

## What to Do When the Right Person Doesn't Come Along

© 1999 Johanna Rothman

You've written the job description. You know just what you want in this employee. You have one tiny problem—you just can't find that person. Now what?

Sometimes you can continue to wait for the right person to come along. Sometimes you choose to hire someone with inadequate skills. In either case, you don't have to just hope for the best. You have other proactive choices: hiring from within, hiring a candidate with some skills and training the rest, changing the way you work, and changing the job description.

### **Staffing Choices**

When you can't hire exactly the right person, you have several options:

- 1) Wait for the right person.
- 2) Hire some skills and train the rest. This generally takes two forms:
  - a) Develop the necessary skills in new or current employees. Decide how much, and in which ways you are willing to invest in employees. Training can be a very cheap alternative to hiring, especially hiring senior people.
  - b) Hire some skills and pray. This is risky, because you have—by definition—an inadequate employee. Your employee may not know how to succeed. You can reduce your risk if you allow for these inadequacies in the project schedule.
- 3) Change the work to be done. You might have better success if you:
  - a) Change the job description of your requisition,
  - b) Add a tool or technique to your development effort, even with the initial decline in employee productivity,
  - c) Change who does what on the project, or
  - d) Change the project lifecycle.

One key to successful hiring, especially in a tight labor market, is to know when to do what. One of my recent clients, Dan, faced just this decision. He analyzed his staffing needs and decided he needed two additional testers and a program manager (cross-functional team manager). The project staff consisted of one architect (Dan), 12 developers, four testers, four writers, and one release engineer. He made different choices for each of the open requisitions based on his ability to find appropriate staff.

Dan looked for some unique qualities in his program manager and testers. Because the program manager had to manage not only the program, but the engineering project as well, the job required some technical know-how. The program manager had to understand how software was developed, and how to assess engineering progress. The testers had to be able to develop black box tests for a new product domain, with inadequately written project requirements.

The four testers already on the project were very strong black box testers. To get the planned black box testing done, Dan had to hire two more testers within the next two weeks. Otherwise,

the project would not make its schedule, because it would take too long to train the testers for them to be useful on the project.

Dan had two problems—finding a rare individual for the program manager position, and finding two testers quickly. The two problems were different and required different tactics. Filling a highly constrained position is not the same problem as filling a less constrained position quickly. We'll look at the different tactics Dan could have used, and how he made his choices.

### ***Wait for the right person***

You may be able to wait to hire if you've planned the project and its staffing before actually starting the project work. If you haven't adequately planned your project or your work, you won't know if you can wait. If you planned the critical paths, and this employee's contributions to the critical paths, you will know the date by which you can no longer wait. The waiting risk is passing the date you need to the employee to start. Decide in advance when you will need to adopt a different plan, rather than continue to wait for just the right person.

Table 1 lists some advantages and disadvantages of waiting.

Table 1: Advantages and Disadvantages of Waiting

Advantages of waiting	Disadvantages of waiting
<ul style="list-style-type: none"><li>• You may be able to make the person productive quickly on the project, assuming you are able to hire just the skills you want in a person who will fit into the team well. At the least, you should be able to avoid an extensive loss in project productivity.</li><li>• If you wait to hire just the right person, you don't have to train the person on the tools or the domain.</li></ul>	<ul style="list-style-type: none"><li>• Presumably, you have some project with a specific staffing need. When you can't hire into that position, the project cannot get its work done. If you don't change the project's work, and if you don't hire into the open position(s), you have doomed the project to failure.</li></ul>

Dan couldn't wait for the staff he needed. His needs were dictated by time. Within the next two weeks, the daily lack of a project and program manager meant Dan would have to do that work. Since Dan was the architect, doing the project and program management meant the architecture work would be delayed or worse, not done at all. That option would have introduced risks that would be too expensive to mitigate for this project and the next one. Dan was also concerned about the testers. If he didn't bring in testers quickly enough, the test plans would not be completely written, and the product would not be completely tested. Dan wasn't sure what the testers did, but he knew he had to hire them, and fast.

### ***Hire Some Skills and Train The Rest***

This tactic is attractive when you know you can't wait to hire the staff you need. Consider growing the new skills in your current employees via training and mentoring. This has the benefit of growing your staff from the inside. You can then hire junior people, who may be easier to find, to take the place of the newly trained employees. Your current employees may have some valuable non-technical attributes such as loyalty, teamwork, dependability,

organizational knowledge, and relationships with others, all valuable productivity enhancers. If your staff are just missing technical skills, you can train and mentor them on the technical skills.

If you have staff who are not capable of learning the new skills, then hire new people and train them. Table 2 shows some skills that I've found to be directly transferable or learnable. If you're looking for skills in the left column, try finding those in the right.

Table 2: Skills that can be transferred or learned

Role to be performed	Look for these other skills (not experience)
Software architect	Ability to see the whole system. May already be working with an architect or working as a high level designer. Able to discuss requirements in the context of the functional pieces of the system. Able to establish good working relationships with the rest of the project team.
C++ developer	Smalltalk, LISP. Experience with working with other object-oriented systems and the ability to learn
Tester	Observant person. If a black-box tester, domain experience is helpful. If a white-box tester, understanding the code.
Release Engineer	Organized person who understands how the product needs to fit together. Works well with others.
Project Management	Technical lead, organizing functional-area-specific pieces. Ability to articulate goals. Technical team leader. Able to establish good working relationships with project team, functional manager, and program manager. Excellent problem solver.
People Management	Project or program manager. Someone who's already worked via influence instead of authority may be less likely to offend or otherwise make bad people decisions. Able to motivate, energize, and lead a group of people. Able to observe the organization. Thrives on ambiguity and willing to make decisions in the face of never-enough data.

You may perceive that you have a need for a certain kind of person. Your perception may arise from the way you do things— that is, your processes— rather than from the tasks you do.

Assume you decide you need more testers. If you don't test your work products along the way, via reviews and inspections, you will need more testers to be able to assess the state of the software. If you change the project activities, by initiating reviews and inspections for example, — you may not need more testers.

Maybe you think need more developers on your project. See if there's another way to staff the project:

- 1) Rearrange the work on the project, so that the project work happens in different ways. Some examples are: hiring technical writers to reduce developer time spent writing user manuals; hiring a project coordinator to handle project administrative tasks to free technical lead time.

- 2) Use an architect instead of more developers. Sometimes software organizations appear to need an "army" of developers because the system is not well architected and designed to fit all the pieces together. An architect may help you leverage your current developers.
- 3) Use more toolsmiths or testers. If your developers are spending more time than you prefer on building test infrastructure, or on black box GUI testing, consider hiring people who prefer to build infrastructure or test product.

If you reconsider people's roles, you may be able to rearrange the project work to avoid hiring people, or at least free up people to do tasks that are more important.

Dan was concerned that he needed two more testers on the project in only two weeks. He asked the project team whether they could think of another way to proceed. The team worked together and decided that if they did design reviews and code inspections, they could effect these changes on their project:

- They could "test" the code while still in development, by testing (reviewing and verifying) the design as a team.
- Design reviews could reduce the number of defects inserted into the code.
- Code inspections might find many of the defects the testers would have found.
- The testers could focus on building complex tests to provide more test value, because their "test-find a defect-submit defect- wait for fix- retest" loop would be greatly reduced.
- They could reduce the overall number of inserted defects, including the number of defects that had to be fixed and verified.

Members of the project team had begun by planning the project schedule to work the way they always had. But if they changed the way they would work, they could reduce the system test time required, and thus avoid having to hire more testers. Nevertheless, they still needed one more person who could read and write code. Dan chose to hire a developer who would be apprenticed to one of the senior engineers. The senior engineer would handle much of the review and inspection moderation and planning work, while mentoring the more junior engineer. In addition, Dan arranged for review and inspection training and mentoring for his entire project team.

Rearranging the work, doing things differently and doing different things gives you an opportunity to create and implement a different plan. The new plan (Plan B) may have less work overall and be generally more manageable and less risky. Your new Plan B has a very different emphasis on the project work. You can then understand what you're hiring for and the training you need to provide. Plan B may allow you to staff the project with your current people, train them differently, and then determine the follow-on impact to the project.

Table 3 shows some advantages and disadvantages of hiring some skills and training the rest.

Table 3: Advantages and Disadvantages of Hiring Some Skills and Training the Rest

Advantages of hiring some skills and training the rest	Disadvantages of hiring some skills and training the rest
<ul style="list-style-type: none"> <li>• May not have to cancel or postpone the project</li> <li>• Training generally motivates people</li> </ul>	<ul style="list-style-type: none"> <li>• May not be able to retain current project schedule</li> <li>• Providing training to some staff and not to others can demotivate your non-trained staff.</li> </ul>

***Change the way you work***

If you can't staff a project so you can work the way you want to, change the way you work to fit the project staff you have. This can range from changing when a project is scheduled to start or ship, to changing your lifecycle on a specific project.

**Rework the project's place in the company's schedule**

I once worked for an organization that had more projects in the pipeline than the engineering organization could create. We couldn't find enough test people with appropriate domain expertise. With the help of senior management, we ranked the projects, and decided which projects would be staffed when. We did not hire new people. The effect was to do fewer projects faster. By putting each product out into the world, we created more people with domain expertise.

The original project schedule looked like this:



The changed project schedule looked like this:



We couldn't overlap Projects A and D, and since they were the highest priority, we did them sequentially. Project C could start before Project D ended, without the risk of context switching the Project C staff. Because Project B had a number of features in common with C, we arranged for them to reuse each other's work.

## Do the project differently

An alternative to rescheduling projects is to change the project approach. I was the SQA Manager at a highly matrixed organization. We had 15 people in SQA, and 8 projects to test. If we were to staff them and run the projects the way we always had, each project required 4-6 SQA staff-people. I worked with the project managers to get them to agree that we could test the products differently. We eliminated traditional black box testing<sup>1</sup>, and focused the project's efforts on unit and integration testing.

We generated this responsibility matrix:

Table 4: Project Responsibility Matrix

	Old Responsibilities	New Responsibilities
Developers:	Design, implement, and walk through code	Design, implement code and unit tests, inspect code and unit tests
Testers:	Develop automated system tests	Moderate code and unit test inspections. Run unit test regression suite. Develop unit test harnesses

The testers in this organization had the same capabilities as the developers. We created several projects that made the developers much more responsible for creating and testing their own work. The testers and developers became allies—not adversaries. We chose to reorganize the work to use change what people did and their roles on the project.

When you do things differently, you need different people to do it. If the people you have don't match the work you have, and you can't change the people, change the work.

One example of changing the work comes from history. When George Washington took over as General of the Army in the American Revolution, the army had a very specific job description. As part of the army, you stood in long parallel lines and shot at your enemy. Your enemy stood in long parallel lines and shot at you. Washington's army couldn't do that, as he found out early in the Long Island campaign. The British surrounded him on three sides, with his back to the water. Convention then required that he surrender—the defined job in this situation. But during the night, he “changed the work”—he ordered a nighttime evacuation across to New York and saved the army, and probably the revolution. The people he had couldn't do the conventionally defined work, so he changed the work to be something they *could* do. He also changed warfare. One hundred sixty years later the British did the same thing at Dunkirk, and they saved their country too.

Another example comes from science fiction. In Star Trek, all StarFleet officer cadets have to experience a test simulation to graduate from the academy. The cadet, acting as Captain, receives a distress call from a stellar-freighter, the Kobiyashi Maru. The simulation has two potential outcomes: either ignore the distress call and let countless people die on the freighter, or respond to the distress call that leads the cadet into an ambush and destroys the rescuer. The purpose is to

---

<sup>1</sup> I don't generally recommend eliminating system testing from a project. This particular product had no GUI, just a well defined and delimited API and Command Line Interface (CLI). We would not have been successful using this strategy on a product with a high GUI component.

show the cadets that you just don't always win. Cadet James T. Kirk succeeded in his Kobiyashi Maru simulation by changing the work—rewiring the console the night before the test. He became the only cadet in the history of StarFleet to successfully complete the simulation.

### ***Change the job description***

When you change the job description, reanalyze the skills you are looking for. Ask yourself questions about both the work involved in the job, and the skills required to perform it. What are the essential job functions—what does this person absolutely need to do? What results do you require? What secondary skills would you like to get in this person, if you could? What are the minimum required technical skills the candidate absolutely must have?

If you can sort candidate skills and job functions by order of importance, you'll be better able to make tradeoffs. For the candidate, ask yourself: What does this person absolutely have to know? What else would I like this person to know how to do, if I can find the right person? For the job, ask yourself: What tasks absolutely have to be performed by the person in this job? What other work would I like this person to do?

In addition to the minimum technical skills, what work experiences are required before you are willing to consider the candidate for your job? Must the candidate have worked on a similar kind of product? Potential candidate experiences range from:

- Technology experience. Candidates may be quickly productive if they have the same or related language, tools, and operating system experience. Even if the candidate does not know your product line, technology familiarity may make it easier for the candidate to be productive.
- Subject domain experience. You may want someone who has been thinking about the problems of this particular kind of product for a while. Or, you may want someone new to this kind of product, so that you get the benefit of fresh eyes and attitude.
- Industry experience. You may want specific industry experience, if you want candidates who:
  - already understand your customers' issues, or
  - understand how to work with another organization (on a strategic partnership project for example), or
  - know something about future industry trends, or
  - understand why something is the way it is (past mistakes).

Decide what parts of the job description provide the most value for your open position now. Dan looked for a project/program manager who could be effective in an extremely short period of time. He found someone who had worked on a variety of projects in a similar industry. The project manager's strengths were her ability to work the critical path and to continue to move the project forward. She did not have substantial industry domain expertise, but understood what was required in her behavior to be successful.

Once you've reanalyzed the job, you can reword the job description, and resume your search for candidates.

**Summary**

Dan changed his hiring plans and took several different actions as shown in Table 5.

Table 5: Dan’s Revised Hiring Strategy

Original hiring plan	Revised hiring plan
Two senior testers	One junior developer and inspection training
A project manager with specific industry experience	More experienced project manager/program manager without specific industry experience.

Finding just the right technical person remains a challenge. If you wait for the right person to come along, there may be substantial risks: late or high-defect projects are some common consequences. Hiring an insufficiently trained person may have even worse consequences. If you think you have hired appropriate staff but you have not, the project will most likely fail.

Hiring a person with some of the necessary skills and then training that person in the rest of the skills can have some unintended side effects. Your other employees may be excited that you are willing to invest in your employees, and ask for training themselves.

To get the most flexibility in your hiring, consider changing the rules of the game—change the work to be done. Reorganize the project, change who does what, and change how it gets done. Reorganize the project (can you use a different project lifecycle?), change who does what (who does what kind of testing or writing?), change how it gets done (introduce alternative techniques such as inspections or reviews, to reduce the need for some kinds of testing). This will have the most positive effect on your staff and on the work to be done. This will also reduce your project risk.

Knowing how to solve the staffing problem is critical to your success as a manager. Plan your staffing to create project success, whether it’s the Revolutionary War or just the next release of your product.

**Acknowledgements**

I thank these reviewers for their very helpful comments: Rick Brenner, Brian Lawrence, Barbara Purchia, Steve Smith, Jerry Weinberg, and Becky Winant.



# Johanna Rothman

---

Johanna Rothman observes and consults on managing high technology product development. She works with her clients to find the leverage points that will increase their effectiveness as organizations and as managers, helping them ship the right product at the right time and recruit and retain the best people.

Johanna publishes "Reflections," an acclaimed quarterly newsletter about managing product development. Johanna's handbook, "Hiring Technical People: A Guide to Hiring the Right People for the Job," has proven a boon to perplexed managers, as have her articles in *Software Development*, *Cutter IT*, *IEEE Computer*, *Software Testing and Quality Engineering*, *IEEE Software*, and *Crosstalk*.

Johanna is the founder and principal of Rothman Consulting Group, Inc., and is a member of the clinical faculty of The Gordon Institute at Tufts University, a practical management degree program for engineers. Johanna has an M.S. in systems (software) engineering, a B.S. in computer science, and a B.A. in English literature. She holds two ASQ certifications: Certified Quality Auditor and Certified Software Quality Engineer.