

Charles Ziegler
Site Director
Belleville Customer Interaction Center
Solectron Global Services

INTRANET
IMPROVEMENT
PROPOSAL

Christopher Reed
Owner and Project Team Manager
BlueLightning Studios

CONFIDENTIAL/NON-DISTRIBUTION AGREEMENT

Blue Lighting Studios

PLEASE MAKE A PHOTOCOPY FOR YOUR FILES AND RETURN THE SIGNED ORIGINAL TO
THE MUSC FOUNDATION FOR RESEARCH DEVELOPMENT

Project: _____

BLS Representative: _____

Date: _____

WHEREAS, BlueLightning Studios (hereinafter BLS) manages intellectual property; and

WHEREAS, the BLS holds certain proprietary information relating to the above referenced technology (hereinafter TECHNOLOGY); and

WHEREAS, (Solectron Global Services) (hereinafter COMPANY) wishes to receive proprietary information for the sole purpose of analyzing and evaluating the TECHNOLOGY for possible licensing and commercial use;

THEREFORE, to assure the BLS that all such proprietary information transmitted to COMPANY by BLS or MUSC will be maintained as confidential for not less than five (5) years by COMPANY, COMPANY acknowledges and agrees as follows:

1. Proprietary information means any information relating directly or indirectly to the TECHNOLOGY not generally known to the public provided to COMPANY by the BLS or its assignors/inventors and conveyed in written, graphic, oral or physical form including but not limited to scientific knowledge, know how, processes, inventions, techniques, formulae, products, business operations, customer requirements, data, plans or other records, biological materials, and or software.
2. Proprietary information does not include information which COMPANY can demonstrate and document that:
 - a. was known to COMPANY or in its possession prior to disclosure by the BLS or its assignors/inventors;
 - b. was public knowledge or has become public knowledge through no fault of COMPANY;
 - c. was legally provided to COMPANY by an independent third party who has no obligation of secrecy to the BLS or its assignors; or
 - d. is hereafter independently developed by company without reference to the information from the BLS.
3. COMPANY may use the proprietary information only under the following restrictions:
 - a. COMPANY may duplicate or reproduce the disclosed proprietary information; if duplicated or reproduced in whole or in part, the disclosed information must carry proprietary notice similar to that which it was submitted to COMPANY;
 - b. COMPANY shall not duplicate or use the disclosed proprietary information to develop, manufacture or sell products or technologies for commercialization unless a release, option or license granting same is executed with the BLS;
 - c. Any person employed by the COMPANY utilizing the confidential information will be advised of, and is subject to, the conditions in this Agreement; and
 - d. COMPANY agrees to utilize any transferred material solely for the purpose of evaluation and will not distribute the material to any person external to the COMPANY without the prior permission of the BLS.

4. COMPANY understands that disclosure of the proprietary information could destroy the value of said information, and COMPANY agrees to protect it from further disclosure using at least the same practices and procedures as they would with their own proprietary information.
5. Proprietary information delivered hereby is experimental in nature. The BLS makes no warranties, representation or undertaking with respect to its utility, efficacy, nontoxicity, safety, or appropriateness for a particular purpose. COMPANY hereby agrees to defend, indemnify and hold the BLS and its assignor harmless from any loss, claim, damage or liability, of whatsoever nature, which may arise from possession or use of the proprietary information by COMPANY.
6. Upon completion of COMPANY's evaluation of proprietary information or at BLS's request, COMPANY will discontinue the use of and promptly return all proprietary information and will promptly return materials, samples or specimens embodying that information.

The BLS reserves the right, at its sole discretion, to terminate this contract without notice and require the return of all proprietary information and materials, if any, provided under this agreement.

Any action arising out of this Agreement shall be decided in Charleston County, South Carolina. This agreement shall be constructed under the laws of the State of South Carolina. In the event of a dispute under this agreement, the prevailing party shall be entitled to collect its reasonable attorney's fees and costs.

Nothing in this Agreement shall be construed to constitute the grant of a license to the undersigned under any MUSC or BLS patent, patent application or proprietary information.

The COMPANY agrees, to the best of its ability, that it will comply with all applicable state or federal regulatory requirements.

7. IN WITNESS WHEREOF, the BLS and COMPANY hereby accept the terms and conditions of this Agreement and hereby signify their acceptance by the signing their respective officers hereunto duly authorized on the day and year hereinafter written:

By BlueLightning Studios

By Solectron Global Services

Signature

Signature

Printed Name

Printed Name

Title

Title

Date

Date

TABLE OF CONTENTS

Introduction	1
How the Intranet is Used.....	2
Technicians	2
Management	2
Human Resources.....	3
Training	3
Conclusion.....	3
Anatomy of an Exemplary Knowledgebase	4
Fast and Ubiquitous.....	4
Analysis	4
Thorough and Comprehensible	4
Analysis	5
Thoroughness	5
Comprehensibility	6
Effects.....	6
Conclusions	6
Information Architecture	7
Analysis	7
Add Search	7
Maintain the Taxonomy.....	8
Updated Frequently	8
Analysis	9
Features and Benefits	10
Site Design and Content	10
Thoroughness.....	10
Well-Written Articles	10
Search.....	11
Logical Organization	11
User-Friendly Site Design.....	12
Navigation	12
Readability	12
Visual Design	12
Content Creation Tools.....	12
Permissions	13
Version Control	13
All Users are Writers.....	13
Rationale.....	14
Contributing: Conventional Style.....	14
Contributing: Wiki Style.....	15
Contrast: the Email Knowledgebase	16
Formatting and Templates.....	17
Simple Linking	18
Browser-Based.....	18
Monitoring and Reporting Tools	18
Web-Based Forms and Tables.....	18

Live Monitors	19
Centralized Data Storage	19
Quantitative Data.....	20
Qualitative data.....	20
Custom Report Generation	20
Walkthrough of Core Functionality	21
Other Core Features.....	22
Possible Future Features.....	22
Table: Features and Benefits by Role	23
Implementation	24
Staff and Space	24
Intranet Development Team.....	24
Content Creation Team	24
Usability Testing Participants	25
Training.....	25
Platform	25
Hardware.....	26
Programming Language.....	26
Legend.....	26
ASP.....	27
ASP.NET	27
ColdFusion	27
Java (JSP)	27
Perl	28
PHP.....	28
Recommendations	28
Content Management System	29
Feature Set.....	30
Comparison Table.....	32
TWiki: 1 st Choice.....	33
Tiki Wiki: 2 nd Choice.....	33
Typo3: 3 rd Choice	34
Bricolage: 4 th Choice	35
OpenCMS: 5 th Choice	36
Database.....	37
Microsoft Access	37
Microsoft SQL Server	37
MySQL.....	37
Oracle	37
PostgreSQL	38
Recommendations	38
Operating System.....	38
Windows.....	38
Linux and BSD.....	39
Recommended Implementation.....	41
Project Team.....	42
Team Manager: Christopher Reed.....	42
Web Designer: Jason Coleman	42
Web Developer: David Latremouille.....	42

Web Developer: Natalya Boriko	42
Design Consultant: Bronwyn Boltwood.....	43
Appendix: Workflow.....	44
New Content.....	44
Updates to Existing Content.....	45
Appendix: Sample Task Summary.....	46
Appendix: Text Markup vs. HTML	47
TWiki Shorthand	47
HTML.....	48
Appendix: Sample Live Monitors.....	54
Sample Queue Monitor	54
Sample Team Monitor	55
Appendix: Sample Excel Report.....	56
Appendix: Interface Ideas	57
Microsoft Excel	57
Microsoft Visio.....	58
Adobe Photoshop	59
Opera and Mozilla.....	60
Appendix: Recommended Reading	61
Content Management Systems and Knowledge Management	61
Information Architecture	62
Intranet Usability.....	63
Linux and Package Management.....	63
Programming and Software Development	64
Usability and Design	65
User Testing.....	65
Appendix: Pricing Sources	66
Operating Systems.....	66
Windows	66
Linux and BSD	66
Databases.....	67
Programming Languages.....	67

INTRODUCTION

Reading the title, you probably asked yourself why BlueLightning Studios (BLS) thinks that the intranet at Solectron Global Services' Belleville Customer Interaction Center needs to be improved. Surely it works perfectly well already? The Belleville intranet does work, but it does not fulfill its potential. It can and should be an indispensable tool for employee productivity, instead of merely helpful. By building a superior knowledgebase and other intranet tools, Solectron's return on investment from the intranet will be greatly increased. To present concrete evidence for our position, we have drawn on our experience at Compaq Canada's (now HP Canada) Nepean Call Center. When our team manager Christopher Reed began working there as a technical support specialist in 1999, they had a technical intranet site available, but it was neither complete nor optimized. Starting in early 2000, technicians and management worked together to rebuild the intranet to provide better access to the knowledge and tools they used on a day-to-day basis. Their new intranet was complete by the fourth quarter of 2000, and the effects were clear by the end of 2001:

- Talk time decreased to an average of 12 minutes from 15.
- Wrap time decreased from 85 to 55 seconds.
- Positive customer service feedback calls and letters increased. Some technicians were receiving them weekly, versus once a month on average previously.
- Fewer supervisor calls regarding complaints and technician performance issues were made.
- Quality scores increased by an average of 5 to 10 percent.

As you can see, technician performance improved dramatically. Compaq increased the call volume to a peak of over 100, 000 calls per month for most of the 2001 fiscal year, and the Nepean Call Center was named Compaq's Call Center of the Year for over two years.

Results like these require skilled technicians that can take full advantage of the resources offered them. We know that the technicians at Solectron are exactly that from Christopher Reed's recent experience on the floor, so we are confident that improving the intranet at Solectron will provide similarly pronounced results. It may provide even better ones, since we propose to apply promising new methods and tools together with the principles proven at Compaq. Most of these methods and tools spring from Ward Cunningham and Bo Leuf's Wiki Wiki Web. Their revolutionary collaborative approach worked so well that it has given birth to a genre of site publishing and collaboration software called wiki. When the improvements to the Belleville intranet have proven successful, then the same approach can be applied at other Solectron locations worldwide. Having hosted the pilot project will put Belleville firmly on the company map as a leader in innovation.

HOW THE INTRANET IS USED

The first step is to understand how employees at Solectron already use the intranet, and what kinds of intranet-based tools will help them in their work. Since the majority of employees are technicians, increasing their performance will have the greatest impact, so BLS has focused on their needs. However, we also recognize that management, human resources, and training are vital functions, and have considered what they can gain from improvements to the intranet at Solectron.

Technicians

Above all, technicians use the intranet to research issues in the knowledgebase of troubleshooting information. This resource should be the authoritative source of technical data. It needs to be universally accessible, easily updateable, and easily expandable. Technicians also need to read news and other information about the call center, such as changes to contracts or policies, available training courses, and employment or benefits information. In addition, second-level technicians monitor the queues to ensure that calls are going through smoothly. Lastly, webforms posted on the intranet can be a quicker and less costly way to put in routine internal IT requests.

The intranet serves technicians best by reducing the overhead of troubleshooting. It is much faster to fix a car in a well-equipped and organized garage, because you can find the right tool. Similarly, if troubleshooting information can be found and understood quickly and easily, then the efforts dedicated to finding and figuring out the tools to do the job can be applied to actually doing the job. After Compaq's intranet was redesigned and the technicians trained to use the new site, not only did service levels increase and call times decrease, but reliance on second-level technicians and external sites was reduced. Instead of constantly reinventing the wheel, technicians simply looked up and applied a known procedure. Building such a technical knowledgebase at Solectron will have the same positive effects.

Management

Managers monitor and grow the business: team managers at the technician and team level, service delivery managers at the contract and call center level. The key requirement for them to succeed in this is data. Vast quantities of data must be captured and then stored in a way that provides a single authoritative repository and makes the data available for creating business strategies and reports, yet safeguards it. A central database server backed up nightly and/or configured with a RAID array provides central storage and access, while keeping the data safe from disasters such as disk crashes or corruption. Since writing

reports is an important duty of managers, a custom report generator application that fetches data directly from the database will see heavy use. Managers also read, publish, and collaborate on information regarding various aspects or units of the business, such as contracts, policies, and procedures. Software that leverages the intranet to promote teamwork and manage knowledge and documents will quickly become indispensable.

Human Resources

The needs of human resources staff are not dissimilar to those of management. There is a great deal of data that must be collected, updated, and drawn upon, so a central database server will serve them equally well. Human resources must also create reports, so they too will appreciate a custom report generator. Similarly, they need to read, publish, and collaborate on information about the call center, so they would benefit from having a knowledgebase of their own. Some of this knowledgebase can be made available to the rest of the call center, such as the frequently asked questions, holiday schedules, or benefit information. The intranet can also provide a better way to share documents, so staff can work together more efficiently. Human resources also handles requests about schedules, work status, and so forth. Webforms can be developed so that employees can submit such requests online, providing faster processing of routine requests. Finally, an intranet-based job posting board, especially one permitting electronic applications, is a perfect intranet application for human resources.

Training

Training uses the intranet to read, publish, and collaborate on training materials and online courses, so they too can make good use of a tool to manage knowledge and documents. Training also needs to test their students' knowledge, and hold evaluations of courses and trainers. Webforms for tests and evaluations, configured to deliver results to the database server, will streamline this process and make the results more accessible. When creating reports to show the results of the training activities, Training can make use of the custom report generator to fetch and tabulate the information.

Conclusion

All staff in the call center benefit from software that facilitates teamwork with tools for the sharing and management of knowledge and documents through the intranet. A custom report generator will benefit all departments that make reports. Finally, all staff will benefit from a central database server to store necessary content and statistics.

ANATOMY OF AN EXEMPLARY KNOWLEDGEBASE

BLS has identified four properties necessary for an exemplary knowledgebase and their benefits. After describing each one, we will consider how the intranet at Solectron can be improved. Since the most people would be using the technical knowledgebase that we urge Solectron to implement, we have concentrated on technical examples, but the same principles apply to other departments.

Fast and Ubiquitous

Technicians on calls need information almost constantly, and as fast as possible. Rapid access to that necessary information improves technician performance, which ensures customer satisfaction and better metrics. In a computerized environment such as at Solectron, the best way to make a fast and universally available reference is to publish a knowledgebase over the intranet, and make sure that the intranet is available quickly from every desk at all times. Each department can easily have its own knowledgebase to share and manage information, using the same software on the server.

Analysis

The current intranet is sufficiently speedy, and is universally available. That may sound unexciting, but it is still important: speed and ubiquity are fundamental to an efficient intranet. Solectron has a solid foundation here.

Thorough and Comprehensible

To be useful, a knowledgebase must be thorough, and the information within it must be comprehensible. No one can memorize a solution for every problem, so solutions must often be looked up. If the technician cannot find all the information needed, or cannot understand it clearly, then he will have to rely on slower, scarcer, and more vulnerable resources. What might that technician consult?

1. **External websites.** These are slower to access and to navigate than the intranet, and can be cut off entirely if internet access is not working. These websites can also move, change, or disappear without notice, since they are externally controlled. When an external website cannot be reached, the technician is slowed down by having to figure out what happened and how to cope with it. If he was seeking crucial information, the call's progress will come to a grinding halt. We also need to remember that finding the right information on the Internet is slower and more error-prone, because there is so much irrelevant content to search and navigate through.
2. **Other technicians.** Most of the time other technicians will be on calls themselves, with no time to spare, so they are a very slow and scarce resource. Bob may know what you need to know, but he may not finish talking to Mrs. Fussbudget for another twenty minutes. Other technicians also

cannot give information they do not already possess or cannot find more quickly than the technician asking for help.

3. **Second-level technicians.** These are a better choice of consultant than other first-level technicians, since they are more knowledgeable, and assigned to help first-level technicians. However, there are only so many of them, so our technician may have to wait quite a while on hold for help, meaning that the customer has to wait too, both of them becoming more frustrated by the second. Another difficulty is that second-level technicians cannot draw on information they do not have or cannot locate quickly. While this pool of knowledge is larger than a first-level technician's, it is still limited.
4. **Team managers.** Team managers are usually consulted on questions of policy, which are less frequent. Even so, our technician's team manager might not be available right away, since she could be attending to other people or duties. Like other staff, team managers are limited to what they already know or can look up.
5. **Their own intelligence and imagination.** Flying by the seat of the pants does not guarantee disaster, but it should still be a last resort. A desperate or foolish technician, especially one faced with an impatient and demanding customer, might try to remember a complicated fix whose source is not available, or even make the fix up on the spot. Of course, if he makes a mistake, the customer could get quite angry! There is no way to avoid improvising for some calls, since new issues do arise, but technicians should not have to improvise frequently.

After thinking over these alternatives, it is clearly the best solution to post all the information technicians need in a knowledgebase on the intranet. Articles for it should be written in clear, simple language, with procedures worked out step by step, so that even a new technician – or possibly a customer – can follow it unassisted with confidence. Directions are no true substitute for experience, but they are much better than nothing at all. Quality instructions will allow technicians to perform unfamiliar fixes correctly, so they will be able to troubleshoot faster and better, increasing productivity and customer satisfaction. They will not need to consult external sites or other technicians for routine calls. Instead, these slower, scarcer resources will be reserved for cases that cannot be solved with normal measures.

Analysis

Thoroughness

The current intranet contains useful troubleshooting and product information for the various contracts. However, it is not as complete as it should be for maximum efficiency: we estimate that up to 80% of technical fixes and other resources are only available on external websites. These resources are always slower to access than resources on the intranet, and may be cut off at any time. We are sure that Solectron is already concerned about this scarcity, and understands its consequences: service levels drop, call times go up, and customers and technicians alike become frustrated by the wait times necessary to troubleshoot effectively. The situation worsens when some resources become entirely inaccessible – when Internet access is cut off, for example. We can all agree that this must be remedied. We suggest

that BLS and Solectron work together to reproduce the knowledge in these external resources in an internal knowledgebase. Technicians will then be able to research most issues on the intranet alone, making their performance much less vulnerable to external technical difficulties, and most availability issues can be dealt with in-house.

Comprehensibility

While some existing articles are well-written, others are difficult to use because of cryptic or confusing writing. There are also a number of dead links in the existing material. Dead links are a hurdle for people trying to use the article, regardless of their cause. They should be corrected as they are noticed. We know that technical writing is a difficult job, since even the clearest instructions can be misinterpreted, but Solectron is in the technical support business. Quality writing is not a luxury; it is mission-critical. Solectron will increase its business advantage by having easily understood articles for everything the technicians regularly troubleshoot.

Effects

Because the existing knowledgebase is limited and sometimes hard to interpret, first-level technicians often rely upon the second-level technicians for what should be routine calls. We are certain that you have already recognized that it is not cost-effective to have the most intelligent and knowledgeable technicians, who are being compensated commensurate with these qualities, act primarily as living encyclopaedias of common fixes for the first-level technicians. Second-level technicians are meant to research and troubleshoot emerging issues or intractable problems, but without a complete and comprehensible knowledgebase of routine fixes, necessity dictates that they will continue to be living encyclopaedias before all else.

Conclusions

Thoroughness and comprehensibility is the area where the most improvements are necessary. The bottlenecks we have discussed above reduce the efficiency of technicians, forcing longer calls and lower service levels. By fully populating an internal knowledgebase with high quality content, technician performance will be improved and customer satisfaction increased, because the information that technicians need will be available right away. They will not have to rely on slow external websites and a small number of second-level technicians. Another benefit of improving the quality of the articles is that when a simple-to-fix issue is likely to recur, the technician can send a copy of the article to the customer for reference. This will reduce the number of calls for repeated simple issues.

Information Architecture

Librarians and researchers know that organizing material in a logical, easily discovered, and easily navigated structure allows relevant information to be found quickly. The expert's term for this structure is *information architecture*. To be as useful as possible, then, a knowledgebase needs a well-designed information architecture. Some may think that search alone will make all information easy to find. We know from experience that it is a critical tool, but that search alone is insufficient. The knowledgebase should be organized according to a taxonomy (a system of categories used to classify articles) so that users can browse to find related material. Having such a taxonomy aids navigation and makes it easier to maintain and extend the database and to improve the search engine.

Analysis

Add Search

We need to consider how people, especially technicians, find information. Typically, they search first. If the search has disappointing results, they might try a different query. If the results were relevant but not perfect, and the articles are cross-referenced, then they might browse from article to article. If search fails entirely – for example, if the customer does not provide enough information to find useful results – then the seeker will analyze the information architecture, navigate to related sections and open promising articles until something matches the problem at hand. When these methods are exhausted, then they start culling through external sites or call for help.

An excellent full-text search is a must-have feature in a knowledgebase, because it is the most common and flexible method to find information, as shown by numerous usability studies.¹ It should be flexible, so that several terms, exact phrases, or one term and not another are all valid search queries, and include some advanced functions like Boolean logic (AND, OR, and NOT). The syntax should be as similar as possible to Google so that seekers can reuse their learning. The search function should use metadata to provide more relevant results. A good example of this is that if the search term is the name of a category in the knowledgebase's taxonomy, articles in that category should be considered more relevant. For example, an article that contains “init string” and belongs to the modem category would be ranked higher than articles from other categories. Of course, the article titled “Common Modem Init Strings” from the modem category should be at the top of the search results, because the search query is matched in both the title and the body, and it belongs to the appropriate category.

¹ See <http://www.useit.com/alertbox/20010513.html> and <http://www.useit.com/alertbox/9707b.html>. Remember that technicians can be considered to be advanced users, and will become experts for a site they use every day.

Maintain the Taxonomy

The current intranet is logically organized, but there is room for improvement. Good information architecture is crucial. Why? Because users are *busy*. They are here to complete a task, not to learn the site's stupid idiosyncrasies. Not noticing the site's information architecture is a mark of quality: if it is always obvious where you should go, then the structure is logical and appropriate.²

Browsing by category is especially good for problems that have no obvious starting point for troubleshooting. A technician working on a broken DVD drive might find a solution by drilling down to the correct category and using the optical drives troubleshooting flowchart, when she might not have found anything useful by just searching. This is the obvious reason why it is just as important to use a taxonomy to classify articles as it is to have a good search engine. The second and subtler reason to use a taxonomy is to take advantage of Metcalfe's Law (the value of a network grows by the square of the size of the network) by making it easy to find related articles via the taxonomy or through cross-references. Both these measures can noticeably speed up research when a problem is complex or hard to trace, which means shorter calls and higher service levels. Better yet, more and faster resolutions (especially for difficult issues) can generate increased customer loyalty. The final argument for well-designed information architecture is that not only does it make the intranet easier to use, it makes it easier to maintain. If the knowledgebase's taxonomy is kept clean and tidy, it will be easier to:

- Group the right articles together, because the categories are well-defined. Allowing articles to belong to multiple categories is a great help here.
- Cross-reference articles, so that technicians can quickly find related information covered elsewhere.
- Extend the database with new or revised articles, because the content creation team can keep better track of what knowledge is already in the database.
- Fine-tune the search engine for optimum performance by using articles' category metadata in determining search results and their rankings.

Updated Frequently

As we already know, the knowledgebase should be thorough. The corollary of this is that it must be updated regularly and promptly as new products and new technical fixes are developed. To make it easier to notice new content, there should be a space on the homepage highlighting and linking to freshly added or revised articles.

² See <http://www.useit.com/books/rosenfeldmorvilleforeword2.html> and *Information Architecture for the World Wide Web* (2nd edition), by Louis Rosenfeld and Peter Morville.

Analysis

Unfortunately, the intranet is not being updated as often as it should be. We understand how much work it is to keep a large knowledgebase up-to-date without sacrificing accuracy or good style, but the alternative is to spread new information by email and word of mouth. That cure is worse than the disease. It provides little control over the quality of the information, and it puts the burden of archiving and classifying it onto each and every employee. Having an online knowledgebase provides better quality control for much less work. Consider the amount of email you receive each day, and how much work it is to process it all. The problem is worsened by the fact that the standard mail client at Solelectron has only rudimentary email management capabilities.³ Our design consultant is also a helpdesk analyst, and previously worked with Christopher Reed at Compaq Canada. Researching and writing this proposal caused her to realize that many of the inefficiencies addressed here also exist at her current position. There too, most information is disseminated by email and word of mouth, so she has an insider's perspective on both methods:

I'm a technician at a single point of contact desk. Everyone in my group has a machine on the corporate network and another on the client network, with an email account and intranet access on each side. There are thousands of systems that we take tickets for, and the procedures change and multiply constantly. The intranet is always out of date because neither webmaster can keep up with that many changes, and because both organizations could take two weeks to approve an article whose subject matter might only last one week. New information is spread by email, on whichever network the sender feels like, but may never be added to either intranet, while the information on the two intranets may never have been spread by email. Therefore, none of my four knowledgebases is complete, though they have a lot of overlap. Even though I spend at least an hour a week archiving email by topic, I can never be sure that I have the freshest information. I also have to clean out my folders every six months or so to get rid of duplicates, old versions, and expired procedures. It's all a tremendous waste of time.

Finding the information I need is even harder, because Outlook 2000 can't search more than one folder at once. I track things down by hand most of the time, because it's faster than making the computer do it. Most often, I rely on memory – mine and my coworkers' – to know if something is known or not, since my troubleshooting information is permanently scattered over two intranets and two mailboxes. Sometimes we make mistakes that should have been avoided, and would have been, if there was one place where we could look up everything. With the current system – or lack of it – if our second-level technicians have the wrong information, then the rest of us will be working from that wrong information. I used to be on the same contract as Chris, where there was an intranet knowledgebase that was kept updated. When you needed information, you simply searched or browsed until you found the right article, instead of racking your brain to remember which mailbox, which folder, which intranet, which page where you saw it last, and *then* searching or browsing for it in the most likely places, one after another, until you found it or gave up, the way I have to now. Maintaining a central knowledgebase on an intranet website and regularly updating it is a much better method – it removes a lot of unnecessary work, so the technicians can be more productive.

³ In the mail client's defense, we must say that even Microsoft Outlook's extensive email management features are inadequate to deal with the ever-increasing flood. Email simply is not the right tool for this job.

FEATURES AND BENEFITS

The four qualities just discussed are the cornerstones of an excellent knowledgebase, but they need not be the whole of what BLS can offer Solectron. We have planned many other features that will improve the usefulness and efficiency of Solectron's intranet. They fall under three categories: site design and content, content creation tools, and monitoring and reporting tools. After the detailed description of the suggested improvements, there is a table summarizing how personnel in each department will benefit. We invite Solectron's employees to suggest other features that will help them do their work.

Site Design and Content

Thoroughness

A knowledgebase is only as useful as the information that it contains. To ensure that each department's knowledgebase contains everything its users will regularly need, BLS and Solectron will establish a content creation team of experienced Solectron employees. This team will populate the knowledgebases with quality content before launch (to eliminate the chicken-and-egg problem⁴) and maintain them thereafter. The goal is to have all the knowledge that the staff need posted on the intranet, where anyone with the right permissions can access it. By making the resources used in day-to-day business easily accessible, the staff will only have to consult external sites or other employees for especially difficult or obscure issues.

Well-Written Articles

Having all required information available on the intranet is pointless if that information cannot be understood. It is imperative that all articles be written in clear, simple language and an easily followed style. Procedures should be described step by step, and all information checked for technical accuracy. Different cases (e.g. Windows 98 vs. Windows XP) and different possible outcomes for the same issue should be handled by writing multiple sections or multiple articles, instead of writing one-size-fits-all instructions that never fit anyone perfectly.

One type of article we plan to include that deserves special notice is the troubleshooting flowchart. These will be articles showing how to troubleshoot a particular type of problem (e.g. no

⁴ *Chicken-and-egg problem*: when instituting a new system, there is usually little to no content in it, yet people avoid using it until there is enough content for it to be useful. Of course, there will never be more content if people do not contribute it. Kick-starting the new system by migrating existing content and adding new content before launch can make a huge difference. See <http://twiki.org/cgi-bin/view/Main/TWikiSuccessStoryOfTakeFive>.

sound) or component (e.g. optical drives) when there is very little information. By asking questions or performing tests in a certain sequence, the type of failure is progressively narrowed down, until it is clear what the appropriate fix is, or until the technician can safely conclude that this is not a known issue. The chart will link to the recommended fixes, to speed up troubleshooting and call times. These charts are excellent refreshers for experienced staff, and teach less experienced technicians how to troubleshoot better by ordering the steps and explaining the logic behind them.

Search

As we saw earlier in our information architecture section, search is an essential tool for all employees, but especially for technicians, because they are trying to sort through a huge amount of information very quickly. There will be a prominent search box on every page. A full-text search engine optimized to find and correctly rank the most relevant results is one of BLS's top priorities.

Logical Organization

We have already discussed the benefits of a sound information architecture. Logical organization makes it easier to use content, by grouping and cross-referencing related articles, and to manage content, by organizing what is known, which makes gaps easier to spot. The knowledgebase will have a taxonomy tailored to the content, and be extensively cross-referenced. The decision whether to use a content management system for the site's backend, and which one, will make a large difference to how much of the cross-referencing can be automated.

Browsing articles by category should be an efficient way to find information, with special advantages over searching. Articles should be able to belong to multiple categories, so that users will be able to find them in all the logical places to look. Categories will not be nested too deeply, since that can be more hindrance than help: it is quicker to scan a slightly longer list of articles than to go down the wrong fork and only realize it another three levels down, when the customer mentions a key fact he did not think was important.

The taxonomy will also be used to optimize searches. Articles will be assigned a unique ID, which can be searched for. The article's category metadata will be used in finding and ranking search results. If the search term is also a category name, or a keyword belonging to a particular category, results from that category will be considered more relevant. Finally, an advanced search function could allow the technician to select specific categories to search. This feature is known as scoped search.

User-Friendly Site Design

After putting so much effort into the flesh and bone, BLS will not fail to make the skin user-friendly. Usability is an art of details. Here are the methods we will use to achieve the goal.

Navigation

- Proven navigation tools such as hierarchical menus and breadcrumb trails will be employed.
 - Navigation links should be persistent. A vertical strip on the left or right dedicated to navigation links is a common and well-understood method for complex sites. Animated pop-up menus that disappear when the mouse leaves them are to be avoided.
 - Categories should be easily expanded or contracted. The Windows Explorer folder tree is a good example of the desired behaviour.
 - Ideally, the user's current location should be highlighted in the menu.
- If proven desirable through user testing, search results pages will open articles in new windows. Technicians may find this useful because it reduces navigation time when checking which are the most suitable articles.

Readability

- A highly readable font that clearly distinguishes between similar characters such as O/0, S/5, and i/l/I/1 will be used.
- Intranet pages will not use font sizes below 9 pt, and body text should default to 12 or 14pt.
- Font size will be easily changeable by users via browser controls.
- Text will contrast strongly with its background.

Visual Design

- Minimalist visual design, so that the site does not distract the users from the content.
 - Graphics will be used sparingly so the site loads quickly.
 - Plain backgrounds and quiet colours will be used, to eliminate distractions such as loud colours, animations, or highly textured backgrounds.
 - The browser scroll bars will not be colourized.
 - Peripheral elements will be clear, but not distracting.
- Clickable targets will obey Fitt's Law,⁵ and will be sized according to importance.
- Optionally, there could be multiple stylesheets, so that personnel could choose what look they preferred. Further customization options could be introduced if desired.

Content Creation Tools

Before proceeding, we need to define a term. Wiki Wiki Web is a name based on the Hawaiian term for *quick*. It is often shortened to **wiki**. A wiki is a website where the users work together to collaboratively document and discuss a topic. It is a knowledgebase, a documentation site, and a forum all at once. Some pages can be polished, finished documents, while others are still conversations that will

⁵ For an excellent explanation of Fitt's Law, see <http://asktog.com/basics/firstPrinciples.html> and <http://asktog.com/columns/022DesignedToGiveFitts.html>.

eventually become a finished document, and yet others combine document mode and thread mode. The server software used to run such a site is a wiki engine. Most of the content creation tools described here are common features in wiki engines, or are inspired by them. Version control, text markup, simple linking, an extremely simple workflow, and soft security are all hallmarks of wiki.

Permissions

The improved intranet will have permissions that determine what material users can view, edit, or delete, because sometimes it is necessary to restrict access to certain things, or to edit another's work. Each Soletron employee will be assigned a personal intranet account, just as they receive a LAN account. The account's permissions will provide *access control* by determining what the user can view. They will initially be set according to department and skill set, and then changed as required for the employee's duties. We would like to emphasize that access should not be restricted unless genuinely necessary; otherwise the purpose of the intranet – to share information – will be defeated. Permissions will also provide *editing control*, by determining whether a user is a **guest**, a **contributor** or an **administrator** for a given area of the intranet. Everyone is considered a contributor for all their areas of expertise, but only an administrator will be able to make drastic changes such as deleting articles. A user with guest status in an area may only view existing material and cannot post. As with restricting access, guest status should be used sparingly, since overusing it will defeat the purpose of the intranet.

Version Control

One of the major features of the content creation system, version control works by creating an article history. Previous versions are saved in the database every time a change is made, as well as a log entry stating who did what when. All changes are tracked and easily reversible, simply by rolling back to a previous version. Using the log, the source of the change can be traced, since each employee will have a personal intranet account. Another advantage of versioning is that comparison tools (often called difference engines) show exactly what changed each time. Most wiki engines delegate this function to the standard Unix diff utility (freely downloadable, even for Windows), or have their own built-in difference engine. Some content management systems implement version control for files attached to articles, which will also be useful for Soletron.

All Users are Writers

One of the chief advances we propose for the intranet is that all personnel be able to submit content in their area(s) of expertise, be it a new article or a revision to an existing article. Each page will

have an “Edit this Page” link leading to a form where you can write up and submit your changes. There will also be an “Add New Article” link in the site-wide navigation controls, which leads to a completely blank version of the same form. We will discuss two possible approaches to handling the contribution process after that point below. First, we would like to explain the philosophy behind this policy.

Rationale

Willingness to share knowledge is a critical factor in both workplace efficiency and team spirit. Coworkers should be helping each other, and the most effective way to help each other in a call center is to post what you know. Knowledge in a person’s memory only benefits the possessor, but knowledge posted on the intranet benefits everyone. Therefore Soletron should make it easy to improve the content on the intranet, to maximize the benefits of having an online knowledgebase. It should also be rewarding, because contributors are making an effort to help their coworkers and Soletron instead of looking only to themselves. Good contributions and good contributors should be recognized. For example, someone who regularly adds quality material could be offered a position on the content creation team. By encouraging personnel to add content, not only will more knowledge be collected overall, but it will be more accurate and up-to-date. The content creation team will be able to devote more time to new, unreleased material, or to maintaining the information architecture. Between the efforts of the users and the content creation team, the intranet will not only stay up to date, it will thrive.

Contributing: Conventional Style

This method requires another permissions class, called **editor**, to be created, to reduce the number of administrators needed (and thereby, the potential for serious error). Everyone will be able to contribute content, but only editors and administrators will be able to post it. New content and changes to content will be submitted to the content creation team (who will have editor permissions) for them to approve and post it. We have diagrammed possible workflows for new and existing content in the Workflow appendix. Of course, this approval process will take time, and could delay the posting of necessary information.

The summary:

1. Submitted content is sent to the appropriate group within the content creation team.
2. An editor reviews the submission and revises it as necessary.
3. The article is integrated into the database.
4. The article is posted.

We have planned a task summary view to streamline this approval process. Taking a leaf from Microsoft, editors will have a personalized page summarizing their current tasks, much as Outlook Today summarizes the day’s appointments, tasks, and new messages. There will be a table of the editor’s current tasks, and a second table for the team’s current tasks. These lists could be reordered by their various

criteria, such as status (new, rejected, revising, integrating, approved), author, title, unique ID, date submitted, and date due. Please see the appendix Sample Task Summary for a mock-up. Each listing will have a link to the details page for that submission, where the editor can see and change the body text, the attributes listed above, and the area(s) and categories for the submission. It should be possible to save working versions and compare them to other versions of the article, including the one currently posted.

Contributing: Wiki Style

The assumption that the readers are intelligent adults capable of contributing responsibly has always been a cornerstone of wiki culture. Accordingly, all members are granted editing privileges, rather than merely contribution privileges. A key advantage of wiki style contribution is that staff can make necessary changes on the spot, rather than waiting for an approval process to grind slowly along. However, “everyone’s an editor” is clearly a startling concept, since newcomers to wiki commonly ask “why is there no security?” The answer is that wiki operates on a soft security model, preferring to rely on people’s natural behaviours instead of on artificial controls.⁶ There are in fact a number of security measures in place. The first is version control. The second, and more powerful, is *setting expectations*. The first part is that “cracking” a system that is already wide open provides neither thrills nor reputation. It is nigh pointless to post “M1cr0\$0f7 sUx, l337 |-|ax0r5 u5e L1|\\ux” when the next sensible reader can and will roll back to the previous version of the article with a few clicks – or worse, will request you to support that opinion with a tightly-reasoned argument instead of invective. The second part is that granting editing permission implies trust. Most people, when granted trust, will do their best to live up to it rather than destroy it. Being trustworthy is a fundamental part of being a decent human being, and most of us want to be decent human beings. The last part is that, when the obstacles to destruction are removed, one finds that creation is much more challenging, engaging, and prestigious than simple destruction.

If Solelectron opts to use wiki style contribution, the following measures will provide sufficient security and accountability:

- Each employee will be assigned an account by the system administrators, so each account’s owner is always known.
- Only administrators will be able to delete articles.
- Version control will ensure that there is always be a backup copy. Damaged or deleted articles can be repaired by rolling back to the previous version.
- The entire database will be backed up regularly.
- The difference engine will pinpoint how an article changed between versions.
- The changelog associated with each article will record who changed what when. This audit trail means that a determined saboteur is only singling him or herself out for a swift termination.

⁶ For more discussion of this issue, see <http://tavi.sourceforge.net/WhyWikiWorks>, <http://usemod.com/cgi-bin/mb.pl?SoftSecurity>, and <http://c2.com/cgi/wiki?WhyNobodyDeleteWiki>.

- Revised articles will be highlighted on the Recent Changes page. Some systems can send email notifications as well, so administrators need never be out of the loop.

Concerns about content quality can also be laid to rest. The content creation team will monitor contributions via the Recent Changes page or optional email notifications, and promptly check new content for accuracy, style, and technique, and edit it if necessary. Content will be just as accurate and well-written as with the conventional style, but will be updated much more quickly and freely for Solectron's having empowered its employees.

Contrast: the Email Knowledgebase

If Solectron still harbours doubts about the wisdom of having all users be writers, BLS invites Solectron to consider how the suggested contribution methods compare with their current solution of a static intranet combined with repurposed email, as described in the analysis under Updated Frequently.

Feature	Email	Existing Intranet	Conventional Contrib	Wiki Contrib
<i>Single authoritative knowledgebase</i>	No	No ⁷	Yes	Yes
<i>Search</i>	No	No	Yes	Yes
<i>Cross-references</i>	No	Few	Yes	Yes
<i>Logical organization</i>	Per user	Weak	Strong	Strong
<i>Content checked for accuracy and style</i>	No	Yes	Yes	Yes
<i>Access control</i>	No	No	Yes	Yes
<i>Editing control</i>	No	Yes	Yes	Yes
<i>Users can delete articles</i>	Yes	No	No	No
<i>Version control and difference engine</i>	No	No	Yes	Yes
<i>Regular backups</i>	No	Yes	Yes	Yes
<i>Easy rollback for damaged articles</i>	No	No	Yes	Yes
<i>Centrally or locally stored</i>	Locally	Centrally	Centrally	Centrally
<i>Burden of maintenance</i>	Each user	Intranet team	Communal	Communal
<i>Frequency of updates</i>	High	Low	Moderate	High
<i>Speed of updates</i>	Excellent	Moderate	Good	Excellent
<i>Complexity of workflow</i>	None	High	Moderate	Low

It should be clear that all users are already writers at Solectron, and that they are currently operating without the safeguards built into even the simplest wiki engine. Updating the existing intranet is a lengthy process, so email becomes the *de facto* knowledgebase. Email provides no quality control, nor is there any form of access or editing control. There is no search function, no cross-references, and no inherent logical organization. Since each user has a locally stored repository that must be maintained by the user, there is no single authoritative source. Because there are no backups and no version control or difference engine, users can accidentally delete articles, and there will be no hope of recovery, let alone a quick rollback to the previous version. Maintenance time to keep the knowledgebase clean is multiplied

⁷ As is clear from technician comments, the current intranet site is not meeting their needs fully. Since it is poor in needed resources, lacks search, and is rarely updated, it cannot be used as the single authoritative source.

by the total number of users. Email's only advantages are speed and simplicity. Migrating to an online knowledgebase based on a content management system will provide similar speed and simplicity, but with the safeguards and centralization that email lacks.

Formatting and Templates

Instead of requiring all personnel to learn HTML, or leaving those who do not know it out of the content creation process, a good content management system provides ways for such users to contribute. One common method is the much-beloved what-you-see-is-what-you-get (WYSIWYG) editor. MS Word, MS Frontpage, and Macromedia Dreamweaver are common examples of the type. The editors included in content management systems are nowhere near that fancy, to everyone's relief. Some commercial ones let the administrator limit what markup users are allowed to use. The alternative is text markup, which is commonly employed by wiki engines for its cross-platform compatibility and low system requirements. Certain patterns are recognized as formatting and are turned into the correct HTML upon rendering. It may sound intimidating, but it is quite easy in practice – think back to how you formatted plain text emails, back in the good old days before HTML mail. Most systems provide for links, anchors, paragraphs, headings, bold, italic, multi-level numbered and bulleted lists, and tables. Other capabilities, such as automatically generated tables of contents or the ability to interpret HTML markup inline, are included in some systems, such as TWiki's. For an example, please see the appendix Text Markup vs. HTML. Text markup encourages contributors to focus on the content, instead of on the precise shade of lime green text that will best complement their magenta stationery. Since it is easy to read in its raw form, even complex articles can be worked on easily in a good text editor,⁸ instead of needing a graphical HTML editor.

The content is not the only part of a webpage. There is the look and feel, navigation areas, and branding, all of which need to be more centrally managed. Most content management systems take care of this by using a template system. The templates define global elements, navigation, layout, and presentation separately from the page-by-page content. Changes to those global or section-wide elements are much more easily accomplished, because only a handful of template files need be changed, rather than editing hundreds or thousands of hand-coded pages. Only those in charge of designing the templates will need a graphical HTML editor such as Dreamweaver for working on the intranet.

⁸ Metapad (<http://liquidninja.com/metapad>) is an excellent completely free, full-featured text editor.

Simple Linking

Simple linking, including *accidental* linking, is one of the most important features of any wiki engine. It is one of the primary reasons that we advise using a wiki engine for the intranet's content management system. As you can see in the appendix Text Markup vs. HTML, there is a multitude of ways to make links. The simplest is to use a WikiWord, and the next simplest is to use a [[free link]] or to drop in an URL. The benefit is that the writer no longer has to look up URLs for internal resources by hand, or wait through a pretty but slow wizard. If the writer mentions a useful article by its WikiWord name, then it is automatically⁹ linked by the wiki engine's text parser. It just happens – no fuss, no muss. Some wiki systems also have a backlinks feature: the page knows about the other pages that link to it, which means that an automatic cross-reference tool comes built-in. All these things make it exceptionally easy to create a network of cross-references within the intranet, which, in turn, will make the knowledgebase livelier and more useful.

Browser-Based

The last major feature of the content creation system is that it will be accessed through the web browser. Neither contributors nor administrators will require any specialized software simply to access the system. Instead, content can be fully edited in the browser (or a text editor, if a content management system with text markup is employed), although administrators in charge of creating site templates will still require a good graphical HTML editor.

Monitoring and Reporting Tools

This group of features is meant to simplify administration and other paperwork. One is that forms and tables will be made web-based. Two is that data will be centrally stored on a database server. Three is that a custom report generator will offer a drag-and-drop interface to lay out tables, and will be able to automatically fetch and calculate their contents by connecting to the database server.

Web-Based Forms and Tables

Forms and tables will be made web-based, so that any employee with the correct permissions will be able to view them from any desk without specialized software, so long as she is logged into the computer. They will streamline the collection and use of data, because the conversion from paper to

⁹ The term is quite common in the wiki (and programmer) community. It is a more colourful version of “automatically.” See the Jargon File's definition at <http://www.cnam.fr/Jargon/jargon.html?71>.

digital – and illegible handwriting – will no longer slow down data entry. Some possible uses for these tables and forms are:

- Displaying phone switch statistics.
- Siebel offline case notes form to store and input ticket information:
 - If interface with Siebel permits, automatically adds the data to Siebel once it is back online. (Otherwise, the system should remind the technician to put the notes into Siebel.)
 - Email notification to technician in case of autoinput failure.
 - Email notification of ticket number to customer, if email addresses are captured.
- Input of technician data such as audits, compliments, or complaints.
- Input and display of online tests and evaluation forms for training courses.
- Input and display of schedule-related information:
 - Webforms to put in timesheets and schedule requests (availability, time off).
 - Tables to display old timesheets, schedules, requested time off, or attendance events.
 - Display of time spent logged into the phone (meant as an aid to filling out timesheets).
- Processing work status changes.
- Job postings and applications.
- Input and display of health and dental plan claims.
- Internal IT tickets that do not require diagnosis before dispatch.

Live Monitors

Live monitors are an example of a web-based table and of the potential of the central database. Taking advantage of the fact that phone switch statistics are captured and archived in the database, these dynamic webpages compile that data and present it in real-time in your web browser. We have included two screenshots in the Sample Live Monitors appendix. By setting them to auto-refresh, the information displayed will always be up to date. To view them, you need only the address and the correct permissions on your account. Based on our experience at Compaq, we suggest that the live queue and team statistics be made accessible to their teams, so that technicians can monitor call volume and their own auxiliary code usage more carefully.

Centralized Data Storage

The improved intranet will provide a database for centralized data storage, which will be backed up regularly. This approach is superior to storing important information on personal systems, where a hard drive crash can erase years of work. The central database will accumulate a comprehensive history, which can be effectively displayed and analyzed with the custom report generator discussed in the next section. Many kinds of data can be archived, which will permit a very complete picture of operations. Also, since all data will be stored centrally, in an organized fashion, it will be easier not only to change what data is collected, but also to create flexible, extensive reporting functions that can call on and analyze many types of data. We have put together a list of what types of data should be collected, but it is

only a prototype. Solectron will of course let BLS know if needed types are missing or some included types are extraneous, and it will be modified to suit.

Quantitative Data

- Calls offered
- Calls accepted
- Calls abandoned
- Talk times
- Wrap times
- Hold times
- Aux times
- Cases solved (all)
- Cases solved (on first call)
- Cases solved (without escalation)
- Attendance stats
- Compliments
- Complaints
- Audit scores
- Survey scores

Qualitative data

- Quality audits
- Customer surveys
- Full text of compliments and complaints
- Personal notes by managers on particular individuals and groups

Custom Report Generation

We think that the improved intranet's showstopper feature for management will be the custom report generator. This application will draw on the database of call center statistics to create flexible custom reports. To fully meet management's needs, this feature must be developed iteratively, with feedback from its future users, so it cannot be made available right away. We will begin with a Quick Reports page that will generate the most commonly needed reports automatically. Even after the full reporting tool is developed, this page will remain as a convenience.

The full-fledged report generator will be extremely flexible, able to fetch any number of the quantitative data fields available to the user for a user-specified list of technicians and/or groups, over a user-specified date range, at a user-specified granularity, with functions such as averages and service levels, and create a table and/or trend graph. Imagine that a team manager wants a report showing the talk, wrap, aux and hold times for all the technicians in his team, and for the team as a whole, for the past week, with daily and weekly averages. This may be a simple report, but it actually requires a great deal of data from the phone switch, and a surprisingly large amount of calculation. The report must figure the

daily averages for each statistic for each technician, as well as calculating the weekly and team averages. We noticed while creating our sample report in Excel (which is included in the Sample Excel Report appendix) that it is quite trying to enter elapsed time. To have Excel automatically calculate the averages using formulae, the elapsed time must be in the correct number format, but the exact procedure is not fully explained in the help.

Walkthrough of Core Functionality

We think that a network-aware application designed for creating custom reports would be far superior to struggling with Excel and other inappropriate solutions. The report generator will have a drag-and-drop graphical user interface, and will be able to connect to the database server to fetch information. Our goal is to let the user direct the action, while making the system do the work of finding and juggling the information.

Before we begin, we would like to note that there is an Interface Ideas appendix with screenshots and comments, in case there is any difficulty visualizing. Having said so, please imagine a blank report. Blank reports begin in Arrange view, so it will look very much like an Excel spreadsheet, except for the set of axes visible somewhat toward the top left corner of the grid, to help users place items. But to the left, there will be an items palette such as you find in Visio. This groups items, which are simply field names or data types, into different categories, so that you can quickly find the item you want. The items can be dragged from the palette into the work area, and then dragged and dropped within the work area to lay out the report.

Most items will need to have their properties set to be useful. For example, the All Technicians item will have properties dictating which group you want to see all the technicians for: Team 1, the XYZ contract, the ABC call center, and so on. When the user switches into Formatting view, this item will magically expand itself, to show all the technicians from the chosen group. This feature is the reason that the Arrange and Formatting views are distinct. Users will need a convenient way to set these all-important properties, so on the right, there will be a properties palette. If possible, the same property controls should be accessible through the Properties item in the item's context menu.

After the user is satisfied with the arrangement of fields, the program can be commanded to fetch the data from the server, so that the report can be printed or exported for inclusion in a document such as this one. (The report generator is meant solely for creating tables, not word processing.) The report can also be saved in the program's native format as a layout, allowing users to create custom report templates. If our team manager needs to create our sample report once a week, she can lay out the report as usual, but instead of setting the date fields to the exact date necessary, she can set them to an appropriate variable, such as "this week". Then our team manager opens that saved layout once a week, has the

report generator fetch the data, exports or prints the result, and her weekly report is ready in moments.

Layouts are a powerful convenience.

Other Core Features

- Quick Reports webpage will generate commonly needed reports automatically.
- Access control via logins, just as with the rest of the improved intranet.
- Print or export to standard file types, or save as a layout for future use.
- Standard editing and formatting commands such as cut, copy, paste, and delete.
- Templates will autoformat cells according to their status as headings or data when exporting or printing reports.

Possible Future Features

- Create trend graph from report.
- Transpose (make the vertical axis into the horizontal axis and vice versa).
- User can change report formatting in Formatting View:
 - Font properties: font, size, bold/italic/underline, block caps, colour.
 - Justification: left, center, right, justified, and top, middle, bottom.
 - Gridlines: on/off, line width, line style, cell background colour.
- Batch functionality: fetch and export/print a group of layouts to complete your weekly reports in one step.

We believe that this would be a suitable feature set, but we strongly encourage management at Solelectron to suggest features and refinements they would find useful.

Table: Features and Benefits by Role

Role	Duties	Tools
1 st level technician	Answer questions and troubleshoot issues quickly and accurately.	More complete knowledgebase will make it easier to research issues.
	Write useful notes.	• Easy to cite unique article IDs. • Siebel offline web form for saving case notes during Siebel issues.
	Stay up-to-date on supported products.	New content will be added regularly and highlighted on main page.
	Monitor own performance.	Personal statistics webpage (access restricted to the subject technician) that can summon standard reports from database concerning: • Talk, wrap, hold, and aux times • Calls taken • Solved cases • Audit and survey scores • Text of compliments, complaints, quality audits, and surveys
	Keep abreast of queue.	Able to view live monitor pages for team and queues.
	Contribute technical content for posting.	Content creation tools; permissions for Technical area.
2 nd level technician	Assist 1 st levels with troubleshooting and questions.	More complete knowledgebase will make it easier to research issues.
	Research and resolve exceptionally difficult issues.	
	Research and document emerging issues.	
	Create, edit, & approve technical content for posting.	Content creation tools; permissions for Technical area.
	Monitor queues.	Able to view live monitor pages for teams and queues.
	Write useful notes.	• Easy to cite unique article IDs. • Siebel offline web form for saving case notes during Siebel issues.
Team manager	Stay up-to-date on supported products.	New content added regularly and highlighted on main page
	Monitor own performance.	Personal statistics page as for 1 st level techs.
	Monitor queues.	Able to view live monitor pages for teams and queues.
	Monitor individual and team performance.	• Team and technician profiles with space for confidential notes. • Reporting tools.
Service delivery manager	Create, edit, & approve policy-related content for posting.	Content creation tools; permissions for Policy area.
	Monitor team and contract performance.	• Contract and team profiles with space for confidential notes. • Reporting tools.
Human Resources	Create, edit, & approve contract and policy-related content for posting.	Content creation tools; permissions for Contract and Policy areas.
	Inform other personnel about health coverage, workplace policy, job postings, etc.	Create site for HR knowledgebase. • Subset of pages publicly viewable; others restricted to HR staff. • Job posting bulletin board with online application forms. • New content will be highlighted, and notices will be cross-posted to other areas when critical information is added/updated.
	Create, edit, & approve HR-related content for posting.	Content creation tools; permissions for HR and Policy areas.
	Create reports about payroll, benefits, etc.	Custom report generator.
Training	Process requests such as work status changes, health plan claims, job postings, etc.	Webforms will automate submitting routine tickets.
	Conduct training of other personnel in classroom settings or through self-teach courses.	Create site for Training knowledgebase. • Subset of pages publicly viewable; others restricted to Training staff. • Post course materials and/or entire online courses. • New content will be highlighted, and notices will be cross-posted to other areas when critical information is added/updated.
	Evaluate effectiveness of training.	• Online tests can be automatically marked or processed by humans as applicable. • Test results, enrolment data, and user surveys will be archived in database like other call center data. • Reports can be generated and easily viewed in browser or sent to other locations as necessary.
	Create, edit, & approve training-related content for posting.	Content creation tools; contributor/editor permissions for Training areas.

IMPLEMENTATION

What would be required to implement these improvements to Solectron's intranet? While there are certain minimum requirements, BLS is prepared to offer Solectron a variety of options, so that Solectron can choose the most appropriate platform for the intranet and control costs.

Staff and Space

In order to carry out the improvements described earlier, BLS will need the following personnel and office space for them in the building.

Intranet Development Team

This group will consist of the BLS project team outlined later in the report. Based on the projected size of this initiative, a five person team is appropriate, since there will be enough work to require all members, without overworking any one of them. It consists of the team manager, two web developers skilled in web application programming, a web designer, and a design consultant.¹⁰ The development team will also need to consult with Solectron's incumbent intranet team, to better integrate the existing material and tools into the improved intranet. Regarding seating requirements, there are two locations that would serve well. One is located in the human resources section; another is located in the quality section. Both of these would suffice in housing the staff needed.

Content Creation Team

As the editors of the intranet, this team will consist of experienced Solectron employees, including one or more second-level technicians from each contract, and personnel from management, human resources, and training. The group's first project will be to populate the improved knowledgebase by migrating existing content and reproducing external content. BLS realizes that these employees cannot be spared from their regular duties for extended periods. However, a reasonable amount of time must be allocated to content creation duties if the knowledgebase is to be of high quality. We expect that the content creation team will be given a number of hours per week to perform editing tasks uninterrupted by other duties, in accordance with the workload. The project will occupy a significant proportion of time initially, but will require less time once migration is complete, much as the first hill on a roller coaster is always the highest. The members will work from their existing desks, but the team should have regular

¹⁰ The design consultant will be working remotely and will not require seating.

meetings to make sure that requirements are being met and work going smoothly. A virtual meeting space such as forums or chat is also desirable.

Usability Testing Participants

In order to optimize usability, BLS proposes to use an iterative design process, widely known to be an industry best practice for developing websites and software.¹¹ Groups of three to five users will be required periodically to test the knowledgebase and other tools as development proceeds. User testing will bring design problems to light quickly, so they can be corrected as they arise, minimizing costs and maximizing usability.

Training

With such extensive changes being planned, it is important for Solectron to teach its employees how to take advantage of and contribute to the improved intranet, especially if wiki-style contribution is used. Wiki culture, which includes reading, discussing, and revising, is quite different from the usual web culture of merely browsing. Cultural changes are necessary for wiki to take root properly and become part of the staff's way of working. Community experience proves that personal training, in a combination of one-on-one and classroom settings, is the best method to encourage this.¹² It need not be an extensive course; even an hour or two can make the difference, especially when well-respected and oft-consulted staff are trained personally in the benefits, so that they become evangelists for the new system. The BLS team will teach Solectron's training department how best to use the improved intranet and what it is capable of, and they will teach the general employee population.

Platform

BLS has investigated a large number of different platform options, the best of which will be described in this section. The intranet's platform will consist of these components:

- Hardware
- Operating system (OS)
- Database
- Programming Language
- Content Management System (CMS)

¹¹ Please see the User Testing section under the appendix Recommended Reading for sources.

¹² See <http://twiki.org/cgi-bin/view/Codev/ArgumentsAgainstTWikiOnIntranet> and <http://twiki.org/cgi-bin/view/Codev/HowToGetInternalBuyInForTWiki>.

These components must all work together in order to extract maximum performance from the server, and we will consider them in this order: hardware, programming language, content management system, database, and operating system. The content management system is the topmost layer. There are many different CMSes, and the CMS will have a marked impact on the results of the project. It is such an important part of the improvements that components like operating systems and databases are in fact secondary. So rather than allowing an initial leaning for a particular OS to limit Solectron's choices for the content management system, we will instead allow the CMS's requirements to select operating system and databases for our recommended platform.

Why are we discussing programming languages before content management systems? The programming language the CMS is written in must be taken into account for several reasons. One is that it will have a direct impact on the CMS's performance. Two is that the rest of the intranet's dynamic pages should be written in the same language if possible, to lower the overhead on the server. Three is that there are many excellent open-source CMSes. If one of these is chosen, then BLS can freely extend and customize the CMS source code to better meet Solectron's needs. It is only good sense to choose a language that is suited to Solectron's purpose and already well known to the BLS team.

Hardware

The first necessity is to have computers to serve up the improved intranet. Fortunately, Solectron already has a web server that will satisfy this requirement. When the load increases enough to lower performance, we recommend that another computer be set up as a database server.

Programming Language

Legend

The programming languages will be assessed for the following things.

Source:	Where the programming language is obtained.
Platforms:	What operating systems the language can run on.
Webservers:	What webservers the language is compatible with, and which are preferred.
Footprint:	How well the language uses memory and other server resources. A smaller footprint equals better performance.
Databases:	What databases the language can talk to, either natively or through layers or drivers. Native support improves performance.
Learning Curve:	How difficult the language is to learn and use.
Familiarity:	How well known the language is to the BLS team at this time.
CMSes:	Availability of content management systems written in the language, according to BLS's research.

ASP

Source:	Comes as part of the IIS framework included in Windows NT4, 2000, and XP.
Platforms:	Aforementioned versions of Windows.
Webservers:	Microsoft IIS
Footprint:	Large. ASP wastes memory, bandwidth, and disk space because it treats all values as 32 bit, even when they are much smaller. Imagine how much space would be wasted if each piece of cutlery in your kitchen had a drawer to itself.
Databases:	Native support for MS SQL Server only. Other databases must use an ODBC layer. This results in a higher cost (with SQL Server) or lower performance (from using other databases).
Learning Curve:	Moderate.
Familiarity:	Well known to the BLS team.
CMSES:	Few CMSES are written in ASP; and those that are are primarily commercial.

ASP is also Solelectron's current web programming language.

ASP.NET

Source:	Free download from Microsoft; recent Windows Server versions.
Platforms:	Windows.
Webservers:	Microsoft IIS (included with Windows NT4, 2000, XP and 2003).
Footprint:	Medium to large. ASP.NET addresses ASP's memory issues to some extent, but it still is not as efficient as some other languages.
Databases:	Native support for MS SQL Server only. Other databases must use an ODBC layer. This results in a higher cost (with SQL Server) or lower performance (from using other databases).
Learning Curve:	Steep. As well as being more difficult than ASP, learning ASP.NET is complicated by the fact that it has two different scripting languages (VB.NET and C#).
Familiarity:	Moderately well known to the BLS team.
CMSES:	Even fewer CMSES are written in ASP.NET; and those that are tend to be commercial.

ASP.NET is also the successor to Solelectron's current web programming language.

ColdFusion

Source:	Commercial application sold by Macromedia.
Platforms:	Windows and Linux.
Webservers:	Includes a webserver.
Footprint:	Medium. Like most languages, ColdFusion must be parsed, but there are no add-ons to speed this process, as there are with Perl and PHP.
Databases:	Supports every database being considered.
Learning Curve:	Gentle. ColdFusion, like HTML, is tag-based, and its structure is very clear.
Familiarity:	Moderately well known to the BLS team.
CMSES:	BLS found no CMSES written in ColdFusion during research.

Java (JSP)

Source:	Free download from Sun.
Platforms:	Linux, BSD, Windows, and others.
Webservers:	Apache, IIS, and other major webservers.

Footprint:	Large. Java includes many features not required to meet Solectron's needs, and requires an application server. With a Java-based CMS, the server will need more memory to serve the same number of users as it could using a PHP-based CMS.
Databases:	Supports every database being considered.
Learning Curve:	Steep. Java is a very extensive language with many related technologies, such as J2EE applications servers.
Familiarity:	Moderately well known to the BLS team.
CMSes:	Some CMSes are written in Java, but their target audience is typically system administrators who already have a J2EE application server set up. Solectron does not have this infrastructure in place currently, and this proposal does not plan any tools specifically requiring it.

Perl

Source:	Free download from http://www.cpan.org/ .
Platforms:	Linux, BSD, Windows, and others.
Webservers:	Apache, IIS, and other major web servers. Apache has a module called mod_perl, which adds a Perl interpreter to Apache, increasing performance from 20% to 2000%, depending the code's purpose and the coder's skill.
Footprint:	Moderate. Like Java, Perl includes many features not required to meet Solectron's needs. Mod_perl does alleviate this, but it is not quite as efficient as PHP for the purpose at hand.
Databases:	Supports every database being considered.
Learning Curve:	Moderate. Perl is an extremely flexible language with a long history, but it is also well-known. It is also an ancestor of PHP, which is better known to the BLS team.
Familiarity:	Moderately well known to the BLS team.
CMSes:	Some CMSes are written in Perl, and provided that the desired feature set is implemented, should be seriously considered.

PHP

Source:	Free download from http://www.php.net/ .
Platforms:	Linux, BSD, Windows, and others.
Webservers:	Apache, IIS, and other major web servers. Apache has a module called mod_php, which adds a PHP interpreter to Apache, increasing performance. While not as dramatic as the benefits of mod_perl, it increases performance significantly.
Footprint:	Small. PHP is a very efficient language. Native database support and Apache's mod_php lessen the already small footprint.
Databases:	Natively supports most databases and optimized for MySQL and PostgreSQL. Requires an ODBC layer to access MS SQL Server and Access.
Learning Curve:	Gentle. PHP is a web-specific child and subset of Perl with some changes. ASP programmers in particular will learn it easily because many functions are similar.
Familiarity:	Well known to the BLS team.
CMSes:	Many CMSes are written in PHP, especially open-source ones.

Recommendations

BLS unhesitatingly recommends using PHP as the programming language of choice. It is fast, with low overhead, and an excellent partner for MySQL, since it has native support. Not only that, but it Apache's mod_php further increases its already impressive performance. Perl is a mature, versatile and

efficient language, but it is not quite as good a choice, since Solectron will not be making full use of its impressive capabilities (no web project does) and it is not as familiar to BLS as is PHP. Java too is very flexible, with a thoroughly debugged codebase, but it has more features than necessary, and its footprint is increased by the requirement for an application server. The extra time and software necessary to deploy Java is not cost-effective for this project. ColdFusion does not offer sufficient features to tempt, given its high price. Last but not least, if Solectron prefers to use a strictly Microsoft platform, then ASP.NET should be the language of choice, because of its improvements on ASP.

Content Management System

There is a wide variety of content management systems in the marketplace, ranging from freely downloadable portal packages to expensive consultingware for enterprise. When researching, BLS weighed the benefits of writing a custom CMS for Solectron versus using an existing product. Given the feature set required and desired, BLS would need approximately one year of development time (including testing) to write the program in the language of Solectron's choice (e.g. ASP.NET). This would be a costly option, but allow the most flexibility and customization. Using an existing CMS will allow the improved knowledgebase to launch much sooner, but there are some issues to consider. First is that there may not be a CMS that has all the features needed or wanted, yet does not cost the Earth. Second is that the CMS will have particular system requirements that must be met for it to function. Third is that if a closed source or commercial product is selected, Solectron will either have to pay the vendor for any additions to the feature set, or accept that such modifications may not be possible at all.

BLS researched a large number of content management systems while working on this proposal.¹³ We could not find any commercial CMS that came even close to our recommended feature set that did not cost tens of thousands of US dollars, and we did not want to force Solectron to incur that kind of cost unless they preferred it. Accordingly, we concentrated our research on free, open-source products. Our initial impression, and our conclusion, is that free, open-source CMSes are better than all but the very best (and usually the most expensive) commercial offerings, both in absolute terms and in cost-effectiveness. Our five favourite CMSes are all free, open-source projects that can run on Linux or BSD as well as Windows,¹⁴ so using one will lower costs while increasing customizability and reliability above any comparable solution. Two are written in PHP, two in Perl, and one in Java. This bears out the general pattern that the majority of free, open-source CMSes are written in PHP or Perl, some in Java, but very few in ASP or ASP.NET. Similarly, our favoured five CMSes are more stable, better tested, and better supported on Linux or BSD than on Windows, although they do all run on Windows except for Bricolage.

¹³ Our notes are available upon request.

¹⁴ With the exception of Bricolage, which requires a Unix-based environment like Linux, BSD, or Unix proper.

Most of them also require or prefer Apache as the webserver and MySQL or PostgreSQL as the database. These preferences are partly due to financial advantages, but also to the philosophical ones – open source programmers often do not want to use closed-source technologies in their personal projects, and most free, open source CMSes begin as personal projects. Since many wildly successful dynamic websites demonstrate the stability and performance advantages of open source solutions, we have no qualms recommending the same for Solectron.

Feature Set

Needed Features

Full-Text Search	CMS's search function searches article titles <i>and</i> the article's full text.
• Boolean support • quotes for phrases	Supports Boolean operators such as AND, OR, and NOT. Searches for quoted strings as literal phrases (e.g. searching for "modem init strings" finds articles with the exact phrase "modem init strings").
Multilevel Categories	CMS allows categories within categories.
• # of levels allowed • multi-category membership	How many levels of categories are allowed by the CMS. Whether an article can belong to multiple categories.
User Accounts	CMS requires users to login with a username and password and can store preferences and personal information.
• uses domain accounts	CMS can identify the network logon being used and use the appropriate CMS account. (Similar to how Outlook recognizes your network logon and logs into your mailbox automatically.)
Permissions	CMS has permissions on articles, to enable or restrict read and/or write access.
• view, edit, & delete control • users & groups • flexible & fine-grained	Permissions control whether an article can be viewed, edited, or deleted by the person logged in. Permissions are based on users and groups. Easier and faster to administer. Permissions are flexible enough to handle complex real-life scenarios. One possibility is two users from the same group needing different permissions for the same documents. Another is needing to grant a user different permissions on document A than on document B, when A and B are both in category X, and permissions are set by category.
Version Control	CMS backs up and labels previous versions, and allows articles to be rolled back to an earlier version.
• saves >1 level of history • shows who changed what • recent changes list • email notification of changes	Backs up more than one previous version. Very desirable. Uses a difference engine or other mechanism to compare versions. Maintains a list of recently edited/created articles. Notifies selected users of changes to selected articles by email.
Audit trail/Changelog	CMS keeps a detailed log showing who changed what when.
Wiki-style contribution	Allows informal, wiki-style contribution by users.
Conventional contribution	Allows conventional contribution, with formal workflow and editing phase.
Browser-based content creation	Content can be created in the browser, without needing special client software.
Simple linking	CMS has simple method to create links with minimum effort.
Simple editing	CMS does not require user to learn HTML to contribute content.
• text formatting rules • WYSIWYG editor	Uses text formatting rules to format article, which CMS turns into HTML. CMS offers what-you-see-is-what-you-get editor for contributors to format articles via point-and-click, much as in Word.
Templates	CMS has a templating system to centrally manage and automate presentation, layout, and navigational elements.

Nice Features

Attachments/File Galleries	Virtual space to keep files. Users can download/upload them according to permissions. Some systems attach them to articles.
• version-controlled	Version control for the above files, to prevent accidental overwrites etc.
Cross-reference features	CMS has backlinks pages or other method to find articles referencing other articles.
Trash (undelete for articles)	Just like the Windows Recycle Bin. Deleted articles can be undeleted in case of mistakes, so long as the Trash was not emptied.
Advanced workflow features	More complicated workflows are possible.
Page Statistics	CMS records page statistics and keeps track of popular pages, orphan pages, etc.
Sandbox	Area where new users can play and learn how to use the CMS without damaging real material.
Forums	Allows non-realtime conversations on topics. Very good for questions and answers, or as a space to discuss new issues.
Posting board	Place to post notices about problems such as a client network outage, or widespread issues with new models.
Quizzes/Surveys	Will facilitate online tests for the training department, and let Solelectron gather employee opinions.
Chat	Application for realtime conversations, as in instant messaging. Could be used to obtain help from second level technicians.

Comparison Table

Platform	Bricolage	OpenCMS	Tiki Wiki	TWiki	Typo3
Operating System ¹⁵	Linux/BSD	Any	Any	Any	Any
Programming Language	Perl	Java	PHP	Perl	PHP
Web Server ¹⁶	Apache	Not applicable ¹⁷	Any	Apache	Any
Application Server	mod_perl	Tomcat, Jboss, Jrun		mod_perl	
Database ¹⁸	PostgreSQL	MySQL, Oracle	MySQL	Any or none	Any
Other Requirements	HTML: Mason, SOAP, SSL				

Needed Features	Bricolage	OpenCMS	Tiki Wiki	TWiki	Typo3
Full-Text Search	Yes	Yes	Yes	Yes	Yes
• Boolean support	No	No	No	No	No
• quotes for phrases	No	No	No	No	No
Multilevel Categories	Yes	Unknown	Yes	Yes	Yes
• # of levels allowed	Unlimited	Unknown	Unlimited	Unlimited	Unlimited
• multi-category membership	Yes	Unknown	Yes	Yes	Yes
User Accounts	Yes	Yes	Yes	Yes	Yes
• uses domain accounts	No	No	Next version	No	No
Permissions	Yes	Yes	Yes	Yes	Yes
• view, edit, & delete control	Yes	Yes	Yes	Yes	Yes
• users & groups	Yes	Yes	Yes	Yes	Yes
• flexible & fine-grained	No	Yes	Yes	Yes	Yes
Version Control	Yes	Yes	Yes	Yes	Yes
• saves >1 level of history	Yes	Yes	Yes	Yes	Yes
• shows who changed what	Yes	Yes	Yes	Yes	Yes
• recent changes list	Yes	No	Yes	Yes	Yes
• email notification of changes	No	No	No	Yes	No
Audit trail/Changelog	Yes	Yes	Yes	Yes	Yes
Wiki-style contribution	No	No	Yes	Yes	No
Conventional contribution	Yes	Yes	Yes	Yes	Yes
Browser-based content creation	Yes	Yes	Yes	Yes	Yes
Simple linking	Yes	Yes	Yes	Yes	Yes
Simple editing	Yes	Yes	Yes	Yes	Yes
• text formatting rules	Unknown	No	Yes	Yes	No
• WYSIWYG editor	Unknown	Yes	No	No	Yes
Templates	Yes	Yes	Yes	Yes	Yes

Nice features	Bricolage	OpenCMS	Tiki Wiki	TWiki	Typo3
Attachments/File Galleries	Yes	Yes	Yes	Yes	Yes
• version-controlled	Unknown	No	Yes	Yes	Yes
Cross-reference features	No	No	Yes	Yes	No
Trash (undelete for articles)	No	No	No	Yes	Yes
Advanced workflow features	Yes	Yes	Yes	Yes	Yes
Page Statistics	No	No	Yes	Yes	Yes
Sandbox	No	No	Yes	Yes	No
Forums	No	No	Yes	No	Yes
Posting board	No	No	Yes	No	No
Quizzes/Surveys	No	No	Yes	Yes	No
Chat	No	No	Yes	No	No

¹⁵ “Any” means that the CMS supports Windows, Linux, and BSD.

¹⁶ “Any” means that the CMS supports IIS and Apache.

¹⁷ For this CMS, webserving duties are handled by the application server.

¹⁸ “Any” means that the CMS supports MS Access, MySQL, Oracle, PostgreSQL, and SQL Server.

TWiki: 1st Choice

This CMS was expected to impress, and impress it did. TWiki provided tremendous inspiration: the decision to use a CMS for the site, the strong partiality to and recommendation of wiki, and approximately half of the desired feature set spring directly from our having learned about TWiki. It is very well known in the wiki community for being an enterprise-strength wiki with abundant features.

TWiki has almost everything except the kitchen sink, including calendars, drawing tools, workflow, and more, and we are sure that one of these days some bright child will find a way to write a kitchen sink plug-in. TWiki can store content in ordinary text files on the server, or in any database that a Perl:DBI module¹⁹ has been written for. The TWiki Forms toolkit provides a flexible method to implement categories and other form-based applications, including simple workflows, and there is a plug-in for creating more complex workflows. Between its out-of-the-box functionality and the wealth of plug-ins and add-ons available, TWiki is the most complete match to our feature set, for both the necessary and the nice. Without a doubt, it is the very best choice.

However, nothing is perfect, not even TWiki. The simple search built into it does not follow common syntax, does not use Booleans (it uses Perl regular expressions, which are quite cryptic-looking), and does not use quotation marks to denote literal phrases. Some time should be devoted to developing a better search. This functionality is very much in demand, so contributing it to the TWiki project would bring recognition. TWiki also lacks LDAP/domain username support, which would permit it to recognize you in TWiki according to your network or workstation login. TWiki is written in Perl, so modifying it will take a little more learning time than it would for a PHP CMS, but it is not a major hurdle. Despite its blemishes, TWiki's solid foundation earns it a first place finish.

Tiki Wiki: 2nd Choice

Tiki Wiki was the programmers' favourite during the CMS research, since it is written in PHP and well known in the open-source CMS community. It lost first place by an inch, since TWiki is a very mature and featureful CMS, while Tiki Wiki is somewhat younger and not quite as closely targeted to this application. Unlike the other favourite candidates, Tiki Wiki has bells and whistles in spades.

This is not to say that Tiki Wiki skimps on necessary features. The most interesting is that the next version (of which a beta is already available) will have LDAP support for domain-based usernames. Tiki Wiki is the only CMS being considered that offers this support. With the exception of email notifications, Tiki Wiki has a very good version control system. It also has all the features that we need for categories, including multi-category membership, which will improve searches, category browsing,

¹⁹ DBI stands for DataBase Interface, and it is used by Perl to send and receive information to and from databases. TWiki's page on it is at <http://twiki.org/cgi-bin/view/Codev/PerlDBI>. There is also a database plugin that has been tested with MySQL, at <http://twiki.org/cgi-bin/view/Plugins/DatabasePlugin>.

and cross-referencing. Another benefit is that Tiki Wiki is developed in PHP. PHP is a very efficient scripting language, and is BLS's preferred scripting language, so it will be quicker and easier for BLS to modify Tiki Wiki than TWiki, which is written in Perl.²⁰

Tiki Wiki does have drawbacks too. The first is that MySQL is currently the only supported database, though support for others is on their to-do list. Admittedly, BLS believes that MySQL is the best database for this application, but if Solectron prefers another database, the database calls would need to be rewritten, increasing development time. Just like the other CMSes, Tiki Wiki does not support searching with Boolean operators or quoted phrases, so some work would have to be done to improve the search function. Modifying Tiki Wiki to completely match the desired feature set is an option that deserves serious consideration.

Typo3: 3rd Choice

Typo3 is the sleeper hit: research was almost complete when we realized how strong a contender Typo3 is. Their website is both polished and replete with excellent documentation. We are quite serious about that "excellent." Typo3's documentation is not only plentiful, but substantial with content instead of fluffy with marketing, and it is unusually well-written. Of all the CMSes researched, only TWiki and Tiki Wiki had even half as much documentation, and not always of such a high quality. This is especially notable in a market that tends to post the download and let the hapless user stumble around the system mostly blind, and often does not even trouble to fully detail the system's features for the prospective user.

Typo3 has most of the necessary features and most of the nice ones. It finishes third solely because it has met worthy opponents: Typo3 is missing a few important features that Tiki Wiki and TWiki possess, such as wiki-style contribution. It seems to lack cross-reference features and email notifications, but we would not be surprised if they suddenly turned up – as they say in the feature list, "in fact, we have left out a large number of smaller features here."²¹ Typo3 possesses a WYSIWYG content editor when used from Internet Explorer, so if this is a do-or-die feature for Solectron, Typo3 stands head and shoulders above the other CMSes. It also has several different editing modes at increasing levels of complexity, according to the user's ability. Typo3 supports all major databases, allowing Solectron more choices for the platform. Finally, it is written in PHP and therefore easier for BLS to modify.

Like the other candidates, Typo3 suffers from the usual missing features of Boolean operators and quoted phrases for searches, and LDAP authentication for recognizing network logins. Neither does it have email notifications of changes, or cross-reference features. Multi-category membership is achieved

²⁰ While PHP is BLS's preferred scripting language, the team is skilled in all languages being considered.

²¹ Please see http://typo3.com/Feature_list.1243.0.html. Unfortunately, the demo site was offline during this proposal's writing because of insatiable demand. The developers are trying to find it a new home with enough server power and bandwidth.

by inserting shortcuts to existing pages, so it is merely a good workaround rather than a proper implementation. Altogether, Typo3 lags slightly behind TWiki and Tiki Wiki in terms of the ideal feature set, but it places an exceptional emphasis on user-friendliness for naïve users who cannot or do not want to deal with plain text. If front-end editing (as in Red Dot CMS) or WYSIWYG editing is critical, remedying the other lacks will result in a superior CMS.

Bricolage: 4th Choice

Bricolage, which runs on top of the Mason templating engine, was the dark horse in this race. At first glance, it looked to be cryptic and lacking too many important features that this project needs. That impression was wrong, although our impression that Bricolage is short on documentation is quite correct. This CMS may be lacking the bells and whistles found in other products that we researched – most of which did not make the cut – but it meets most of the requirements, and brings some unusual and useful features to the table.

The first of these is that Bricolage can use SOAP (Simple Object Access Protocol). In the CMSes researched, this feature is unique to Bricolage. This simple protocol uses little bandwidth, and excels at broadcasting messages, relaying chat, and other functions that require frequent small exchanges of data, such as requesting and receiving a dictionary definition. It is not needed for any essential functions, but will be a useful tool to add features over time. The second is that Bricolage is one of the more secure CMSes, with native SSL support. On the web, information is typically sent in the clear. SSL encrypts the information being sent and decrypts it upon receipt, so that sensitive data is safe from snooping. Other CMSes cannot do this out of the box, and would need another program on the server. The downside of SSL is that built-in or not, every packet of data needs to be encrypted and decrypted. This extra overhead slows down the server. For an e-commerce or similar secure site, SSL is mandatory unless the group running it enjoys hackers and lawsuits. However, Solectron is running an intranet knowledgebase, not an online banking site. Native SSL support is nice but not especially important.

The first and largest disadvantage is that Bricolage only supports the PostgreSQL database. PostgreSQL is a good database used by many websites for storage, and it can handle heavy loads of customers. However, as documented in the database section, PostgreSQL has a higher overhead than MySQL. A more flexible database acceptance system, as found in other CMSes, would be useful. We also could not determine whether Bricolage uses text formatting rules, a WYSIWYG editor, or some other system. The other missing features are the common lack of Boolean operators or quoted phrases for searching, and LDAP/domain username support. Bricolage is more oriented to a news site than to a knowledgebase, but is a strong, battle-tested choice.

OpenCMS: 5th Choice

This product was another nice surprise in the CMS comparison. OpenCMS is designed for wide-scale deployment in enterprise, rather than for an easy-to-use collaborative knowledgebase. OpenCMS offers a secure environment. Because it is written in Java, the source code is not clear text (as it is with Perl or PHP) but instead is already compiled into machine code. This makes it almost impossible to decompile the source code or to modify the file unless you have the original .java file itself. This approach has a performance benefit too: since it is already machine code, the source does not need to be parsed extensively. OpenCMS has a more detailed permissions structure than the other candidates and excellent version control because of its enterprise emphasis. Its ability to work with Oracle databases as well as MySQL is a good feature, but we should remember that Solectron might not use Oracle or MySQL for this project.

While OpenCMS brings enterprise-level features such as load balancing and automated backups to the table, it does not have other features that would be more useful for this project. As with the other CMSes, advanced search features such as Boolean operators or quoted phrases are not supported, and there is no LDAP/domain username support. Much more importantly, we could not find any documentation regarding category features, so without acquiring and configuring a spare machine to its requirements and installing OpenCMS, we have no way to verify its capabilities in this area. Unlike every other final candidate, documentation in general was quite scarce. The final detraction is that OpenCMS is written in Java. Java applications like this one require an application server to run, which puts a greater overhead on the server machine. Most application servers are mature products ranging from Open Source (Apache's Tomcat/Jakarta, JBoss), to large commercial vendors (BEA's Websphere, Oracle's Application Server 9), so support and credibility is not an issue, and it is not too difficult to recruit a system administrator familiar with the technology.²² If Solectron already had this Java infrastructure set up, OpenCMS would be more interesting, but as explained in the description of Java as a programming language, this solution is over-engineered for the task it will be performing. Our conclusion is that OpenCMS leaves out frills to concentrate on useful features for enterprise projects, but that it is better suited to building a large, carefully-regulated website than to creating an easy-to-use collaborative knowledgebase.

²² Of the BLS team, only Chris Reed and Natalyia Boriko are familiar with Java application servers. They have worked with only the open source servers. No team member has experience with the commercial grade servers.

Database

Microsoft Access

Advantages

- User-friendly and easy to maintain.

Disadvantages

- Meant as a personal or small group productivity tool, not as server software.
- Extremely low performance; only a handful of individuals can view the same file at once.
- Does not use standard SQL, so data cannot be easily migrated to other databases.
- Only runs on Windows.
- Initial and maintenance cost is high. This being a Microsoft product, there are concerns regarding licensing and being locked into an upgrade cycle.

Microsoft SQL Server

Advantages

- Integrates well with a Microsoft/ASP.NET platform.
- Polished administration tools.

Disadvantages

- Most features provided are not required.
- Client Access Licenses are required, which increases cost.
- Only runs on Windows.
- Initial and maintenance cost is high. This being a Microsoft product, there are concerns regarding licensing and being locked into an upgrade cycle.

MySQL

Advantages

- Runs on a wide variety of platforms, including all those being considered.
- Uses standard SQL, so data can be easily migrated to other databases.
- MySQL Control Center graphical administrative utility eases maintenance.
- Very stable and fast running on Linux and BSD.
- Freely downloadable. Commercial licenses are unlikely to be necessary.

Disadvantages

- Less stable and fast running on Windows.
- Primary interface is still the command line.

Oracle

Advantages

- Very secure and reliable.
- Runs on many platforms.

Disadvantages

- Extremely costly given the application. Oracle has many features not required for our purposes, such as transaction support (used to keep bank computers from losing data in system crashes).
- Uses its own syntax for programming, so knowledge of Oracle commands is not interchangeable with knowledge of standard SQL commands and vice versa.

PostgreSQL

Advantages

- Runs on a wide variety of platforms, including all those being considered.
- Stable and fast running on Linux and BSD.
- Freely downloadable.

Disadvantages

- Not as stable and fast when running on Windows.
- No native Windows port currently available – it must be run under Cygwin – although a native Windows version is in development.
- Includes features not required by this project, such as transaction support.
- Command line interface only.

Recommendations

The main function for this database will be to store and fetch information quickly on demand for roughly 500 users. Speed and reliability are required, but advanced features such as transaction support are not. Therefore, products like Oracle and SQL Server are not cost-effective, especially when compared to freely downloadable products like MySQL and PostgreSQL. Access is clearly unsuitable, since it is designed for ease of use for ordinary office workers, and is not capable of handling more than a few simultaneous users. Since MySQL offers the ability to handle many users at once, support for a wide variety of platforms, high stability and speed, standard SQL syntax, and optional graphical configuration, it is the best choice for the improved intranet.

Operating System

Windows

Advantages

- Windows is Solectron's current standard operating system.
- Familiarity: the BLS team is comfortable setting up and administering Windows. So are many other IT professionals, including Solectron's existing staff.
- Web server software comes as part of the operating system.
- Graphical interface eases the learning curve for applications.

Disadvantages

- Cannot be customized beyond what Microsoft or third-party add-ons allow, as it is proprietary.

- Less stable and secure. The vast majority of viruses target Windows, so an excellent antivirus package will be a necessity, and an additional cost. Operating system patches are only released at the discretion of the vendor, often only for very prominent security issues.
- Initial cost is steep compared to other solutions.
- Client access licenses must also be purchased, thereby increasing cost.
- Maintenance cost is also comparatively steep, since Solectron may become locked into an upgrade cycle. For example, Windows NT4 Server is not being considered, because Microsoft is discontinuing support patches for it in less than one year.²³

Linux and BSD

Advantages

- Globally deployed as a server platform, especially for web and other Internet servers.
- Many distributions are freely downloadable at no cost, with varying license terms.
- No client access licenses are needed.
- More stable and secure. Linux is known for exceptional stability and security, BSD even more so. Even minor security patches are released quickly. The OS architecture foils typical virus tactics, so there are very few viruses that can damage Linux or BSD.
- Lower system overhead for better performance, especially on older hardware.
- All needed software for web and database servers and software development is included or freely downloadable. Interoperates smoothly with Windows networks via Samba.
- May be freely configured and customized in-house, including modifying the source code.
- Large community following, and growing quickly – over twenty million worldwide, including many IT professionals. BSD especially is a favourite of ISPs and system administrators.
- Many distributions are also available as commercial or retail boxed software for a charge, in desktop and server varieties, which typically include (1) paper manuals, (2) non-free applications and vendor-enhanced functionality, and (3) technical support from the vendor.

Disadvantages

- Not as user friendly as Windows, although usability is increasing constantly.
- Not Solectron's current standard platform, so migration would be necessary. However, if the current web server is a Windows NT4 machine, this would be a perfect opportunity to migrate, since Microsoft will soon be discontinuing support.
- Not as familiar to the BLS team as Windows.

Linux and BSD Primer

A distribution is a collection of software packages that make up a Linux or BSD system. On top of the *de facto* standard core packages, the distribution maker selects extra packages, adds enhancements, and in some cases, services such as limited technical support. The maker may choose to charge a fee for the distribution because of these enhancements and services, or maintain multiple versions: a full version for a fee, or the basic downloadable version, with minimal enhancements. Some offer further services such as additional applications, technical support, and training, for fees. Solectron does not need anything other than the freely downloadable standard tools to run the systems described in this proposal.

²³ See <http://www.microsoft.com/ntserver/ProductInfo/Availability/Retiring.asp> for more information.

The next consideration is packaging formats, because the format used to package the software determines the infrastructure used to manage the machine's software. The two most common formats are RPM (Red Hat Package Management) and DEB (Debian), although others exist, such as TGZ, or installing from source code. Right now, it is widely agreed in the Linux community that the DEB packaging system is superior to RPM, but that the apt4rpm utility may narrow the gap, especially from the administrative perspective. While distribution makers like Mandrake, Red Hat, and SuSE offer commercial services and support as well as brand recognition, they use RPM packaging, which simply is not as slick as DEB is.

There is only one Debian core, on which distributions such as Libranet and Xandros are based, but RPM-based distributions have much more internal variation, so RPMs must be compiled for each distribution and version thereof. Finding the correct RPM to successfully install an unusual application in an obscure distribution may only be the beginning of the ordeal. Standard RPM management tools are less than competent at dealing with dependency or database problems,²⁴ so updating or adding packages can be a gamble. If a package depends on a different version of a core system package like glibc, part or all of the system may break if the software is forcibly installed. RPM distributions are working to lessen these problems, but they have not yet reached the level of elegance achieved by Debian, source-based distributions, or FreeBSD, which allows administrators to either use TGZ packages or compile from source. Version upgrades to RPM-based distributions are possible in theory, but they tend to be tricky, and may not work at all, especially for major upgrades (e.g. Red Hat 6.1 to 8.1). By contrast, install Debian, FreeBSD, or a source-based distribution once and it can be upgraded forever after, so long as the package sources exist. Apt4rpm attempts to bring these benefits to RPM-based distributions, so the availability of an apt4rpm repository should be a factor when choosing an RPM-based distribution.

BSD and Debian-based distributions are especially suitable for servers, since they are unusually stable and secure, and are more thoroughly tested before release than most desktop Linux distributions. While their learning curves for both installation and use are steeper, the reward is infinite automatic updates via a simple cron job. This if nothing else should make them worthy of serious consideration over their rivals: reliability is paramount for a server, whose care and feeding is a specialized task, no matter how usable the OS is. FreeBSD in particular should be considered, as it has a large and active community, is known for being friendlier to novices, and is rich in documentation.²⁵ Another advantage is that the BSD license would allow Solectron to develop modifications to the OS and sell them. The GPL

²⁴ Most packages depend on other packages being installed to function properly. These dependencies can have dependencies, and so on. RPM keeps a database of installed packages, but it does not recognize packages installed from source, etc. Therefore the database can become inaccurate, causing problems when managing packages related to those non-RPM packages.

²⁵ Please see the Linux and BSD section of the Recommended Reading appendix for links to some overview articles.

forbids this practice: modifications to GPL software must be released under the GPL license, so that they too may be freely distributed and modified, in a practice known as copyleft. However, if Solectron prefers to have commercial support available from the system vendor, then Red Hat Linux may be the most suitable choice, since it has a very strong support network and there are many apt4rpm repositories providing packages for it. SuSE also provides commercial support and maintains an apt4rpm repository, but does not have as strong a North American presence.

Recommended Implementation

Based on the requirements and research presented above, BLS believes that this recommended implementation will provide the most featureful yet cost-effective platform possible for Solectron's improved intranet.

Hardware	Existing webserver
Operating System	Debian GNU/Linux
Database	MySQL
Programming Language	PHP
Content Management System	Tiki Wiki

This platform will be inexpensive yet fast and powerful. Debian is an excellent operating system for servers for the reasons discussed just above. MySQL is a very popular database, which is in active development and improving constantly. Similarly, PHP is wildly popular scripting language because it is powerful, fast, easy to learn, free, and open-source. Tiki Wiki has nearly all the required CMS features out of the box, and the few that are missing can be developed for it easily. It is in active development and has a vibrant community, so new features and new extensions are developed frequently. Finally, the platform's open-source nature means that Solectron will be able to modify it freely as their needs change. BLS is more than willing to accommodate Solectron's business needs with a different platform, according to their needs and budget, but our research does show this to be an excellent low-cost option.

PROJECT TEAM

Team Manager: Christopher Reed

As head of the programming department and owner of BlueLightning Studios, Chris brings over four years of website planning and programming experience, and almost twenty years of experience in computers. He is a seasoned programmer who has provided solutions in every major web-based programming language, from ASP to Cold Fusion to PHP. Chris also brings experience in developing and managing personnel: during previous projects, he has been in charge of handling any issues that arise and keeping the team focused, ensuring that the project comes in on or before the deadline. With Chris at the helm, you can be sure that the project will be up to the professional standards that Solelectron is entitled to.

Web Designer: Jason Coleman

Jason is the head of web design at BlueLightning Studios, and brings five years of experience in designing webpages for businesses and individuals. He has been with BLS for over two years. Jason is an excellent template designer and budding graphic artist, and his sure hand with graphics software and HTML makes his webpages into works of art. His skills in layout and graphic design will make the intranet a joy to use and a joy to view.

Web Developer: David Latremouille

Our senior developer brings a wealth of experience, and specializes in web applications, database construction and maintenance, and advanced ASP functions. He is one of the three senior programmers who developed the new intranet at Compaq's Nepean Call Center. Before that, David was the senior support specialist for developer applications (including FrontPage, Access, and Microsoft Visual Studio), for Microsoft's internal corporate helpdesk. David's programming and troubleshooting expertise will enable him to craft solutions out of nothing and quickly resolve whatever issues may arise.

Web Developer: Natalya Boriko

Our latest addition, Natalya has worked extensively with many different platforms and undertakings of varying scope, from websites for small businesses to large government projects. Natalya also brings a legendary work ethic to the team: she will not stop working on the task at hand until it is complete, and does not countenance half-measures. Another asset that Natalya provides is her

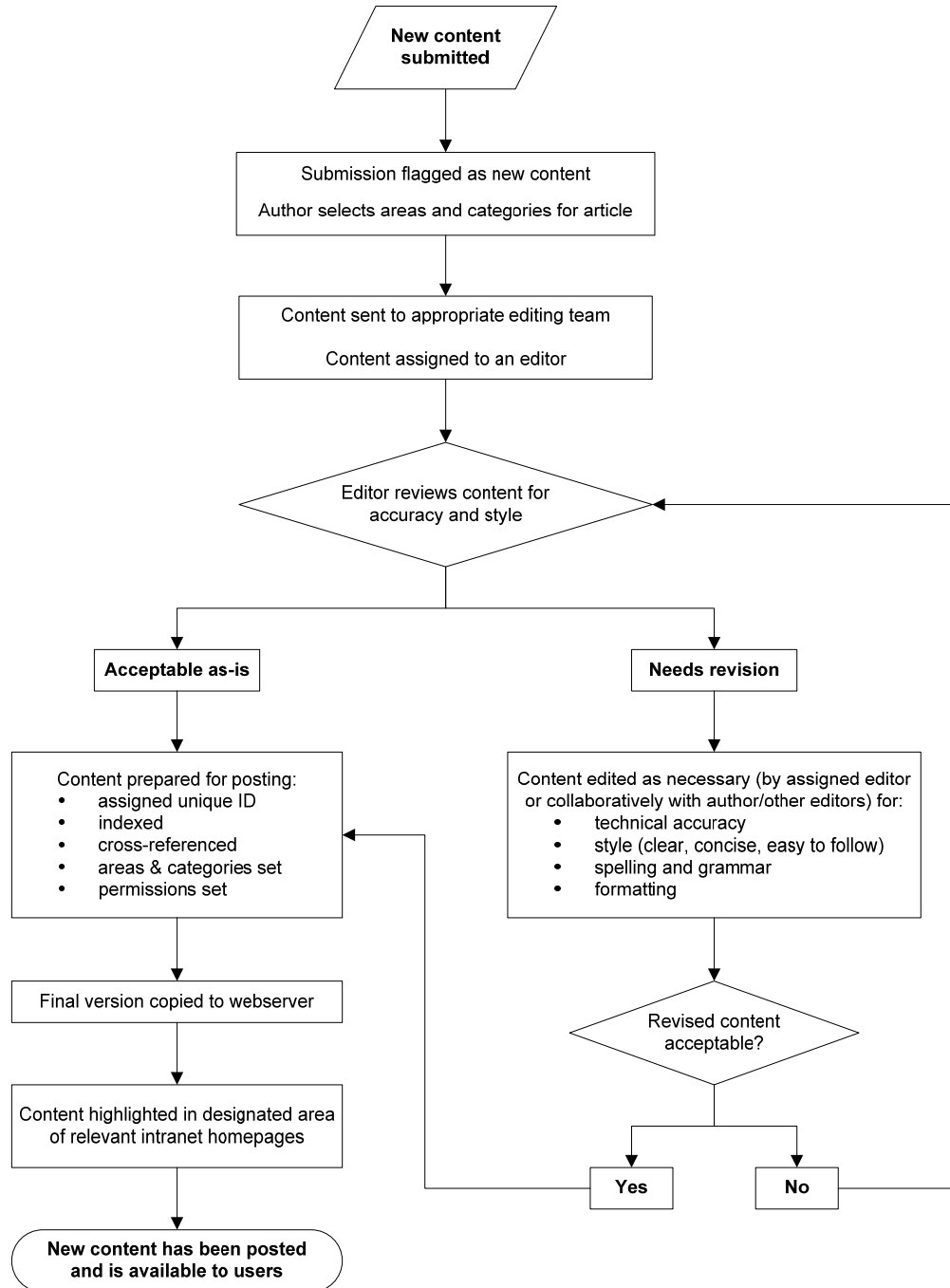
proficiency in developing web-based applications in both Microsoft and non-Microsoft platforms, which is a rare qualification. Natalyia's proficiency and exceptional work ethic will guarantee that the project stays on schedule throughout its lifetime.

Design Consultant: Bronwyn Boltwood

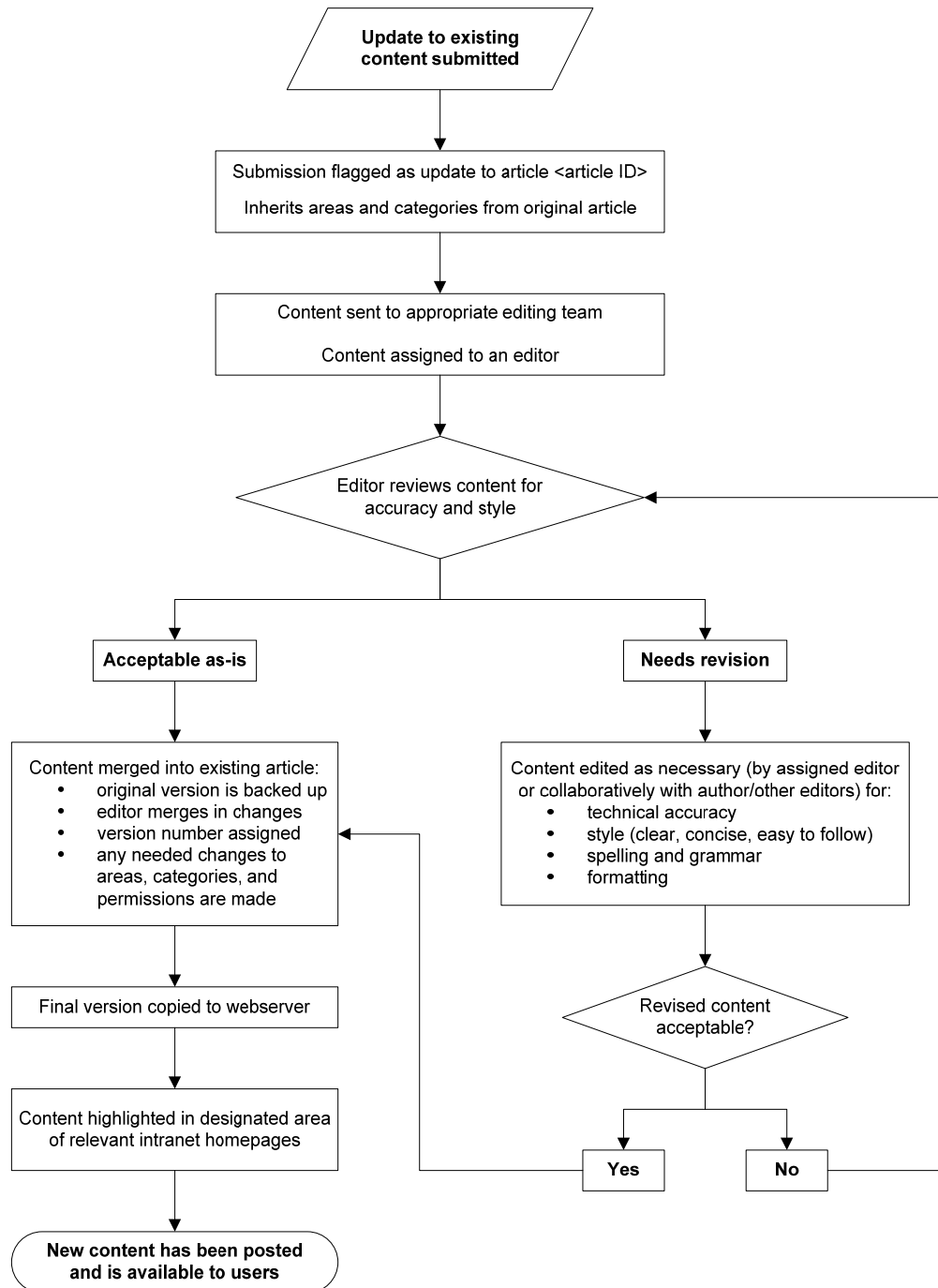
With nearly ten years of experience in computers, writing and design, Bronwyn contributes an essential set of skills. Her role in this project encompasses business needs analysis, information architecture, process design, usability consulting, user interface design, and technical writing. A natural fountain of ideas trained in critical thinking and logic, Bronwyn fuses these talents to analyze and optimize systems for efficiency and ease of use. Her vision, research, and planning add tremendous value to the team.

APPENDIX: WORKFLOW

New Content



Updates to Existing Content



APPENDIX: SAMPLE TASK SUMMARY

Current Tasks

New Items
1 article

To Revise
3 articles

To Approve
2 articles

To Post
0 articles

Harry Golden

Status	ID	Title	Author	Submitted	Due
New	001836	Memory Errors in Open Desktop	Horace Quirt	14-Mar-03	19-Mar-03
Revising	000836	Common Modem Init Strings	Melanie O'Brien	12-Mar-03	17-Mar-03
Revising	000467	Monitor Dispatch Instructions	Beverly Duggleston	14-Mar-03	19-Mar-03
Revising	000928	Internet Explorer Fails to Start	Albert Todd	15-Mar-03	20-Mar-03
Integrating	000172	Optical Drives Troubleshooting Flowchart	Jeff Nichols	13-Mar-03	18-Mar-03
Integrating	001047	Hard Drive Failure Codes	Eric Ewing	16-Mar-03	21-Mar-03

Desktop Team

Status	ID	Title	Author	Submitted	Due	Editor
New	001834	Common Errors Installing Windows XP Home	George Gilmour	13-Mar-03	18-Mar-03	Jennifer Mallard
New	001836	Memory Errors in Open Desktop	Horace Quirt	14-Mar-03	19-Mar-03	Harry Golden
New	001840	IFHELP.SYS Not Found	Paul MacMillan	16-Mar-03	21-Mar-03	Diana Carver
New	001841	No Sound Troubleshooting Flowchart	Fern Cooper	16-Mar-03	21-Mar-03	Diana Carver
Rejected	001842	Keyboard Errors on the EN2816	Doris Ironside	12-Mar-03	17-Mar-03	Diana Carver
Revising	000836	Common Modem Init Strings	Melanie O'Brien	12-Mar-03	17-Mar-03	Harry Golden
Revising	000465	Turning off the MS Office Assistant	Olivia Allen	13-Mar-03	18-Mar-03	Diana Carver
Revising	000467	Monitor Dispatch Instructions	Beverly Duggleston	14-Mar-03	19-Mar-03	Harry Golden
Revising	000373	Reinstalling Drivers from Driverbase	Larry Higgins	15-Mar-03	20-Mar-03	Jennifer Mallard
Revising	000928	Internet Explorer Fails to Start	Albert Todd	15-Mar-03	20-Mar-03	Harry Golden
Revising	000179	Boot Failure Troubleshooting Flowchart	Tamara Reid	16-Mar-03	21-Mar-03	Diana Carver
Revising	001038	ISP Contact Numbers	Sebastian Johnston	17-Mar-03	22-Mar-03	Jennifer Mallard
Integrating	000172	Optical Drives Troubleshooting Flowchart	Jeff Nichols	13-Mar-03	18-Mar-03	Harry Golden
Integrating	000637	Scandisk Runs on Every Boot	Neil Forester	14-Mar-03	19-Mar-03	Diana Carver
Integrating	001837	Managing Outlook Profiles and Services	Victor Sawyer	15-Mar-03	20-Mar-03	Diana Carver
Integrating	001047	Hard Drive Failure Codes	Eric Ewing	16-Mar-03	21-Mar-03	Harry Golden
Approved	001264	Emergency Windows Product Keys	Isabel Livingstone	13-Mar-03	18-Mar-03	Jennifer Mallard
Approved	000946	Dial Up Networking Errors	Kirsten Parker	15-Mar-03	20-Mar-03	Diana Carver
Approved	001362	Using MSCONFIG and the Startup Item List	Colin Knox	16-Mar-03	21-Mar-03	Diana Carver
Approved	000826	Disabling Onboard Video	Rowan Brown	17-Mar-03	22-Mar-03	Jennifer Mallard

APPENDIX: TEXT MARKUP VS. HTML

To clearly demonstrate the difference between a text markup system and HTML, we created this sample article in a TWiki Sandbox topic, at <http://twiki.org/cgi-bin/view/Sandbox/SandBox65>. Please follow the link to see it rendered in your web browser. Then compare the sources. The article was written in the TWiki shorthand below, and then processed by a template, which generated the HTML source.

TWiki Shorthand

---+Example Text Markup for TWiki

%TOC%

I'm using this entry as an example of TWiki's text markup in a proposal explaining the benefits of using a wiki system for their intranet. They are a technical support provider, and therefore need an extensive knowledgebase. TWiki is great for knowledgebases, especially technical ones that need constant updating. This is borne out by Main.TWikiSuccessStoryOfTakeFive and Main.TWikiSuccessStoryOfWindRiver.

---++Links

#BackToLinks

Linking via WikiWords or free links is one of the core features of all wiki systems. TWiki's text markup system is simple enough that I can just type normally in Metapad, my favourite text editor. The full help page for TWiki Shorthand is at the [[<http://twiki.org/cgi-bin/view/TWiki/TextFormattingRules> TWiki Text Formatting Rules page]]. When posting on TWiki's site, I can just put TWiki.TextFormattingRules and TWiki will make it a link for me. I can drop in an URL for any common protocol such as http/https, ftp, gopher, news, file, or telnet and have it work, e.g. <http://google.ca>. TWiki is a good neighbour and automatically spamtraps email addresses against harvester programs too: <no>newbie@sucker.com is changed to <mailto:newbie@sucker.com>. I can also write a <no>WikiWord and force it *not* to link. TWiki supports [[#MyAnchor][anchors]] but your anchors must be WikiWords.

---++Bold and Italic Text

TWiki supports ***bold*** and *_italic_* text, and even does it in the way that UNIX/Freenet text mail system veterans like myself would expect. There is also __bold italic text__ and `=fixed-width formatting=` (even in `==bold==`) available.

---++Numbered Lists

I can make a numbered list like so, with bulleted sublists. If I use TWiki regularly, I should set my text editor's tab stops to 3 spaces, because list levels are indented in multiples of three.

1. list item 1
2. list item 2
3. list item 3
 - * sublist item 1
 - * sublist item 2
 - * sub-sublist item 1

* sub-sublist item 2

---++Headings

As you can see, Twiki supports headings, six levels of them to be precise, and will even make a table of contents for you. You can choose to leave a heading out of the table of contents:

---++++!!Sample Heading

---++Multimedia

Twiki supports inline images like this one

<http://twiki.org/p/pub/Twiki/PreviewBackground/preview2bg.gif>

and also attachments. Attachments are under version control, just like the pages they belong to.

---++Tables

Twiki's markup supports tables, which is an extremely nice feature. Not only that, but they're much much easier to mark up than HTML tables. Compare the sources for this table:

```
| *Twiki Table* |||
| normal left-aligned cell | center-aligned cell | right-aligned cell |
| There is colspan but no rowspan. Nothing's perfect. Now I'm adding space
so you can see the different alignments. |||
```

---+++Twiki Shorthand Version

```
<verbatim>
| *Twiki Table* |||
| normal left-aligned cell | center-aligned cell | right-aligned cell |
| There is colspan but no rowspan. Nothing's perfect. Now I'm adding space
so you can see the different alignments. |||
</verbatim>
```

---+++HTML Version

```
<verbatim>
<table border="1" cellspacing="1" cellpadding="0">
<tr><th colspan="3" bgcolor="#99CCCC"> <strong>Twiki Table</strong>
</th></tr>
<tr><td bgcolor="#FFFFFF"> normal left-aligned cell </td><td align="center"
bgcolor="#FFFFFF"> center-aligned cell </td><td align="right"
bgcolor="#FFFFFF"> right-aligned cell </td></tr>
<tr><td colspan="3" bgcolor="#FFFFCC"> There is colspan but no rowspan.
Nothing's
perfect. Now I'm adding space so you can see the different alignments.
</td></tr>
</table>
</verbatim>
```

```
#MyAnchor
[[#BackToLinks][Go back to Links]]
```

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title> Twiki . Sandbox . SandBox65  </title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <link rel="alternate" type="application/rss+xml" title="RSS Feed"
href="http://twiki.org/cgi-
bin/view/Sandbox/WebRss?skin=rss&contenttype=text/xml" />
  <base href="http://twiki.org/cgi-bin/view/Sandbox/SandBox65" />
</head>
<body bgcolor="#ffffff">
<a name="PageTop"></a>
<form name="main" action="/cgi-bin/view/Sandbox/SandBox65">
<table width="100%" border="0" cellpadding="3" cellspacing="0">
  <tr>
    <td bgcolor="#D0E0DD" rowspan="2" valign="top" width="1%">
      <a href="http://Twiki.org/"></a>
    </td><td>
      <a href="http://twiki.org/cgi-bin/view/Main/WebHome">Twiki</a>
      &gt; <a href="http://twiki.org/cgi-bin/view/Sandbox/WebHome">Sandbox</a>
      &gt;
      <font size="+1"><b>SandBox65</b> </font>
    </td><td align="right">
      <font size="-2">Twiki webs: <br />
      <a href="/cgi-bin/view/Main/WebHome">Main</a> | <a href="/cgi-
bin/view/TWiki/WebHome">Twiki</a> | <a href="/cgi-
bin/view/Codev/WebHome">Codev</a> | <a href="/cgi-
bin/view/Plugins/WebHome">Plugins</a> | <a href="/cgi-
bin/view/Support/WebHome">Support</a> | <a href="/cgi-
bin/view/Sandbox/WebHome">Sandbox</a> </font>
    </td>
  </tr>
  <tr bgcolor="#D0E0DD">
    <td colspan="2">
      Sandbox . { <a href="/cgi-bin/view/Sandbox/WebChanges">Changes</a> | <a
href="/cgi-bin/view/Sandbox/WebIndex">Index</a> | <a href="/cgi-
bin/view/Sandbox/WebSearch">Search</a> | Go <input type="text" name="topic"
size="16" /> }
    </td>
  </tr>
</table>
</form>
<h1><a name="Example_Text_Markup_for_TWiki"> Example Text Markup for Twiki
</a></h1>
<p />
<p />
<ul>
<li> <a href="/cgi-
bin/view/Sandbox/SandBox65#Example_Text_Markup_for_TWiki">Example Text Markup
for Twiki</a>
<ul>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Links">Links</a>
</li>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Bold_and_Italic_Text">Bold and
Italic Text</a>
</li>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Numbered_Lists">Numbered
Lists</a>
</li>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Headings">Headings</a>
</li>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Multimedia">Multimedia</a>
</li>

```

```

<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Tables">Tables</a>
<ul>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#Twiki_Shorthand_Version">Twiki
Shorthand Version</a>
</li>
<li> <a href="/cgi-bin/view/Sandbox/SandBox65#HTML_Version">HTML Version</a>
</li>
</ul>
</li>
</ul>
</li>
</ul>
<p />

```

I'm using this entry as an example of Twiki's text markup in a proposal explaining the benefits of using a wiki system for their intranet. They are a technical support provider, and therefore need an extensive knowledgebase. Twiki is great for knowledgebases, especially technical ones that need constant updating. This is borne out by [TwikiSuccessStoryOfTakeFive](/cgi-bin/view/Main/TWikiSuccessStoryOfTakeFive) and [TwikiSuccessStoryOfWindRiver](/cgi-bin/view/Main/TWikiSuccessStoryOfWindRiver).

```

<p />
<h2><a name="Links"> Links </a></h2>
<p />

```

[BackToLinks](#)

Linking via [WikiWords](/cgi-bin/view/Sandbox/WikiWord) or free links is one of the core features of all wiki systems. Twiki's text markup system is simple enough that I can just type normally in Metapad, my favourite text editor. The full help page for Twiki Shorthand is at the [Twiki Text Formatting Rules](http://twiki.org/cgi-bin/view/TWiki/TextFormattingRules) page. When posting on Twiki's site, I can just put [TextFormattingRules](/cgi-bin/view/TWiki/TextFormattingRules) and Twiki will make it a link for me. I can drop in an URL for any common protocol such as http/https, ftp, gopher, news, file, or telnet and have it work, e.g. <http://google.ca>. Twiki is a good neighbour and automatically spamtraps email addresses against harvester programs too: newbie@sucker.com is changed to newbie@suckerSTOPSPAM.com. I can also write a WikiWord and force it **not** to link. Twiki supports [anchors](/cgi-bin/view/Sandbox/SandBox65#MyAnchor) but your anchors must be [WikiWords](/cgi-bin/view/Sandbox/WikiWord).

```

<p />
<h2><a name="Bold_and_Italic_Text"> Bold and Italic Text </a></h2>
<p />

```

Twiki supports **bold** and *italic* text, and even does it in the way that UNIX/Freenet text mail system veterans like myself would expect. There is also ***bold italic text*** and `fixed-width formatting` (even in `<b>bold</b>`) available.

```

<p />
<h2><a name="Numbered_Lists"> Numbered Lists </a></h2>
<p />

```

I can make a numbered list like so, with bulleted sublists. If I use Twiki regularly, I should set my text editor's tab stops to 3 spaces, because list levels are indented in multiples of three.

```

<p />
<ol>
<li> list item 1
</li>
<li> list item 2
</li>
<li> list item 3
</li>
</ol>

```

```

<li> sublist item 1
</li>
<li> sublist item 2
<ul>
<li> sub-sublist item 1
</li>
<li> sub-sublist item 2
</li>
</ul>
</li>
</ul>
</li>
</ol>
<p />
<h2><a name="Headings"> Headings </a></h2>
<p />
As you can see, Twiki supports headings, six levels of them to be precise,
and will even make a table of contents for you. You can choose to leave a
heading out of the table of contents:
<p />
<h4><a name="Sample_Heading"> Sample Heading </a></h4>
<p />
<h2><a name="Multimedia"> Multimedia </a></h2>
<p />
Twiki supports inline images like this one
<p />

<p />
and also attachments. Attachments are under version control, just like the
pages they belong to.
<p />
<h2><a name="Tables"> Tables </a></h2>
<p />
Twiki's markup supports tables, which is an extremely nice feature. Not only
that, but they're much much easier to mark up than HTML tables. Compare the
sources for this table:
<p />
<table border="1" cellspacing="1" cellpadding="0">
<tr><th colspan="3" bgcolor="#99CCCC"> <strong>Twiki Table</strong>
</th></tr>
<tr><td bgcolor="#FFFFFF"> normal left-aligned cell </td><td align="center"
bgcolor="#FFFFFF"> center-aligned cell </td><td align="right"
bgcolor="#FFFFFF"> right-aligned cell </td></tr>
<tr><td colspan="3" bgcolor="#FFFFCC"> There is colspan but no rowspan.
Nothing's perfect. Now I'm adding space so you can see the different
alignments. </td></tr>
</table>
<p />
<h3><a name="Twiki_Shorthand_Version"> Twiki Shorthand Version </a></h3>
<p />
<pre>
| *Twiki Table* |||
| normal left-aligned cell | center-aligned cell | right-aligned cell |
| There is colspan but no rowspan. Nothing's perfect. Now I'm adding space
so you can see the different alignments. |||
</pre>
<p />
<h3><a name="HTML_Version"> </a> HTML Version </h3>
<p />
<pre>
<table border="1" cellspacing="1" cellpadding="0">
<tr><th colspan="3" bgcolor="#99CCCC"> <strong>Twiki
Table</strong> </th></tr>

```

```

</tr><td bgcolor="#FFFFFF"> normal left-aligned cell </td><td
align="center"
bgcolor="#FFFFFF"> center-aligned cell </td><td align="right"
bgcolor="#FFFFFF"> right-aligned cell </td></tr>
<tr><td colspan="3" bgcolor="#FFFFCC"> There is colspan but no
rowspan. Nothing's
perfect. Now I'm adding space so you can see the different alignments.
</td></tr>
</table>
</pre>
<p />
<a name="MyAnchor"></a>
<a href="/cgi-bin/view/Sandbox/SandBox65#BackToLinks">Go back to Links</a>
<p />
<hr />
<p />
Polite request: please leave the Twiki Shorthand display above intact, as I
am using it to demonstrate the markup. It's an appendix in a proposal to
convince the company to use Twiki for their tech support KB. I'm sure you
all want us to start using Twiki at work!
<p />
-- <a href="/cgi-bin/view/Main/BronwynBoltwood">BronwynBoltwood</a> - 23 Feb
2003
<p />
<p />
<table width="100%" border="0" cellpadding="3" cellspacing="0">
  <tr bgcolor="#D0E0DD">
    <td valign="top">
      Topic <b>SandBox65</b> . { <a href="/cgi-
bin/edit/Sandbox/SandBox65?t=1046186420"><b>Edit</b></a>
      | <a href="/cgi-bin/attach/Sandbox/SandBox65">Attach</a>
      | <a href="/cgi-
bin/search/Sandbox/SearchResult?scope=text&regex=on&search=Sand%20*Bo
x%20*65%5B%5EA-Za-z%5D">Ref-By</a>
      | <a href="/cgi-bin/view/Sandbox/SandBox65?skin=print">Printable</a>
      | <a href="/cgi-bin/rdiff/Sandbox/SandBox65">Diffs</a> | r1.12 | <a
href="/cgi-bin/rdiff/Sandbox/SandBox65?rev1=1.12&rev2=1.11">&gt;</a> | <a
href="/cgi-bin/view/Sandbox/SandBox65?rev=1.11">r1.11</a> | <a href="/cgi-
bin/rdiff/Sandbox/SandBox65?rev1=1.11&rev2=1.10">&gt;</a> | <a
href="/cgi-bin/view/Sandbox/SandBox65?rev=1.10">r1.10</a>
      | <a href="/cgi-
bin/oops/Sandbox/SandBox65?template=oopsmore&param1=1.12&param2=1.12"
>More</a>
    }
  </td>
</tr>
</table>
<table width="100%" border="0" cellpadding="3" cellspacing="0">
  <tr>
    <td valign="top">
      Revision r1.12 - 25 Feb 2003 - 15:20 GMT - <a href="/cgi-
bin/view/Main/BronwynBoltwood">BronwynBoltwood</a> <br />Parents: <a
href="/cgi-bin/view/Sandbox/WebHome">WebHome</a>
    </td>
    <td width="40%" valign="top">
      <font size="-2">Copyright &copy; 1999-2003 by the contributing authors.
      All material on this collaboration platform is the property of the
      contributing authors. <br />
      Ideas, requests, problems regarding Twiki? <a
href="mailto:Peter@Thoeny.com?subject=Twiki&#32;Feedback&#32;on&#32;Sandbox.S
andBox65">Send</a> feedback
      <!-- NOTE: The following image should not go into Beta!! -->

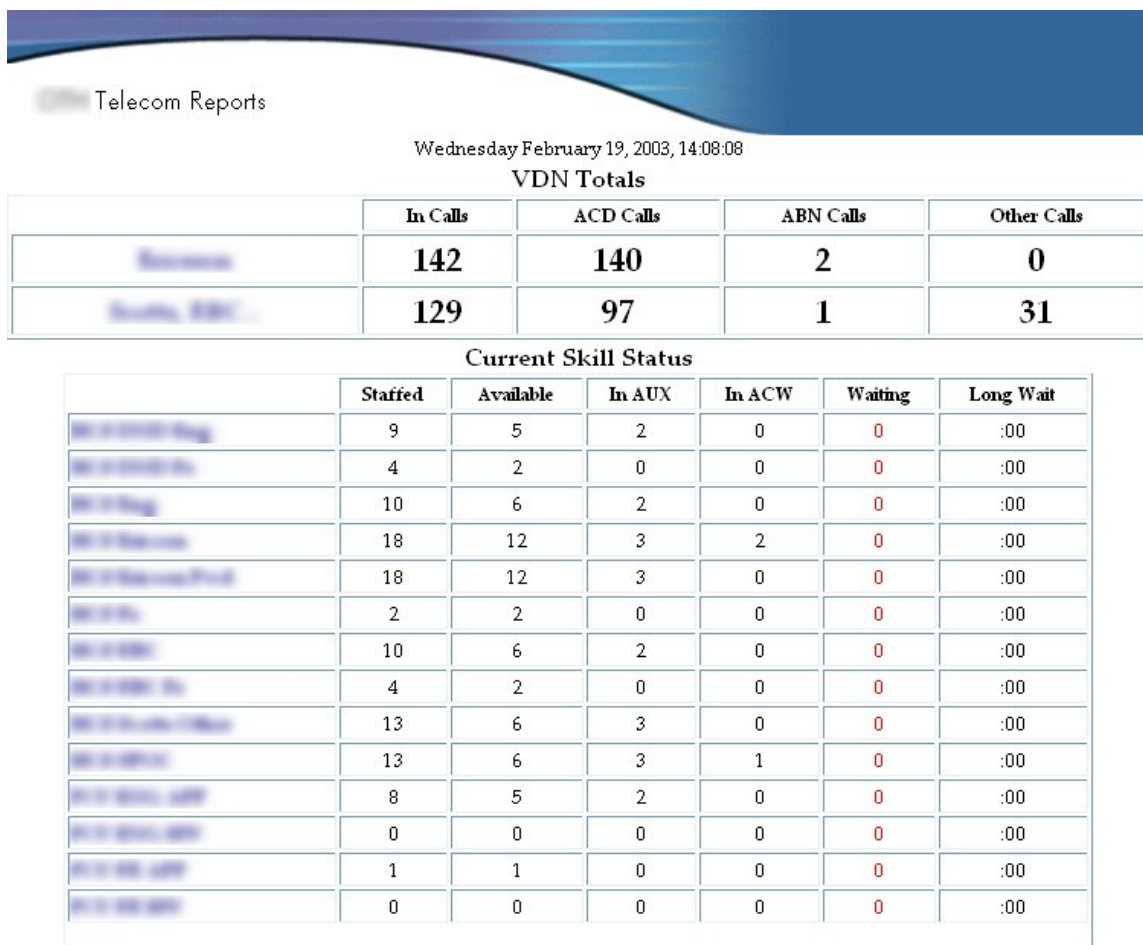
```

```
        <a href="/cgi-
bin/viewauth/Sandbox/SandBox65">.</a>
        </font>
    </td>
</tr>
<tr><td colspan="2"> </td></tr>
</table>
<a name="PageBottom"></a>
</body>
</html>
```

APPENDIX: SAMPLE LIVE MONITORS

These two screenshots should demonstrate what we mean by live monitors: web pages that fetch and display statistics in real-time, so that they can be easily checked without special software. These particular samples use data from the phone switch to keep all members of the team apprised of how the call volume is flowing and if they have exceeded a reasonable amount of time when using an aux code or wrapping up a call.

Sample Queue Monitor



Telecom Reports

Wednesday February 19, 2003, 14:08:08

VDN Totals

	In Calls	ACD Calls	ABN Calls	Other Calls
Service	142	140	2	0
Accts. Billed	129	97	1	31

Current Skill Status

	Staffed	Available	In AUX	In ACW	Waiting	Long Wait
AC 10000-Reg	9	5	2	0	0	:00
AC 10000-Reg	4	2	0	0	0	:00
AC 10000-Reg	10	6	2	0	0	:00
AC 10000-Reg	18	12	3	2	0	:00
AC 10000-Reg	18	12	3	0	0	:00
AC 10000-Reg	2	2	0	0	0	:00
AC 10000-Reg	10	6	2	0	0	:00
AC 10000-Reg	4	2	0	0	0	:00
AC 10000-Reg	13	6	3	0	0	:00
AC 10000-Reg	13	6	3	1	0	:00
AC 10000-Reg	8	5	2	0	0	:00
AC 10000-Reg	0	0	0	0	0	:00
AC 10000-Reg	1	1	0	0	0	:00
AC 10000-Reg	0	0	0	0	0	:00

Sample Team Monitor

Telecom Reports						
Group Report						
As of Wednesday February 19, 2003, 14:12:07						
Agent	Ext.	Mode	Aux Reason	Dur.	Dir.	VDN
AGENT 8662	8662	AUX	Queue Review	:05:12		
AGENT 8450	8450	AVAIL		:06:05		
AGENT 8448	8448	AVAIL		:14:51		
AGENT 8828	8828	AVAIL		:15:09		
AGENT 3340	3340	AVAIL		:07:04		
AGENT 8717	8717	ACD		:00:06	IN	
AGENT 8829	8829	AVAIL		:06:30		
AGENT 8711	8711	AUX	Queue Review	:29:49		
AGENT 8595	8595	AVAIL		:09:06		
AGENT 8567	8567	AUX	Callbacks	:00:49	OUT	
AGENT 8709	8709	ACW		:01:16		
AGENT 8505	8505	AUX	Scheduled Breaks	:00:55		
AGENT 8141	8141	AVAIL		:06:01		
AGENT 8732	8732	ACD		:05:07	IN	
AGENT 8713	8713	AVAIL		:05:31		
AGENT 8525	8525	AVAIL		:10:16		
AGENT 8718	8718	AVAIL		:01:38		
AGENT 8714	8714	AVAIL		:23:57		
AGENT 8712	8712	ACD		:01:28	IN	
AGENT 8668	8668	AUX	Queue Review	:00:24		

APPENDIX: SAMPLE EXCEL REPORT

N.B.: the upper rightmost number for each statistic is the average of the team averages. The lower rightmost number is the average of the technician averages. The bolded and italicized number in the Team column is the average of those two averages.

		Adam	Brenda	Charlie	Darlene	Team	
Talk time	<i>Mon</i>	05:56	05:59		15:36	<i>09:10</i>	
	<i>Tue</i>	05:31	07:14		11:48	<i>08:11</i>	
	<i>Wed</i>	04:37	06:45	11:45		<i>07:42</i>	
	<i>Thu</i>	03:21	07:05	13:24		<i>07:57</i>	
	<i>Fri</i>	06:15	05:26	09:56	10:37	<i>08:04</i>	
	<i>Sat</i>			10:23	13:29	<i>11:56</i>	
	<i>Sun</i>			12:34	09:07	<i>10:51</i>	09:07
	<i>Average</i>	<i>05:08</i>	<i>06:30</i>	<i>11:36</i>	<i>12:07</i>	<i>08:59</i>	08:50
Wrap time	<i>Mon</i>	01:34	01:36		00:34	<i>01:15</i>	
	<i>Tue</i>	01:48	02:48		00:15	<i>01:37</i>	
	<i>Wed</i>	01:03	01:35	03:48		<i>02:09</i>	
	<i>Thu</i>	00:48	01:51	02:56		<i>01:52</i>	
	<i>Fri</i>	03:25	02:07	02:16	00:21	<i>02:02</i>	
	<i>Sat</i>			02:42	00:24	<i>01:33</i>	
	<i>Sun</i>			03:18	00:48	<i>02:03</i>	01:47
	<i>Average</i>	<i>01:44</i>	<i>01:59</i>	<i>03:00</i>	<i>00:28</i>	<i>01:48</i>	01:48
Hold time	<i>Mon</i>	00:04	00:37		02:48	<i>01:10</i>	
	<i>Tue</i>	00:34	00:11		01:56	<i>00:54</i>	
	<i>Wed</i>	00:42	00:23	01:48		<i>00:58</i>	
	<i>Thu</i>	00:06	00:48	01:21		<i>00:45</i>	
	<i>Fri</i>	01:41	00:26	01:36	02:26	<i>01:32</i>	
	<i>Sat</i>			00:59	01:59	<i>01:29</i>	
	<i>Sun</i>			01:06	02:14	<i>01:40</i>	01:12
	<i>Average</i>	<i>00:37</i>	<i>00:29</i>	<i>01:22</i>	<i>02:17</i>	<i>01:12</i>	01:11
Aux time	<i>Mon</i>	94:12	64:34		68:45	<i>75:50</i>	
	<i>Tue</i>	87:56	62:14		72:39	<i>74:16</i>	
	<i>Wed</i>	106:05	72:35	117:12		<i>98:37</i>	
	<i>Thu</i>	73:42	103:52	103:53		<i>93:49</i>	
	<i>Fri</i>	128:43	69:45	99:51	64:31	<i>90:43</i>	
	<i>Sat</i>			81:43	67:36	<i>74:40</i>	
	<i>Sun</i>			112:34	69:56	<i>91:15</i>	85:36
	<i>Average</i>	<i>98:08</i>	<i>74:36</i>	<i>103:03</i>	<i>68:41</i>	<i>85:51</i>	86:07

Talk: average mm:ss per call for the day, then for the week

Wrap: average mm:ss per call for the day, then for the week

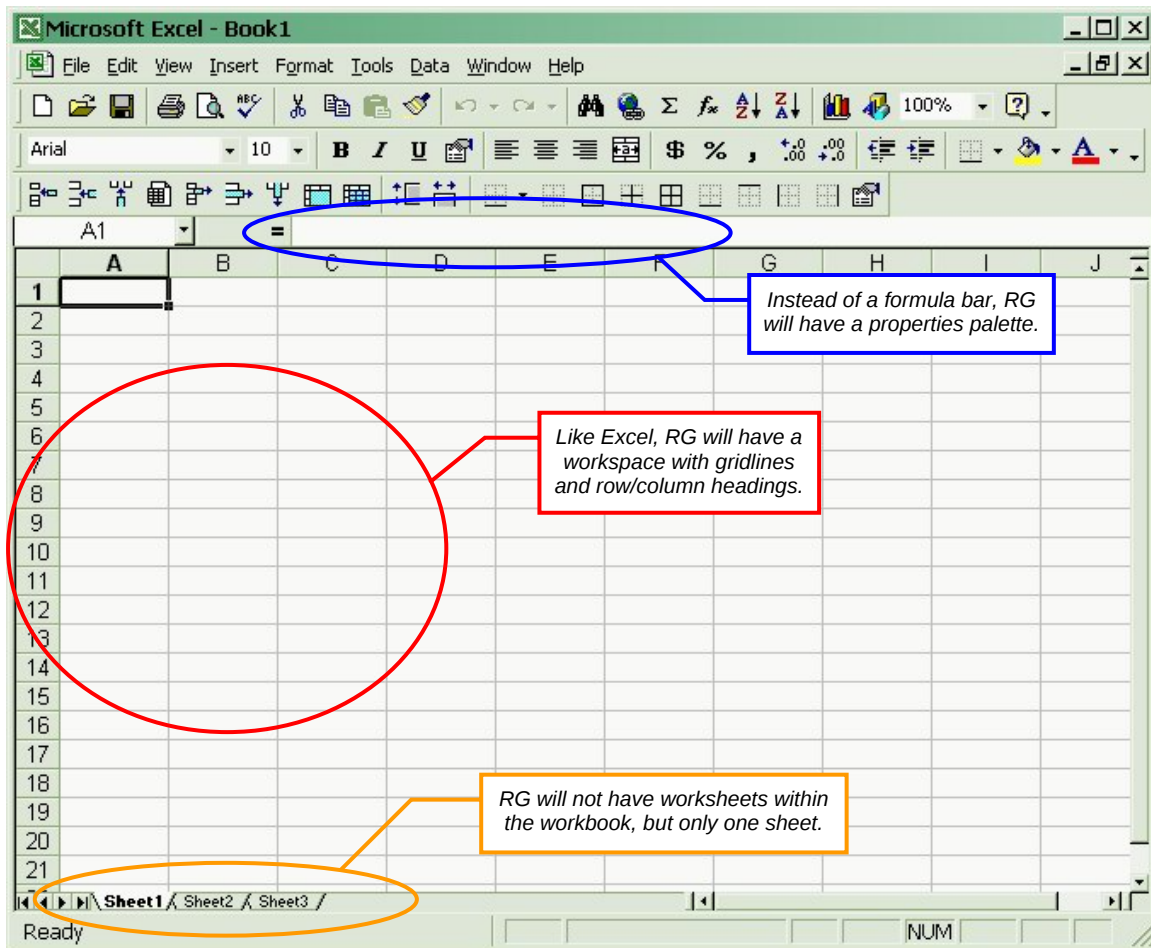
Hold: average amount of hold time used by agent, in mm:ss, for the day, then for the week

Aux: amount of aux time used by agent, in mm:ss, for the day, then daily average for the week

APPENDIX: INTERFACE IDEAS

This is a small collection of screenshots, with comments, from several different programs, as examples of how the interface for the report generator application (RG for short in the comments) should or could look and act. Comments on the application itself are in regular type, while comments on how the report generator could make use of or differ from that feature are in *italics*.

Microsoft Excel



Microsoft Visio

Microsoft Visio - [workflow.vsd:New Content]

File Edit View Insert Format Tools Shape Window Help

Type a question for help Ask a Question

Flow Normal Arial 8pt. B I U

Shapes

- Backgrounds
- Borders and Titles
- Basic Flowchart Shapes

Process Decision Document

Data Predefined Stored data process

Internal Sequential Direct data storage data process

Manual Card Paper tape input operation Preparation

Display Manual operation Preparation

Parallel Loop limit Terminator mode limit

On-page Off-page Flowchart reference reference shapes

Auto-height Dynamic Line-curve connector box connector

Control Annotation transfer

Annotations:

- Tabs for shape groups – click tab to view group's items. Note bad tab design: tabs do not look flippable although they are. *RG will have groups like Duration and Phone Statistics.*
- Shapes palette, docked. Note drag handle by palette title. Resized by dragging edge. *RG will have an items palette instead.*
- Items – drag into workspace to create blank item of that type in the document. *RG will have items like Team, Hold Time, and Service Level.*
- Search taskpane, resized and floating rather than docked.
- RG will have properties palette here, showing properties of item selected in workspace.*

Workflow Diagram:

```

graph TD
    Start([New Content]) --> Step1[Submission flagged as new content  
applicable area and category  
sent to appropriate editor  
content assigned to an editor]
    Step1 --> Step2{Editor reviews content for  
accuracy and style}
    Step2 --> Step3[Workspace. Will look like blank  
Excel sheet in RG.]
    Step2 --> Step4[Needs revision]
    Step4 --> Step5[Content edited as necessary (by assigned editor  
collaboratively with author/other editors) for:  
• technical accuracy  
• style (clear, concise, easy to follow)  
• spelling and grammar  
• formatting]
    Step5 --> Step6{Revised content  
acceptable?}
    Step6 --> Step7[Content prepared for posting:  
• assigned unique ID  
• indexed  
• cross-referenced]
    Step7 --> End([Content automatically highlighted in New Stuff])
  
```

Basic Search

Search for:

Search text:

Search Restore

Search Tips...

Other Search Options:

Search in:

Selected locations

Results should be:

Selected file types

Fast searching is currently disabled

Search options...

See also

Advanced Search

Page 1/2

Adobe Photoshop

Adobe Photoshop

File Edit Image Layer Select Filter View Window Help

team monitor2.bmp @ 100% (RGB)

0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900

As of Wednesday February 19, 2003, 14:12:07

Agent	Ext.	Mode	Aux Reason	Dur.	Dir.	VDN
AGENT1 NAME TIME	8662	AUX	Queue Review	:05:12		
AGENT2 NAME TIME	8450	AVAIL		:06:05		
AGENT3 NAME TIME	8448	AVAIL		:14:51		
AGENT4 NAME TIME	8828	AVAIL				
AGENT5 NAME TIME	3340	AVAIL				
AGENT6 NAME TIME	8717	ACD				
AGENT7 NAME TIME	8829	AVAIL				
AGENT8 NAME TIME	8711	AUX	Queue Review	:29:49		
AGENT9 NAME TIME	8595	AVAIL		:09:06		
AGENT10 NAME TIME	8567	AUX	Callbacks	:00:49	OUT	
AGENT11 NAME TIME	8709	ACW		:01:16		
AGENT12 NAME TIME	8505	AUX	Scheduled Breaks	:00:55		
AGENT13 NAME TIME	8141	AVAIL		:06:01		
AGENT14 NAME TIME	8732	ACD		:05:07	IN	
AGENT15 NAME TIME	8713	AVAIL				
AGENT16 NAME TIME	8525	AVAIL				
AGENT17 NAME TIME	8718	AVAIL				
AGENT18 NAME TIME	8714	AVAIL				
AGENT19 NAME TIME	8712	ACD			IN	
AGENT20 NAME TIME	8668	AUX	Queue Review	:00:24		

100% Doc: 1.49M/1.49M Click and drag to define cropping frame. Use Ctrl for additional options.

Navigator Info Options

Color Swatches Brushes

History Actions

Layers Channels Paths

Normal Opacity: %

Preserve Transparency

Background

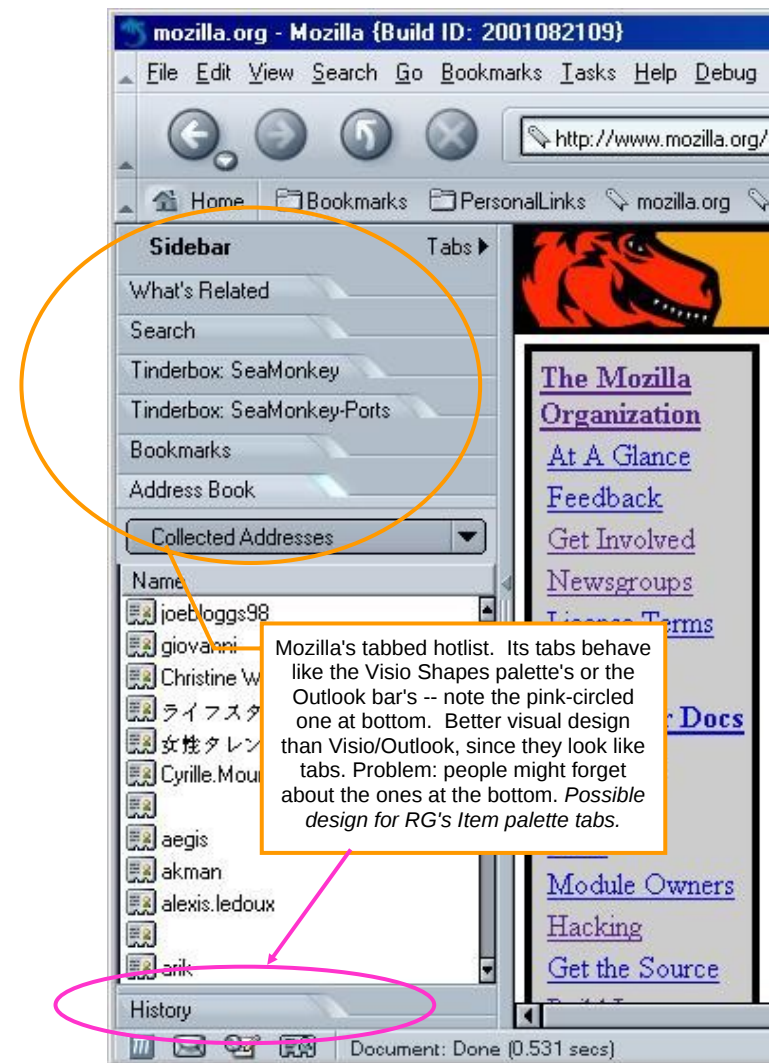
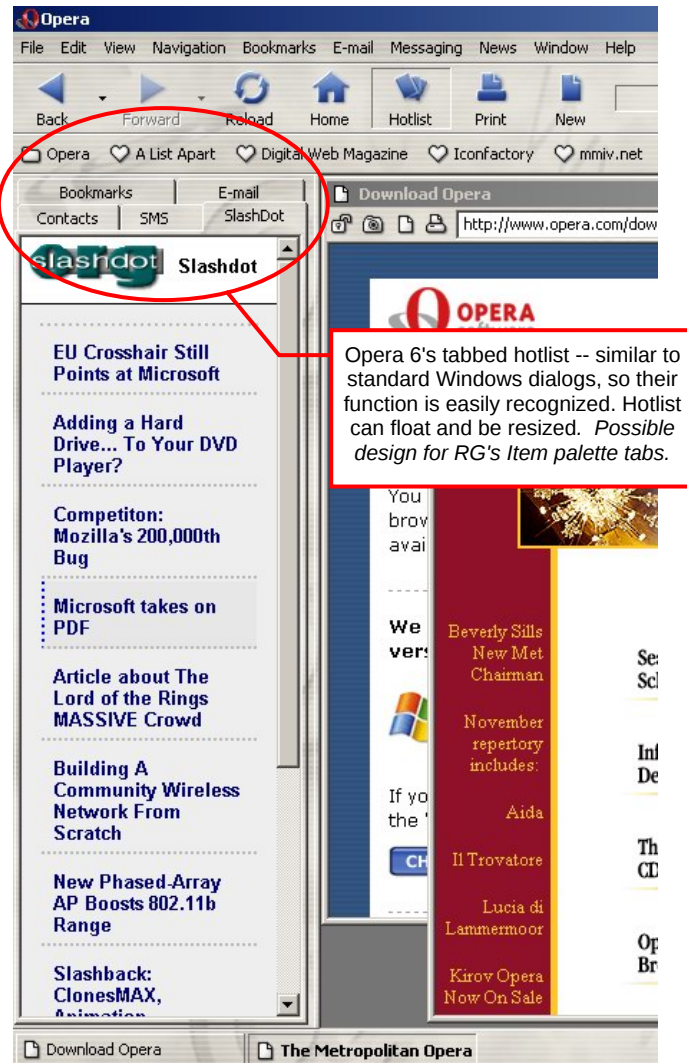
Toolbox. Equivalent to a toolbar, but uses less screen space on large (19"+) monitors. RG should use an ordinary toolbar for familiarity.

Tabs within palettes. Metaphor is obvious so design affords flipping, but only short labels fit. Possible design for RG's Item palette tabs.

Shaded palette saves screen space. Note control button to unshade. RG palettes could be shadeable.

Floating palettes can block workspace. Docked ones cannot because the workspace resizes itself to fit.

Opera and Mozilla



APPENDIX: RECOMMENDED READING

Content Management Systems and Knowledge Management

Advice on how to choose a wiki engine and discussions about the best set of text formatting rules.

<http://c2.com/cgi/wiki?ChoosingaWiki>
<http://c2.com/cgi/wiki?ChoosingaWikiStories>
<http://usemod.com/cgi-bin/mb.pl?WikiSyntax>

Online magazine about CMSes, primarily commercial ones.

<http://www.cmswatch.com/>

Five Mistakes not to make when choosing a CMS.

http://www.cmswatch.com/Features/OpinionWatch/FeaturedOpinion/?feature_id=84

Importance of Metadata.

http://www.cmswatch.com/Features/TopicWatch/FeaturedTopic/?feature_id=85

Explains what can go wrong with getting a CMS. Advises thought, planning, and open standards.

http://www.shorewalker.com/pages/cms_woes-1.html

Brief CMS comparison.

<http://www.commonsgroup.com/ideas/fulltext.shtml?x=212>

How to evaluate a content management system. (Good, but not quite on target for our case, since we would be making a pure intrAnet site.)

http://www.steptwo.com.au/papers/kmc_evaluate/

Open-source CMS: On the rise. (Mostly focuses on the big iron types like Plone.)

<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2897730,00.html>

Will open source rule CMS?

<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2884921-1,00.html>

Brain fade among CMS vendors.

<http://www.intranetfocus.com/blog/archives/000132.html>

Looking towards the future of content management.

http://www.steptwo.com.au/papers/cmb_future/index.html

Convincing people to accept TWiki.

<http://twiki.org/cgi-bin/view/Codev/ArgumentsAgainstTWikiOnIntranet>
<http://twiki.org/cgi-bin/view/Codev/HowToGetInternalBuyInForTWiki>

How to successfully deploy TWiki.

<http://docs.linux.com/documentation/02/09/18/1449248.shtml?tid=38>

2, 4, 6, 8, Let's All Collaborate.

<http://peterme.com/archives/00000349.html>

A small price comparison of some commercial CMS systems vs. TWiki.

<http://twiki.org/cgi-bin/view/Codev/PriceTag>

Knowledge Management 101.

http://www.intranetjournal.com/articles/200011/ic_11_29_00a.html

When Content and Knowledge Management Collide.

<http://www.destinationkm.com/articles/default.asp?ArticleID=1026>

Knowledge Management Horror Stories.

<http://www.destinationkm.com/articles/default.asp?ArticleID=923>

Grass Roots Are Greener: Knowledge initiatives advance from bottom-up successes, not by executive fiat.

<http://www.destinationkm.com/articles/default.asp?ArticleID=198>

3M call center case study.

http://www.cio.com/archive/090100_problem.html

The ABCs of Knowledge Management.

<http://www.cio.com/research/knowledge/edit/kmabcs.html>

Word Wranglers. (See the “Knowledge Taxonomies Facilitate Browsing” section.)

<http://www.intelligentkm.com/feature/010101/feat1.shtml>

Growing a Practical KM System.

<http://www.destinationkm.com/articles/default.asp?ArticleID=1036>

A better approach: requirements-focused CMS selection.

http://www.steptwo.com.au/papers/cmb_requirements/index.html

Various knowledge management links.

http://searchcrm.techtarget.com/bestWebLinks/0,,sid11_tax283210,00.html

Information Architecture

Foreword to Rosenfeld and Morville's book on Information Architecture (2nd edition).

<http://www.useit.com/books/rosenfeldmorvilleforeword2.html>

Search and You May Find.

<http://www.useit.com/alertbox/9707b.html>

Search: Visible and Simple.

<http://www.useit.com/alertbox/20010513.html>

Ask Tog, November, 1998. Reader Mail. (See “An alternative to website navigation bars,” “Representing link destinations,” “Protecting our users’ information,” and “Simplicity is disappearing.”)

<http://asktog.com/readerMail/1998-11ReaderMail.html>

Is Navigation Useful? (Good, but not a perfect fit for the situation. A knowledgebase requires more structure and navigation than most sites, and the users can be expected to learn the structure over time.)

<http://www.useit.com/alertbox/20000109.html>

Site Map Usability.

<http://www.useit.com/alertbox/20020106.html>

Prioritize: Good Content Bubbles to the Top.

<http://www.useit.com/alertbox/991017.html>

Intranet Usability

The different needs of internet and intranet sites.

http://www.steptwo.com.au/papers/kmc_intranetvsweb/index.html

When and How to Renew your Intranet.

http://www.steptwo.com.au/papers/kmc_renewintranet/index.html

Intranet Portals: The Corporate Information Infrastructure. (Also read the comments.)

<http://www.useit.com/alertbox/990404.html>

The 10 Best Intranet Designs of 2001.

<http://www.useit.com/alertbox/20011125.html>

10 Best Intranets of 2002.

<http://www.useit.com/alertbox/20020903.html>

Intranet Usability: The Trillion-Dollar Question.

<http://www.useit.com/alertbox/20021111.html>

Readers' Comments on Content Creation for Average People. (See the New Business: Trouble-Shooting Digital Content Design section.)

http://www.useit.com/alertbox/20001001_comments.html

Linux and Package Management

DistroWatch, the number one resource for researching Linux distributions.

<http://www.distrowatch.com>

An overview of several Linux distributions.

<http://www.distrowatch.com/dwres.php?resource=article-game>

User Review of Debian GNU/Linux 3.0r1.

<http://www.distrowatch.com/dwres.php?resource=review-debian>

Distribution Review: Sorcerer GNU Linux.

<http://www.distrowatch.com/dwres.php?resource=review-sorcerer>

Is RPM Doomed?

<http://www.distrowatch.com/dwres.php?resource=article-rpm>

RPM Hell: A Perfect Example of Good Software Crippled by Bad Design.

http://www.germane-software.com/~ser/Files/Essays/RPM_Hell.html

APT4RPM homepage.

<http://apt4rpm.sourceforge.net/>

FreeBSD homepage.

<http://www.freebsd.org/>

And what about BSD?

<http://www.extremetech.com/article2/0,3973,10423,00.asp>

The BSDs: Sophisticated, Powerful, and (Mostly) Free.

<http://www.extremetech.com/article2/0,3973,555398,00.asp>

Programming and Software Development

Overview of the Joel on Software archive, with large quantities of useful and interesting articles. See especially the User Interface Design for Programmers book online.

<http://www.joelonsoftware.com/navLinks/fog0000000247.html>

Strategy Letter II: Chicken and Egg Problems.

<http://www.joelonsoftware.com/articles/fog0000000054.html>

Why real GUIs are worthwhile versus shoehorning interfaces into browser windows.

<http://joelonsoftware.com/articles/fog0000000296.html>

Comparisons of various web scripting languages.

http://training.gbdirect.co.uk/courses/perl/comparison_php_vs_perl_vs_asp_jsp_vs_vbscript_web_scripting.html

http://php.weblogs.com/php_vs_cold_fusion

http://php.weblogs.com/php_vs_asp

http://php.weblogs.com/php_vs_perl

What can PHP do?

<http://www.php.net/manual/en/intro-whatcando.php>

Using Microsoft databases with PHP.

<http://www.php.net/manual/en/faq.databases.php>

FoxServ is Apache, PHP, and MySQL in one neat package.

<http://www.foxserv.net/portal.php>

<http://sourceforge.net/projects/foxserv>

Usability and Design

Making the Right Technology Decision.

<http://asktog.com/readerMail/1999-08ReaderMail.html>

Justifying Rough Sketches.

<http://asktog.com/columns/005roughsketches.html>

The Three Faces of "Easy to Use."

<http://www.opensourcecms.com/modules.php?name=News&file=article&sid=25>

Let Users Control Font Size.

<http://www.useit.com/alertbox/20020819.html>

Website Navigation Bars.

<http://asktog.com/columns/014WebsiteNavBars.html>

First Principles.

<http://asktog.com/basics/firstPrinciples.html>

A Quiz Designed to Give You Fitts.

<http://asktog.com/columns/022DesignedToGiveFitts.html>

User Testing

If They Don't Test, Don't Hire Them.

<http://asktog.com/columns/037TestOrElse.html>

Why You Only Need to Test With 5 Users.

<http://www.useit.com/alertbox/20000319.html>

Success Rate: The Simplest Usability Metric.

<http://www.useit.com/alertbox/20010218.html>

\$1.98, Close-coupled Usability Testing.

<http://asktog.com/columns/001closecoupleddtesting.html>

Are Users Stupid?

<http://www.useit.com/alertbox/20010204.html>

First Rule of Usability? Don't Listen to Users.

<http://www.useit.com/alertbox/20010805.html>

APPENDIX: PRICING SOURCES

Operating Systems

Windows

Windows 2000 Professional.

<http://www.microsoft.com/windows2000/professional/howtobuy/pricing/default.asp>

Windows 2000 Server.

<http://www.microsoft.com/windows2000/server/howtobuy/pricing/default.asp>

Windows 2000 Advanced Server.

<http://www.microsoft.com/windows2000/advancedserver/howtobuy/pricing/default.asp>

Windows XP Professional.

<http://www.microsoft.com/windowsxp/pro/howtobuy/pricingretail.asp>

Windows Server 2003 Standard Edition, Enterprise Edition, and Web Edition.

<http://www.microsoft.com/windowsserver2003/howtobuy/pricing/default.msp>

Linux and BSD

Mandrake Linux 9.0 PowerPack Edition, ProSuite 9.0, and Corporate Server 2.1.

[http://www.mandrakestore.com/mdkinc/index.php?MDK_STORE_REFERER=&LANG=&STATE_ID=US&UID=&CUSTID=&origin_ =](http://www.mandrakestore.com/mdkinc/index.php?MDK_STORE_REFERER=&LANG=&STATE_ID=US&UID=&CUSTID=&origin_=)

Red Hat Linux 8.0 Professional and Advanced Server 2.1.

<https://www.redhat.com/apps/commerce/>

SuSE Linux Professional 8.1.

http://www.suse.com/us/private/products/suse_linux/i386/index.html

SuSE Linux Enterprise Server 8.

<http://www.suse.com/us/business/products/server/sles/prices.html>

How to Get Debian.

<http://www.debian.org/distrib/>

Libranet 2.7 Full and Classic.

<https://libranet.com/download.html>

FreeBSD download page.

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/mirrors.html

OpenBSD CDs and Boxed Sets.

<http://www.openbsd.org/orders.html>

OpenBSD download page.

<http://www.openbsd.org/ftp.html>

How to Get NetBSD.

<http://www.netbsd.org/Releases/>

Databases

Microsoft Access 2000.

<http://www.microsoft.com/catalog/display.asp?site=769&subid=22&pg=8>

Microsoft Access 2002.

<http://www.microsoft.com/catalog/display.asp?site=10864&subid=22&pg=8>

Microsoft SQL Server Pricing.

<http://www.microsoft.com/sql/howtobuy/production.asp>

Overview of MySQL Products.

<http://www.mysql.com/products/index.html>

MySQL Commercial License Pricing.

<http://www.mysql.com/products/pricing.html>

MySQL Control Center Graphical Administrative Utility.

<http://www.mysql.com/products/mysqlcc/index.html>

Oracle Store.

http://oraclestore.oracle.com/OA_HTML/ibeCtpSctDspRte.jsp?a=b

PostgreSQL download page.

<http://www.postgresql.org/mirrors-ftp.html>

Programming Languages

ASP.NET download.

<http://www.asp.net/download.aspx?tabindex=0&tabid=1>

Comparison page for different editions of ColdFusion.

http://www.macromedia.com/software/coldfusion/productinfo/features_by_edition/

The Macromedia Store's page for ColdFusion MX Server Professional and Enterprise.

http://dynamic.macromedia.com/bin/MM/store/US/catalog_server.jsp?BV_SessionID=@@@@0316799445.1047233966@@@@&BV_EngineID=dkadcgjdeikllbeecgemcgchf.0

Sun's Java Downloads Home.

<http://java.sun.com/downloads/>

JSP information.

<http://java.sun.com/products/jsp/index.html>

Perl download. <http://www.perl.com/> and <http://www.perl.org/> are helpful too.
<http://www.cpan.org/>

Download PHP.
<http://www.php.net/downloads.php>

ASP2PHP conversion utility.
<http://asp2php.naken.cc/home.php>