

HAL MILLER

customer requirements specifications for system administrators



Hal Miller, having returned from three years as a recalled reserve officer (Lt. Col., U.S.A.F.), is in the UNIX Infrastructure group at BankOne in Ohio. In a previous life, he was president of SAGE, and before that, president of SAGE-AU, as well as a “working sysadmin.”

■ halm@sage.org

Whether you are trying to find your way around a “new” site, reviewing for a potential rework of your present organization, or planning for a new project, sometime during your career you will probably be faced with having to pin down a set of formal customer requirements. Informally, you face it every day, but most of us deal with that “by the seat of the pants.” Formal specification is much more involved and takes some planning in itself.

This article discusses the issues involved, purposes of the specification, goals, and roles. It is intended to be a guide to those sysadmins who suddenly need to fit this process into their already busy schedule.

We like to think that we are continually observing, understanding, and meeting customer requirements as they occur or effectively anticipating and preparing to meet those requirements before they arrive. As a rule, that is just wishful thinking.

In the world of “firefighting” that describes a typical sysadmin’s workday, such planning is pretty thin, and the time needed to plan around customer needs is just too costly to spend regularly. Thus, periodically we should consider tackling it as a priority project. That periodicity will depend on many things, such as how often we meet with customers already (in an appropriate forum, as opposed to individually at their desk), the nature of the business, and the size or complexity of the organization.

The Customer Requirements Specification (CRS) is a document normally used for computer systems support planning purposes. It identifies in some depth issues the customers have now and foresee for the next period, issues the computing support group foresees, and any potential business climate changes in the works.

The CRS creation process involves four classes of people: “customers,” those who use the computing environment to perform their own daily tasks; the “support team,” responsible for the design, implementation, and management of the computing environment; “management,” responsible both for approving budgets and for validating the requirements given by “customers”; and “vendors,” who can supply information on available technology and tools that might be applied once the CRS is completed.

A CRS should point out to the computing support team any gaps in current coverage, any “sore points” as seen from the customer’s vantage point, and holes

in current planning. It should reassure the customer about the support team's level of interest and ability to meet perceived needs, and the support team about their ability to prevent contention with customers and last-minute emergencies. It should also become the primary input in short- and long-term planning, both for budgeting (including personnel and organizational issues) and for infrastructure design work. The CRS is not an audit—that's a different process with a different set of goals. The CRS is a snapshot of thought, not implementation. It is important to avoid building the environment first, then trying to justify it by weighting the CRS to fit the design!

Lay of the Land

You will need an organizational chart that maps not only levels of official management, but all those who have an impact on budgeting and staffing plans. Begin by asking for the official organizational chart, then modify it to suit your goals, asking employees at all levels for their input. This process may have value beyond your immediate needs: the completed chart may highlight problems in the current organizational structure and thus perform an invaluable service for management as well.

Once you map the organization, assemble an accurate inventory and description of "what is." List physical facilities along with the business purpose of each; current computing facilities, including the space, power, and bandwidth available; and numbers of personnel supported at each facility. If users at a given location fall into multiple categories (e.g., heavy programming, administrative staff, and CAD), chart those uses and their impact. Determine the extent of inter-site computing and managerial relationships, storage currently in use, types of operating systems, software packages, and numbers of users of shared areas at each site.

Take a look at the management structure. Who actually makes decisions, considering both those in authority and those supplying options and recommendations? What process is used to determine what should go into the budget? What parts of the whole cost of doing business go into each manager's piece of the budget? Are personnel costs included in the sysadmin portion? Does that include training and conference attendance? Medical benefits? Team t-shirts (don't underestimate the value of team-building toys)? What is the information flow during the budgeting cycle? Who assigns tasks and responsibilities? Is there a reasonable delegation process, or is authority closely held (read: micro-management)?

Look at the IT Team. It should have some identifiable structure, even if "flat," "chaotic," or "harried" seems to apply. How does information flow to, through, and from the computing support folks? What is the current backlog of "repair" type tasks? Are larger-scale projects getting done? Are they even on the drawing board, or have people given up? What are the current problems—perceived by IT folks, management, and customers—with the IT Team? Are any changes anticipated? Are any agreed upon as being required even if not scheduled?

When you have a handle on the current organizational structure and situation, turn to the business itself. What is the true purpose or type of business? Is the organization there to conduct retail sales? Manufacturing? Supplying services? If a clear answer to this isn't in everyone's mind, something needs fixing right there. Don't be afraid to ask the marketing folks, whose job it is to deduce these items with pinpoint precision.

What sort of business model is followed? How does the organization run—what are the processes and practices? Try to trace on paper, step by step, the normal flow of information throughout the organization, bearing in mind that (1) it probably won't match the official line and (2) there may be many different flows for different purposes.

You might find some interesting “We’ve always done it that way” and “I just ask Joe” situations. What use is currently made of the computing power in place? Fancy typewriters or actual compute engines? Are they being underutilized, or overburdened for the tasks currently assigned? Are changes in business process being anticipated? Most important, how willing is the organization to review those processes?

When an inspection, review, or consultation is announced, people have reactions and expectations. Do customers feel threatened by the CRS? Do they have axes to grind and see this as a way to get their views heard, perhaps anonymously? What are customer expectations of the computing environment, and are those expectations being met? Are they knowledgeable? Are they even reasonable? Is there some change underway that may alter those expectations? Is there fear that the CRS might get someone fired or disciplined? Is there push-back, and, if so, from whom?

What other gains might there be from the CRS? What are the IT team’s expectations of it? Are there affiliated organizations that may need to be interviewed? Suppliers, dependent groups, potential competitors, etc., might have something of value to add.

When you have a good picture of the expectations and a good idea regarding any process reengineering you may need to tackle, goal-setting becomes important. The target becomes a refined set of goals and agreed-upon deliverables.

Two sets of goals will dominate: those of the CRS performer (you) and those of your customer organization, including all of its various parts. The CRS performer should work toward:

- obtaining the information required to recommend computing environment design;
- obtaining information required to size equipment;
- making projections and establishing what the growth patterns are;
- determining support requirements;
- pinning down costs, both up front and recurring; and
- performing process engineering for business and information flow.

The customer will supply a set of goals. Get them in writing; in fact, help them to draft the list. Make each goal clear, achievable, measurable, and focused (each with a single purpose).

Your deliverables include:

- A survey questionnaire
- All the raw data from the survey
- A summary of the reduced/refined data
- A final report

The report should cover projected growth, proposed process changes, proposed computing environment design criteria, a draft overall computing environment design, and a ball-park estimate of up-front, recurring, and personnel costs for the new computing environment as proposed.

Your Knowledge Target

Creating the set of deliverables requires a lot of learning and processing of organizational information. This section discusses the knowledge that will be chased down and how to target it.

A typical analysis of computing needs begins by examining the current systems. How many files are stored and in what sizes, types, and hierarchical layout? To whom do they belong—are they grouped somehow? What sort of file service is in place? Does it cover all machines, or just certain subsets? If the latter, is it by

design, by default (grew that way, or nobody bothered figuring out how to implement something more), or by technical limitation? What types of servers are in place? What types of client machines, operating systems, and applications? Where are they run—are they grouped, and if so, why? Look constantly for things that grew without design or things that should have been cleaned up and just never were.

What numbers, types (including experience levels), groupings, strengths, and weaknesses do you find in the user community? Do some tend to drive requirements more than others? Why? Do some tend to drive priorities more than others? If so, is it by authority or just by being the squeaky wheel?

Analyze the existing IT team. How is it organized and what are its strengths and weaknesses? Is training required? Training will need to be addressed later with regard to the new environment, but holes may already exist that need to be filled regardless of what new computing technology is applied. Include a full-scale, critical review of the non-IT team people in the organization who have root or administrator privileges. Emphasize that these people have the same responsibilities as the regular support staff.

Document the existing computing environment. Include a detailed inventory that covers the networking infrastructure and security implementation as well as servers and desktops, remote capability (location-independent computing, dial-in, etc.), and written policy and procedure. Analyze the help-desk function.

Determine and document your opinion on the adequacy of the environment (and its parts) for the current requirements and processes, including the perceived requirements of various groups in the organization. Carefully review the security status and requirements of these groups. This will be critical later when you try to justify the security implementation you will recommend.

In fact, drive hard on the security issues, noting the number of ways, posted on the Internet, to break into the organization's machines/devices/OSes/whatever. Carefully document the access requirements of those outside the organization; too often people "out there" collaborating with your organization on something demand completely open access, get it, and are not protected. Find all such instances, back doors, etc. Document existing location-independent operations and measures of actual usage.

The next step is to figure out where your information is hiding. Who has it, and is there hard-copy documentation? Electronic documentation is handy for general reading, but if it's on disaster recovery, it's hard to read it when it's most needed. Who has the corporate history for the computing environment, and what chain of control has been used to pass things along? What things are left as is because "things might break" if they're changed?

For all the items above, determine how best to obtain the information: through direct questioning, by probing servers, or by examining fielded configurations.

Writing the Survey

In those areas where people are the best source of information, compile a set of survey questions. Don't be afraid to use a template or gather examples from associates who have done this before or from sites surveyed in the past. There is nothing wrong with plagiarizing questions.

Begin by breaking down the knowledge target issues (see above) for which information is not available. List specific information that needs to be gleaned. Make the list very comprehensive—don't worry about the number of questions or the number of people to be surveyed.

Craft the questions carefully to get directly to the required piece of data. Try hard not to display biases here; avoid telegraphing the preferred answer in the question.

Target the question sets to specific audiences—there's no need to give exactly the same survey to everyone; fairness is not an issue. Bound the questions: instead of “Do you purchase software regularly?” ask “In the past six months, how many software packages have you purchased?” Be exact and avoid ambiguity. Instead of “Are you experienced in installing hardware?” ask “How many internal disk drives have you installed in PCs this past year?”

If you cannot extrapolate accurately from those answers, ask secondary questions, such as adding, “How many PCI video cards?” to the previous example. If you give multiple-choice questions, make the alternatives clear and show no leanings or biases. People tend to try to give the answer they think you're looking for, e.g., if there is an “all of the above” and each option is reasonable, people may tend to use it, meaning data may either be skewed or too weak to mean much.

Ask different questions aimed at getting the same data: “How much time do you spend per week doing X?” and “Order the following based on the amount of time you spend on each,” then cross-check them to validate as well as to build on your knowledge.

Ask some similar questions (perhaps in different forms) of members of different groups, looking for perceptions: “Do you answer all highest-priority trouble tickets within established time guidelines?” and “Did the IT team meet your highestpriority trouble tickets within within established time guidelines?”

Ask both the customers and the IT team members to order some sample tasks based on priority, to find out whether there is agreement and comfort with the prioritization process. Ask at least one text question that gives people the chance to express their opinion on overall issues. Obviously, you should check spelling, ensure sentences are complete (or use the “complete this sentence” approach where appropriate), be grammatical, be certain of your use of vocabulary, and make sure to use words in their most commonly understood meanings. Remember, simple words are fine when properly used.

H, have someone proofread your survey, both for language use (including typographical errors) and for ease of understanding. Discuss with your reader what they got out of each question, to make sure the respondents will be likely to provide the desired data.

Administering the Survey

As noted above, it is easy to bias a survey. The most common ways to invalidate results are to:

- Incorrectly select the participants.
- Be less than fully objective in your language.
- Structure questions in an unclear or ambiguous way.
- Mistime the administration of the survey.

The middle two were covered above; the first and last are addressed here.

Many factors influence how people respond to questions. One significant factor is the participant pool's other members. People tend to try to maintain a consistent reputation for themselves within a group. If the group they are in seems homogeneous based on some given factor, the answers will be homogeneous if questions revolve around that factor. If the group is very diverse, people may feel more free to give independent opinions.

Some factors to consider in selecting the participating audience include the size of the group, social factors, time of day, and the physical setting. Will the survey be administered to a large room full of people, small groups together, or individuals? Will each hear or be heard by others? Will responses be treated as anonymous or confidential? If grouped, will the groups be homogeneous or heterogeneous? Are managers and subordinates, genders, races, etc., combined or segregated? Are only members of certain groups selected, e.g., upper-level but not lower-level staff? Will the survey only cover day shift, excluding night folks? Will people take the survey home or otherwise work on it off-site, or will it be completed during work hours? One session, or multiple sessions? All of these affect the results.

The number of questions on a survey has some bearing on the accuracy of the results. Too few questions may yield too little information, but too many may yield contradictory, or, if the participant tires of responding, poor information. Decide exactly what information is desired and construct your questions to get that information. Be prepared to give out different sets of questions, in multiple combinations, to employees in different situations.

Data Reduction, Refinement, and Analysis

Once the information has been gathered, how is it processed? What does it mean? Reducing data from raw form to something structured, refining it to address the target knowledge list, and then analyzing the results is the whole reason for the survey process.

Begin by setting the knowledge target list on the desk in plain sight. Continually refer to it to keep on track. As the saying goes, when you're up to your knees in alligators, it's hard to remember that you're there to drain the swamp.

As much as it may be of interest to go down any side tracks that appear, just make a note to get back to them another time and press on. Otherwise the process will take forever. The exception, of course, is if you run into something so significant and so unexpected that it completely changes the purpose of your project, such as an organizational decision to change from manufacturing automobiles to selling hot dogs.

Good questions will generally yield some reasonably quantifiable responses, at least in most cases (some things are better asked in non-quantifiable form). Tallying responses is a fairly easy form of data reduction, and percentages can quickly be calculated. Of course, the wording of a question may turn out to be less clear than had been hoped. Responses may then fall into two or more groups: those who thought the question meant one thing, and those who thought differently. In other words, you may actually be looking at answers to multiple questions.

Solutions to this problem could involve resurveying, throwing out questions, or guessing. This is a good time to recall that surveys are not an exact science, but a picture of people's opinions based on their understanding of what is being asked. Don't rely on the results as "fact," but merely as indicators for planning purposes.

The hard part of data reduction is keeping nonquantifiable responses in context as part of the whole viewpoint of the participant. Many textual answers tend to get off the point of the question, and may be misleading if not understood within the framework of the rest of that individual's answers. Don't assume it's possible to cut and paste all the answers to question 37 into one large file and make complete sense of it—critical issues could be obscured by doing so.

Take the time, when reading a text answer, to glance through the quantifiable answers to relevant questions on the same survey response, to ensure that you

understand the participant's full intent. Yes, this is time-consuming, but it is important, especially for responses that cannot be tallied easily.

Once the data has been reduced, it's time to refine it. Raw data is numbers and text strings. Reduced data is effectively a summary of those numbers and text strings. Refined data is information. Analysis, which comes shortly, is the interpretation of that information.

Refining quantifiable data might include selecting a value-added method of presentation. To some, that's a chart or graph; to others, a list of percentages. Determine the preferences of the audience for the final report and try to accommodate their wishes.

For some of the process, you yourself are the target audience, so you get some input too! Refining this kind of data again includes throwing out questions, rephrasing, and asking again as necessary. Take whatever measures are required to give the results meaning. Regard this step as a reality check.

Refining unquantifiable data is a more difficult task but has the same goal. Presentation is more difficult, especially to managers who tend to base decisions on trends, budgets, or percentages. Some information doesn't fit neatly into numerical form, however. Bear in mind the most important points to be conveyed to the target audience, and try to see the presentation through their eyes—might they misunderstand any of these points? jump to a conclusion without seeing what is really there? This step doesn't deal with the results of the survey, just with ensuring that whatever the results were will be conveyed.

Once the summarization, presentation, and reality checks are done, the analysis phase commences. If the knowledge target list was clear and valid, the questions simple and to the point, and the data collection methods trustworthy, the refined results should neatly fill in the blanks on the knowledge target list.

If they don't, it is your job to decide whether you should interpolate, guess, or try again. The actual analysis varies dramatically depending on what the scenario provides and cannot be covered here in any depth. Don't get bogged down in numbers or semantics—the information is based on opinion and survey and is not “proven fact.” Use it to gain an understanding of what the real needs are, with the purpose of determining how best to plan to meet them.

A good CSR has plenty of built-in flexibility, since organizational needs change constantly, so aim for being reasonable and appropriate, as opposed to being right in any absolute sense.

Final Report and Follow-on Proposals

Once the data analysis is complete, it's time to assemble a report of the customer's requirements, along with proposals for meeting them. *This is the set of deliverables.*

During the process of preparing this report, keep the sheet describing the agreed-upon deliverables in front of you. Sometimes it is a good idea to structure those deliverables as a set of questions, at least for the purpose of writing the report—this gives a clear structure for responding to the customer and serves to remind you of which issues have not yet been addressed thoroughly.

The final report should briefly restate the goals and deliverables and then quickly review the process followed to achieve those goals and deliverables. The final report should be a reasonably short paper (effectively an executive summary) that summarizes your findings and recommendations in high-level, rather generic terms. The reader should be referred to appendices for both raw and refined data, with detailed findings and recommendations in additional appendices.

Appendices should include the following:

- Survey questions
- Raw survey data
- Refined survey data, in various presentation formats
- Findings and expert opinions
- Various options, including pros and cons, costs, and proposed timelines
- Recommendations

The findings are really the meat of the report. This is the assessment of the actual customer requirements, enumerated by category—CPU server power, storage space, reliability issues, network bandwidth, etc.). Explain what is needed, with occasional references as to why (but not in depth). Be prepared to defend those findings orally, and possibly later in writing, to those who will read your report.

Remember, you are the expert here, and the customer has asked for your opinion: render it. Two rules apply: professionalism, and the KISS Principle. Gear the CRS to your target audience, but even if they're computer-savvy, it's preferable to focus on the big picture rather than getting bogged down in detail.

If you are supposed to do additional consulting, provide proposals at this point that address your recommendations. For instance, you might be suggesting that you build out the new computing environment you're recommending, or that further investigation/research be undertaken for business process consulting. Proposal documents should be separate from the final report—you want your report accepted by the customer as a completed consulting task, regardless of whether or not they accept your proposals.

Periodic Reviews

One of your recommendations is likely to be a periodic review or performance of this sort of CRS. Needs change, organizations change, people change, technology advances, and (over time) the purposes for which a computing environment exists may have changed sufficiently to make the existing one obsolete or irrelevant. In any case, the organization will benefit from regular review to ensure that the original requirements specification was valid—remember, we're working with a survey of opinions here; relying on people's thoughts of the day is an inexact science.

Write your CRS, including goals, explanations (e.g., why you grouped the participants as you did), and recommendations, with a review in mind. Assume that someone else will conduct the next one. Even if it's you, it'll likely have been long enough ago that you will have forgotten your reasoning for whatever isn't written out in detail.

Appendix: Examples

KNOWLEDGE TARGET LIST

This is a sample list of information to be considered during the drafting of recommendations. Anything not known is a candidate for survey questions. A box (■) indicates “critical”; a dash (–) indicates a normal level of importance.

- Purpose of the business/organization
- Purpose of the computing environment
- Expectations of the computing environment

- Current information flow in the organization (not specifically computing-related)
- Projected information flow after a new computing environment is implemented
- Disaster recovery and business resumption models and plans
- Plans for hardware and software migration, life-cycle replacement, legacy issues
- Perceived problems to be resolved by a new design
 - Current network layout
 - Current computing policies
 - Current security implementation and policies
 - Who drives actual priorities for support work
 - The organization of the existing support group
 - Physical plant layout, number of floors and buildings, remote sites related
 - Number, size, and interrelationship of sysadmin groups involved; their depth and abilities
 - Number of current servers, OSes, manufacturers, models, hardware configurations (cards and card layout, space, cabling, OS versions, usage), life expectancy
 - Number of clients, purposes, integration requirements (AppleTalk, NFS, Samba, printing), file-sharing requirements
 - Applications in use, versions, predicted changes in patterns
 - Quantity of storage applied, models, brands, amount in use, usage patterns (e.g., periodic, long-term, number of readers/writers, aging process), data types, rates of change of files, databases, raw vs. cooked file systems, filesystem specifics (sizes, chunks), RAID levels, snapshot usage, mirrors, types of service (NFS, CIFS, etc.)
 - Number of users (on-site and off-site), external connectivity issues, collaborators or sharing issues, location-independence requirements, impact on security
 - High availability requirements, uptime
 - Software installed, license issues, version issues, restrictions
 - Computer room construction, availability of resources (power, space, air conditioning, cabling, security)
 - Number of servers accessing the same data
 - Business hours, maintenance windows
 - Interoperability issues between groups, sites, machines/OSes
 - Tools in use to manage site, monitoring, logging
 - Centralized vs. decentralized services: printing, email, servers, management reasoning
- Applications to be used in the new environment
- Growth expectations, storage, compute power, users, reasoning
 - Off-site connectivity requirements upcoming, collaborations, “shared” areas outside the new environment
 - Backup plans and intentions, off-site and on-site mirrors, reasoning
- High availability and uptime requirements predicted, reasoning
 - Hardware and software vendor biases, reasoning

- Possible changes to the business model, business purposes, collaborations
- Software version requirements anticipated, potential conflicts, support issues
- Support group plans
- Other vendors involved in design phase, implementation phase, internal people involved
- Corporate stability, financial status, funding for this project, budget model timeline
- Plans for training support group and users
- Support conflict managements, SLAs
- Special requirements, e.g., 10-minute snapshots, root/administrator access for users, high performance CPU crunching

SAMPLE SURVEY

INTRODUCTION

It is always a good idea periodically to review the computing environment to ensure that it continues to meet the needs of the user community. When resources are stretched and need to be expanded, or when significant change to the user community is in process, this review becomes even more important.

The organization is out of IP address space and is short of disk space. The backup process leaves significant gaps in coverage. Funding changes are on the horizon. The current computing environment grew in a piecemeal fashion over a period of years, from a time when the needs of the organization were different. Although this is a common situation, this environment needs review and possible reorganization for effectiveness and maintainability. This survey is designed to lead to a Customer Requirements Specifications document which will provide the basis of a plan for computing environment changes.

The CRS document will help define the services to be provided. Your timely participation in this survey is absolutely crucial to its success and will lead to an improved computing environment. Please feel free to ask questions or to make comments in the margin. Answer as many of the questions as you can. Thank you for your cooperation.

PEOPLE

- P1. What is the name of your group or project?
- P2. Who are the people in your group?
- P3. What are their usernames?
- P4. What computing equipment preferences does each have?
- P5. What actual computing equipment needs does each have?
- P6. How are these people divided into subgroups?
- P7. What permissions does each person need with regard to files outside of their home directory?
- P8. What permissions do the other members of each group need with regard to files in home directories?

- P9. Who else, outside of the group, needs permissions in your project and/or home areas?
- P10. What computer-use training does each person need? Who provides this?
- P11. How many new positions are scheduled to be added to your group? When?

EQUIPMENT

- E1. What computing equipment belongs to your group?
- E2. What computing equipment do the members of your group use?
- E3. What is the age and end-of-life projection for each item?
- E4. Is each suited to its current tasks?
- E5. Is each suited to upcoming requirements?
- E6. Will any be transferred to another group or organization?
- E7. What are your “uptime” requirements? Why?
- E8. Are there periods where you are able to accept “scheduled outages”?
- E9. How long would you be able to withstand a “catastrophic outage”?
- E10. How much data do you have online now?
- E11. What change do you anticipate to that level of data online? What rate?
- E12. What performance changes do you need? Why?

GRANTS OR OTHER FUNDING

- G1. What grants are current in your group?
- G2. When will each run out, in its current iteration?
- G3. What is the likelihood of each being extended? For how long? When?
- G4. What equipment does each grant own? Are there other co-owners as well? Co-users who are not part of the grant?
- G5. What ties does each grant make upon equipment, in the sense of funding?
- G6. Will any grants be transferred to another group?
- G7. What additional grants are being considered? When?
- G8. What additional non-grant projects are being considered? When?

SECURITY

- S1. What collaborative ties do you have to organizations outside of your group, but within the overall organization?
- S2. What collaborative ties do you have to external organizations?
- S3. What collaborative ties do you anticipate being formed? When?
- S4. How many of your group use location-independent computing? (This includes dial-in or remote login from other dial-in services or from other sites.)
- S5. What services do you and your group use remotely?
- S6. What use do you make of superuser/administrator access on your machines now? Why? When?
- S7. Are there parts of your data that require extra protection?

SOFTWARE

- O1. What software packages does your group use? Please give the version number of each.
- O2. Where do you get those packages?
- O3. What other software packages will you need? When?
- O4. What operating systems does your group use now? Why?
- O5. What changes do you anticipate in your operating system needs? Why?

MISCELLANEOUS

- M1. What current computing needs do you have that are not being satisfied?
- M2. Do you have special needs for additional services?
- M3. What current computing problems do you face?
- M4. What other computing requirements do you foresee?
- M5. What responsibilities do you see as belonging to the support team?
- M6. How do conflicts in requirements or priorities get resolved?