a contractor can cost around $50 an hour, up to $120 an hour. For a project of the scale we are proposing, and with the technologies we are planning to use, it would make the most sense for us to us one of our own staff programmers. If we were planning on tackling a larger project than we are, or if we were using a specialized technology, or if our staff programmer was busy with a more important project, it would make sense to use an alternative person to write our software, but for the sake of our project, we assume that is not the case.

For our project, since it is such a simple project, we have estimated that it will take between 3 – 4 weeks of development and testing, and that our staff programmer costs $40 an hour. That comes out to somewhere between $4,800 and $6,400. Since this is a staff developer, we could factor in the other benefits they receive, such as healthcare, and 401k, but we are not including that in this comparison, since that person would be earning those no matter what project they worked on for us.

Once users have begun using our system, there will be 2 main modes of use. The first is that the user will have a question that needs to be answered immediately or in a timely manner such as a single business day. The other mode is that the suggestions the users enter will be need to be analyzed for enhancements to other products which might occur on a regular release schedule of every few months.

The immediate use mode can best be handled by the organizations help desk department. Our application can know to route those types of questions there, which provides the users potentially 24/7 support for minor questions. Our system could also be used as a feeder into the organizations larger support system by creating tickets automatically on our user's behalf, or could simply send an email to the appropriate person or team. Depending on the volume received, the help desk should be able to handle the new load this will generate, but for budgeting reasons, we can assume we will need to add 10 hours a week of a help desk staffer to respond to these types of questions, at $12 an hour.

Analyzing the suggestions entered for longer term use, such as enhancements to system workflow, or new features requested, will occur on a regular schedule of perhaps once a month. Analysts can comb through the data, and identify either new features needed, or pass information on to trainers when functionality already exists, but users just don't know about it. These efforts can be assumed to be 10 hours a month at $40 an hour for the analyst.

Maintaining a system this small should be a trivial matter. After prolonged use, the users of our system will likely have some suggestions as to how we can tweak the user interface to be better suited to their needs. Perhaps pre-filling some fields, or adding more options to a drop down, or other minor improvements. The database might become slower after prolonged use, and indexes will need to be

[11] http://www.spiderkerala.com/kerala/software/

rebuilt, or new indexes might be identified altogether. A backup plan will also need to be created for the data in the database, as well as the applications source code. The server might need occasional restarts, and patches applied to keep it up to date. It could be estimated that our staff developer would spend around 5 hours a week with upkeep.

Since we are hoping to aggregate suggestions from our users for analysis, we will be deploying a centrally hosted database and we will have a single application that all users of our system will use. For this reason, implementation will be very simple. The application would simply need to embed a link to our suggestion site, somewhere noticeable, or near the help link. Most applications allow for such embedding. Once the link is embedded, it would also be useful for the application to pass on some information to our suggestion system, so that the user doesn't need to do so much typing. Things such as user name and email, as well as application being used, and version are all data that can be passed into the suggestion system easily. This could be expected to take 8 – 10 hours of our staff developer's time to create a single DLL (dynamic link library) that can handle that context sharing. Then it becomes a simple matter of using that module in the link embedding.

When we add all the figures listed above, we arrive at a grand total of $5,120 - $6,720 for the software, plus a weekly upkeep of $420 for the help desk, programmer, and analyst time. These costs are just for the software, and do not include purchase of any new hardware, or licenses you may need. The first years total cost can be calculated to be between $26,960 and $28,560.

Paying for this project presents us with several options. If we use a staff programmer, then it is likely we need to find things for them to do to keep busy, so this work is already paid for with their salary that has been budgeted for that year. Since our software is meant for internal use for the users of our EMR, this is not going to be a revenue model, or a way to generate new income. One way that the cost of this software could be defrayed would be to apply for a grant. Since the suggestion system we are planning to write will help us determine the usage of our EMR system, we could apply for a grant to study the results more formally, and publish them for others to use.[12]

Potential Criticism and Responses

Q: Most large institutions already have some sort of help desk software running already.
Why should I change this to yours if I already have one?

A: The system we are proposing is different from the normal help desk software. First, this software is meant to be integrated within the program, so that it is closer to the user at the time of the problem, which will make the user more likely to use such a system, and much more often. Our system will allow users to not only log problems they see, such as errors, but they can also enter in enhancements they would like to see, or suggest alternative workflows that will help them with their daily work. They will also receive response to their entry, in most cases right away. If a user asks a question, then an email response can be sent back to them, or the system can find an appropriate response from its previously asked and answered questions. This software is also going to be targeted to health care, and won't be generic. This will enable the system to prompt users for relevant information, and not incidental information which is not pertinent to the problem a hand. Finally, our system can act as a feeder into the

[12] http://www.njcleanwater.org/dobi/pressreleases/pr060601.htm, http://www.wpi.edu/news/20089/medrecords09.html

organizations existing help desk software by automatically opening tickets on the user's behalf, or by sending emails to the appropriate person or group.

Q: A centrally hosted system will not provide the same level of uptime as a distributed application, and will create a single point of failure for all the users. Why do you feel that a centrally hosted system is best suited for this application?
A: With today's technology, it is very reasonable to assume we can achieve the five 9's of uptime that we have all become accustomed to. We can use a single server to host the database, and the application. The database will be able to avoid deadlocks with the proper use of ATOMIC principles, and with proper index maintenance, as well as data backup and retention policies. The application can avoid slowness with periodic recycling of the host process, to thwart any memory leaks that may occur, and by using proper threading and resource locking; there will be no problem in handling the number of users we are expecting. Using the proper RAID setting, our hard disks will be able to fail over with out losing data, or shutting our application down to our users.

Q: Why is a system like this even necessary? Didn't we purchase the right EMR system for us?
A: Much thought and effort went into choosing our EMR system, with the assistance from all interested parties. However, no software is perfect right out of the box, nor could we be expected to have foreseen every workflow from every department. Putting in place a suggestion system like the one we are proposing, will allow us to capture inefficiencies from the users of the system directly, and at the time the problem occurs. With careful study of all the data our application receives, we will be able to make our EMR system work even better for us.

Conclusion

It was Arthur C. Clark who said that, "Any sufficiently advanced technology is indistinguishable from magic".[13] Many information technology projects can benefit from some magic, and clinical information systems are certainly among them, but relying on advanced technology for that magic may be taking one's eyes off of the ball. There are a lot of ways that these complex systems succeed and fail of which we are not aware, and it is that awareness we seek with the Suggestion Box. We propose the magic of relationships and communication as a route to that awareness, and the Suggestion Box as one route to those relationships and that communication. The Suggestion Box, deployed as suggested above, is an elegant mix of high tech and high touch and a simple view into a complex dynamic supported by technology but driven by people. It will help connect system users to system designers and system builders. We can use the magic of those connections to realize greater success, better outcomes, and increased satisfaction.

[13] http://en.wikipedia.org/wiki/Clarke's_three_laws