

Article

Why use Requirements Matrix?

Written by: Darrel Raynor

Question

"Why use a Requirements Matrix (RM) when tasks are in the project schedule and requirements are documented in the Requirements Document?"

This question comes up each time I recommend using a RM to improve project performance. The project schedule is usually built around deliverables and most technical deliverables do not correlate well with business requirements. The requirements document is far too lengthy to serve as an easy tracking mechanism. A RM addresses two project problems, requirements tracking and scope creep. It also helps control the project in many ways, providing a short (2-5 page) snapshot of the project in terms business people can understand.

Requirements, according to T. Capers Jones, is, "...the statement of needs by a user that triggers the development of a program or system." He goes on to say that 80% of projects are at risk of scope creep, and that an accurate measuring system greatly helps track scope changes.

William G. Smillie in the "Software Engineering Productivity Handbook" edited by Jessica Keyes, says, "But how do software developers know if they are meeting user requirements or even understanding them? Obviously, I.S. professionals and end users will have to agree on requirements and explore the types of systems features that are most valuable to the end user. Not so obviously, an agreed-upon process to measure whether or not the requirements are being met must also be established." A RM can be one aspect of that process, along with formal conversations to determine the 'rightness' and reasonableness of user requirements.

Suggestions

The uses for a RM are legion, the most important being recording decisions, milestone dates, location of supporting documents, and who is responsible for each requirement from a specifications and a development perspective. It should always be physically posted in the project area and always available (read only...) in a central LAN location. Use a RM for every project, not just I.S. development. To construct your first RM, have all of your project materials handy, especially the requirements document and detailed project schedule. If you are lucky enough to be in the first phases of a project, you can build your RM as you go. Feel free to experiment with format, I have had the best results using MS Excel, since it allows sorting to give you different 'views' of your RM.

Title the first column 'Requirements' and use to record the highest level of business requirement you have. There should be between 2 and 8. Bold and increase the font size on these. Now under each, list the mid-level business requirements (bold), and then the low-level business requirements if needed. Now title more columns: Priority, Owner, Approved (date), Doer, Accepted (date), and Document. Priority should be kept simple, no more than 4 levels and preferably 3, such as Required (show stopper, without which project is useless), Like (demonstrable cost savings or income producing), and Nice (intangible or small benefits only). Now use meetings with the business owners to continually insist that each requirement be prioritized according to your 3 or 4 tiered scale. This is critical, so be professional and relentless over time. Without these priorities and documented cost/benefit, you will have little input for your schedule and scope decisions other than everyone's emotions and 'favors'.



When you have agreement on priorities, get signoff at the highest levels to record, and fill in the Approved column. You may have a few stragglers that you have not yet gotten priority for, you should not work on these until priority is resolved. If the Doer has buy-in to the scope at this point, fill in the Accepted column. If not, work these people who are responsible for doing the task until agreement. Also record the document name and location where the detailed requirements can be found. The completed (to this point...) RM should be a project management deliverable required before moving to the next phase of the project.

Next time we will suggest many more columns to add to your RM as you move through the project, as well as other ways to use the RM to keep the project on time and scope.

Resources

If anyone knows of a mailing list or discussion group on software requirements or on any type of requirements tracking, please let me know.

IEEE Standards Collection, Software Engineering especially IEEE Std 830-1993 IEEE Recommended Practices for Requirements Specifications.

<http://www.apu.edu/~bmccarty/curricula/cs524/srd.html> Azusa Pacific University, Department of Computer Science, CS 524 Software Engineering I is a good, beginning outline for Requirements.

A Guide to the PMBOK - Project Management Institute www.pmi.org especially Project Scope Management 5.3, Scope Definition.

Why use a Requirements Matrix? Part II

Suggestions

Title the first column 'No.' and use a WBS-style numbering system. This will make it easier to refer back to earlier versions if needed as the entire document will not renumber. Next is 'Requirements'; use to record your highest level of business requirement. Now title more columns: Priority, Owner, Approved (date), Responsible (was Doer, updated from a reader suggestion), Accepted (date), and Requirement Document (was just Document...), along with an Approved (date) column.

The RM columns to add next should directly correspond to your project plan, as a high level view of your task list. Put 'n/a' in all cells that do not apply, so that blank cells represent work to be done. For your Analysis Phase, columns could be: Test Plan, Complexity, Security (plan), and other high level topics. For your Design Phase, add more technical areas such as Data Base, Prototype, and Proof of Concept. Design is also the time to add detail rows under the main headings for all your detail requirements at the level 'handy' to track. These will correspond to test cases, which is another column to consider adding now. Each cell can contain more than one name. Don't worry if it wraps and takes up more room!

For your Development Phase, columns to fill in for more detail include making sure all detail requirements have at least one test case covering them. Columns to add here include Code and other deliverables, and the test names your organization uses, such as Unit Test & Integration Test, with columns for person testing and person approving with dates. Useful here are at least one Plans & Docs column, to insure that your initial plans are carried through and approved. Another is Defect Log, to make sure that defects are all being recorded after Unit Test and that they generate updated or new test cases as required. Carry this through the Test Phase, adding User, Business, Parallel, or other test types, with needed detail. For your Implementation Phase, add final sign-offs, user acceptance, and any turnover areas that should be tracked.

As your project progresses, use your RM to record and track all your important tasks, and all the detail that comes with a project. One great value of your RM will be in presenting much of the same data as a project plan, the system or other deliverable, and as the RM. Different views can make the difference between one of your major constituencies understanding where the project stands and their having unrealistic expectations. I

keep an up-to-date copy of my RM in my organizer at all times along with an updated project schedule for all those impromptu hallway discussions or other meetings that come up regularly. The RM allows me to take planning and corrective action earlier than I could without it. Your RM is also a great 'War Room' candidate.

Improve or Reengineer Your Process?

Question

"How do we know when to improve a process or reengineer it?"

This is a frequent question, usually about a specific process, especially from project and team leaders on their way to project management roles. As people start to view project processes as a critical success factor for projects, they naturally want to improve their processes to better fit their custom projects.

Project management encompasses process management for not only the main work of the project, but for the project processes. If project management does NOT include process management in your organization, perhaps it should...

Discussion

When starting new projects for my clients and also when called in to 'fix' projects, examining project processes is at the top of my list. Many times quite a lot of the discord within a project team and between a team and management or other organizations can be traced to project process inadequacies.

Here are factors to mull over when deciding how to handle a process change: the size of the change, the number of and specific people involved, your organization culture, single or multiple organization involvement, complexity of the current process, if technology can be applied, the impact of the results you seek, if the process will affect multiple projects, and whether or not you will have management backing for your project.

Let's take a quick look at three different tactics to consider when one of your processes needs updating: Tweaking, or normal continuous improvement changes; Redesign, for those processes needing significant changes; and Reengineering, when you are in trouble or need a 'big hit' productivity improvement.

Tweaking

If most of the factors above are low, then gently improving your process is the way to go. Without a lot to gain, there are other areas where your efforts will be better spent. Minor, controlled changes can be planned and adapted to optimize or to address small process problems. An example would be switching your change control meeting day from Thursday to Wednesday to make a weekly document update schedule for your public internet pages. This type of change generally requires input and action from those whose work is affected by the change. Management may want to be informed, however will not be concerned. Tweaking of this nature can be successful without using many change agent tactics. Tweak at your whim!

Redesign

If many of the factors above show above the minimum, and the change required is not too complex nor does it involve many organizations, redesign may be your best option. Significant problems or bottlenecks in several areas can also lead to process redesign. An example would be to reform your change control team around the Marketing and Customer Support functions rather than software maintenance.

Perhaps prior, your change control function was driven by production MIS or in-house users. Now management needs different responses as the product may be farther along the product life cycle or for other reasons. Redesign may point you to delete or add a step or two in the process and tweak many of your steps. Redesign normally keeps many of the same process steps, and in this case perhaps changing only the priority setting functions and the major players. Your effort expended during redesign working the people side of the change equation will not be wasted. Redesign any time you see gross inefficiency, when technology in common use in your organization can be leveraged, or when business goals or climate change.

Reengineer

Market changes or at least one significant factor that is clearly in need can be enough to trigger a process reengineering effort. More normally, higher rankings in multiple factors may lead you to attempt to reengineer one of your project (or feeding!) processes. Make sure up front that you are not changing for change sake, and that you will have enough allies and management support to finish what you start. Reengineering is definitely riskier, defining risk here as the chance to gain higher rewards or to lose your investment, in this case your clout, time, and effort. Sometimes the downside to a failed reengineering effort is that NO changes will then be addressed for a long period of time.

Reengineering a process requires a thorough 'as-is' process review, setting your goals, and a complete re-charting of all the tasks and responsibilities that are now deemed worth doing. Today, technology almost always plays a critical role in our capabilities. Many techniques we now use, or at least are familiar with, may not have been available when your process was first laid out, or even as it evolved.

Questioning each step of your process, really getting down to whether or not each step adds value to the overall process, is a painstaking process in itself. Ask yourself when deciding to reengineer, whether your team has the knowledge, the time, and the patience to walk through all the steps one by one, probably with multiple groups multiple times. While your group may have something to gain, another group may end up feeling like they lost, whether it be people or control. Remember, just because you have an AH HA moment and realize where great savings can be had, your associates may well take some hard convincing, over a period of time. Reengineer (at your peril...) when you see huge gains from cutting or crafting many steps, when your change may make the difference between success and failure, or when adopting a new technology will allow you to leapfrog your current, out of date, processes.

Follow Up - Multi-Tasking

Time to Negotiate Task Switching Time?

Here is a quote from one of the project oriented discussion groups I follow: "Multi-tasking, road-runner, or focused effort thinking: Most of our folks have been interrupt driven since they started working in the corporate environment. If a "fire" erupts somewhere, go put it out then start looking for the next fire in addition to doing the project work that you have been assigned. In the Critical Chain (CC)/Theory of Constraints approach, we want people to focus on one and only one task and not multi-task. When people are able to "single-task", their efficiency and productivity on that project goes way up and CC works miracles. When people multi-task, CC suffers tremendously." Jack C. Randazzo, RTR Project Management, Lucent Technologies Inc.

TEAM TRAINING FOR SUCCESSFUL PROJECTS: USE A SKILLS MATRIX FOR PLANNING!

Published in: American Programmer, premier journal of software engineering

Datamation, international large systems journal

Datamation Europe, international large systems journal

Interact, journal of Hewlett-Packard computers

Technology in Business, newsletter

How is training your development team related to successful projects? Time-to-market, or product cycle time, is growing in importance. Methods and tools are rapidly evolving. Project teams are continually being asked to shorten their development cycle. One of the first areas they may be asked to reduce, or even do without, is their own training. As a Quality Assurance Director and as a management and technical consultant, I have been asked to review many 'time-challenged' projects in the past few years.

One common factor I found among the projects I have reviewed, almost without exception, is that adequate training of the development team was not planned for or funded. Very few projects had a team training plan. When the going gets tough, developers were asked to pick up a new method or tool, or even worse, both, while working at breakneck speed on the project. This article looks at the reasons why team training is critical and how to construct and use a team training matrix. SYSTEMS, TOOLS, AND METHODS ARE MORE COMPLEX

A common misconception among traditional MIS and non-MIS managers is that development team members can just "pick up new technologies in their spare time..." In the past, most analysts could learn using a new version of a mainframe sorting package, a new version of Microsoft DOS, or a source library system. These were relatively minor updates that built upon existing skills.

Today's enabling technologies use radically new techniques and tools and do not come without a price. For all that control, complexity is needed. It takes only a project or two that either fails completely or misses functional and time estimates by a factor of three or four to educate managers that integrating such knowledge cannot be done "on the fly."

Many managers are not used to planning for the systems administration and underlying control of many of the low-level functions of LAN and other Client/Server systems. One of my traditional estimates of the likelihood of a project's success was that it only addressed one area of technological change. Changing more than one of the database, operating system, methodology, development tools, languages, or platform along with major functional updates was risky, to say the least. Changing more than two at a time almost guaranteed failure or frequent scaling back of project goals. Today, however, projects are cropping up everywhere that change ALL of the above at once! No wonder there is chaos in application development. Your project team will need in-depth training in each new area.

Estimate Your Training Needs

Identify the skills your team needs for the project. My rule of thumb for methods and supporting development tools training averages tool vendor, third-party training vendor, and my experience:

- A. Ask the vendor for a plan to bring your team up to speed
- B. Ask a third-party training firm that you trust for the same
- C. Add the vendor's and training firm's estimates
- D. Add in your gut feel for the least amount you can get away with
- E. Divide the total by 3: this is the least amount you should plan for!

For example, if your tool or method vendor recommends 12 days per developer to really learn to get the best of their wares, the training firm recommends 8 days, and your best guess is 5 days, total these. Take your total of 25 and divide by 3, giving 8 & 1/3 days. You now have created somewhat realistic training estimates.

Don't divulge your formulae, or your management is guaranteed to hold you to your 5 day estimate, and very well may ask for a reduction. Fight for the full 8 & 1/3 training days in your project planning sessions! Make sure you cover all the days when the vendor or the training firm and your people can devote 100% of their time to training. Offsite is best, although working in a known environment can help direct attention to the areas that need tuning or other support.

Build Your Project Skills Matrix

Constructing and using a project skills matrix will pay back many times the effort expended. The matrix will have three areas of skills to review: application, technical, and team. To get started, meet with your internal or external customer to review any application-level skills needed. Twice in the recent past I have had customers volunteer to help train a team, once at no cost! They may welcome your team's learning more about their business. Your needed vertical skill areas may be financial, distribution, manufacturing, engineering, or a combination of these.

Next, meet with your system architects, technical planners, and technicians that have recently completed a similar project, whether in-house or for another development shop. Contractor programmers and consultants can be invaluable, especially if this is your first foray into a new technical realm. Identify all areas of technical skills that need to be added to or refreshed. Do not forget to list the skills needed to plan and execute a system conversion if you are replacing an existing system.

Finally, meet with all the managers that had any of your team members for their last project. Review each person's strengths in terms of how they relate and help lead the team, how they handle stress, changes to schedules and project content, and how they may best contribute to your project. Ask if they think training in leadership and teamwork would help them work at a higher level. Make sure you list the 'soft' skills your project will require, such as interviewing, prototyping, managing multiple teams, as well as estimating and managing the project plan. Areas frequently overlooked are how to run a meeting, how to gel as a team, accurately figuring and reporting project status, and many times even how to manage a project.

List all the pertinent technical, application, and 'soft' skills on one axis of your matrix. Order them by when they will be needed in the project. For example, if you are doing up-front paper-based usability testing, make sure you schedule interviewing and prototyping training early on. If you are doing usability testing with unit tested modules, those skills can wait until just before system test.

List your team members on the other axis. To have maximum flexibility, add four more columns: users, borrowed resources, consultants, and contractor programmers. You may be able to fill in for lacking skill categories by using an expert for a limited time. If you plan to use these outside resources, try not to schedule them for more than four hours per day project work. You will need the rest of their time to mentor your staff.

Now, being as objective as you can, choose a highlighter color for expert, one for skilled, and one for familiar. Highlight the skills you need to have experts, those you need skilled people, and those you can get by with passing knowledge in their respective colors. You may now want to reorder the matrix to show those skills that are crucial to project success at the top, keeping them in order of when you will need them, working down to the nice-to-haves.

To complete your matrix, distribute it to your entire team including consultants and contract programmers, the project sponsor, and any of your peers or managers that show an interest. Use a legend or key to show that you need four skill levels: expert, skilled, familiar, and completely lacking. One way to fill in the blanks is to meet individually with each team member. Another way is to meet team by team in groups of up to ten. A bonus for meeting in teams is that you can brainstorm ways to fill in your project needs. Make sure your team knows that they will be held accountable for their skill level estimates and assigned deadlines accordingly. This tactic may help reduce the optimistic 'skill creep' that many resumes have.

Many times management will review and approve a carefully thought out training plan where they will not approve your gut feel. Include your methodology, the risks to the project of not training adequately, and the estimated costs to complete your training plan. Offer several alternatives, with project schedules and costs increased to accommodate lack of training. Use your four extra columns to add flexibility to dates and costs by involving more users, borrowed resources, consultants, and contract programmers. Many times having mentors available, as long as they are not 100% scheduled, will allow you to reduce your training.

While no guarantee of success, my experience is that projects that use a skills matrix have less frustration and tend to be more controlled throughout their life. I am sold on the process of constructing and using a skills matrix. I am working on a Lotus Notes version that I will release to the public domain if there is sufficient interest. Please let me know if your firm would be interested in such a template application.

Side Benefits of Training Planning

Training planning sessions highlight the level of experience each of your team has in particular areas. Reviewing their prior training familiarizes you with their interests as well as their proven skills. You may run across someone with unexpected prior experience in an important area. At the least you will have valuable information for input to decisions such as whether to train the whole team on certain tools or set aside 'specialist' time for crucial areas.

Planning for training will keep the 'flail' time and needless experimenting to a minimum. It also allows you to do 'Just In Time Training'. People remember their training best when they begin to use it immediately. Many

people forget quickly that which they do not practice.

Training planning helps you negotiate with tool and training vendors. When you can spell out exactly what your needs are for the project, you are more likely to be able to price and close a package deal. Don't overlook points such as free seats for a vendor's public classes, free passes to annual user group conferences, and documenting your experiences as a 'success story' for their marketing efforts.

In evaluating project managers, how well their people were prepared for their tasks can be a critical measure. It can indicate how important increasing their people's skills are to a manager. We all know that meeting your dates with a quality product is paramount. Holding the interests and loyalty of team members by not asking them to do the impossible will increase the likelihood of your next project being successful.