

**SKILLS FOR**  
**SOFTWARE**  
**SMOKEJUN**  
**PROJECT MID**  
**BY**

GETTY IMAGES



# RE MPERS

**INING A  
STREAM**  
DON GRAY

**D**o you know about smokejumpers? They're brave, self-sufficient firefighters who parachute into remote areas wearing eighty pounds of gear and ready to fight a forest fire. If the jump goes well, they land safely. After extinguishing the fire, they may have a ten-mile hike out. It's not a job for the faint of heart, slow of mind, or weak of back.

Have you considered that you may be a smokejumper? Think about it: Many of you join software projects midstream because sometimes a project needs additional contributors—some add brains, others brawn. Sometimes mentors are needed to improve project performance. Sometimes management needs an outsider's view of the project status.

No matter why you join a project after it begins, you will encounter challenges. To be successful you must:

- Determine your role
- Build trust
- Learn the territory
- Gather information
- Do your job
- Declare victory

## Determine Your Role

“If you don't know where you are going, you'll probably end up somewhere else.”

— *Laurence J. Peter*

Smokejumpers work on well-defined teams. Everyone has a job to do and knows how to do it. Before jumping into a project you should determine your role. Ask:

- What specifically is my sponsor asking me to do?
- How can I demonstrate to my sponsor that I have been successful?
- What will my relationship be with others on the project?

Knowing what my sponsor wants keeps me focused. Difficulties arise when the sponsor has difficulty explaining the problem. He knows the project is late and he wants better quality, but he can't say exactly why the project is late or what's creating sub-standard quality. Generally, the greater the pain (lateness, poor quality), the less articulate about the problem the sponsor becomes. When this happens, I like to use the SMART acronym Johanna Rothman described in “Release Criteria: Is This Software Done?” (*STQE* magazine, March/April, 2002). SMART reminds me to get a problem definition that is: specific, measurable, attainable, relevant, and trackable.

Next, I need to know what “done” means. Knowing how I'm going to demonstrate “done” gives me information on what to track so I can provide my sponsor the information needed to prove the fire is out. A good question to ask is “What will you see, hear, and feel when this problem is solved?”

I often use a simple reporting format when I check in with the people for whom I'm working. I describe what we've done since our last discussion, what we're currently working on, and any barriers to progress we're encountering.

Last, but not least, I need to know who I will be working with and in what capacity. Based on what I'm being asked to do (brawn, brains, mentor, or project review), I'm going to relate to the team in different ways. I may be a coworker, a coach, or an investigator. Knowing which role I'll be in guides me as I work on getting a demonstrable “done.”

Currently, I'm working with a client whose staff has been trying to solve a problem for a month. In our kick-off meeting, we established that my job was to get the project “done,” and they don't care how. On this project, “done” means all the applications have been switched to the new server and tested and the old server decommissioned. I'm going to function primarily in the brains/brawn role, as a coworker helping solve the problem. Along the way, however, I'm going to be asked, “Why couldn't the team solve the problem?” which will put me in an investigator/reporter role.

## Build Trust

“The first thing to build is trust.”

— *Brad Appleton*

Smokejumpers work in integrated teams to put out small fires before they spread or to provide additional manpower on larger blazes. As a project smokejumper, it's likely you'll be joining a pre-existing team. So when your boots hit the ground and your chute is secure, you'll need to hook up with the team. Your success in working with this team will depend on how well you understand them and how much they trust you.

Building trust is a relatively straightforward activity. If you say you're going to do something, do it. If you say you're not going to do something, don't. The team—and its management—will be looking for discrepancies between your words and your actions. Building trust is an action-based activity. When I hear “Trust me!” from someone I do not know well, that is a red flag that throws me into the “Put up or shut up” mode. So make commitments and meet them.

Keeping activities, information, and decisions visible helps build trust. It's not always possible to achieve immediate success (however minor it may be). Keeping things in the open helps allay fears, enhances communication, and enables better decisions. On one of my jumps, a team member heard me discussing a spreadsheet I was using to keep track of assets, current status, and needed changes. He asked for a copy of the spreadsheet and later returned it to me with the names and phone numbers of the key people I should coordinate with at each plant. By sharing the information I had, I received more.

Asking questions opens the door for team members to share what they know about the fire. Listening to and understanding their answers creates rapport and builds trust. This also helps you learn the territory and gather information.

## Learn the Territory

“You gotta know the territory.”

— *Meredith Wilson in “Rock Island” from The Music Man*

“I know the territory.”

— *Meat Loaf in “I’ll Do Anything For Love”*  
from *Bat Out of Hell II*

Smokejumpers work in an ever-changing environment where understanding the territory can be the difference between putting out the fire and not making it out alive. Their territory includes fuel types, wind direction, and the topography where they’re working. Changes in any of these can change the possible outcomes quickly.

Project smokejumpers also work in highly dynamic environments. Personality differences fuel the project flames. Some team members may be more equal than others. Who’s really in charge? Even though someone can’t help me, can he hurt my ability to succeed? Changes in any of these can quickly change the possible outcomes.

In all my years of jumping, I never have landed in a situation that lacked energy. Most situations follow the pattern of a jump I did a couple of years ago. For reasons no one could determine, a stable, proven process suddenly started generating a 25 percent defect rate. The Big Boss flew in from the home office to handle the situation personally. On Saturday, I got the call to jump. When I arrived Monday morning, I surveyed the situation, listened to the management screams and the worker apologies and decided it was a good time to be calm. I took a deep breath and started asking questions.

In her book *Communication Gaps and How to Close Them*, Naomi Karten lists my three favorite questions:

1. How did you happen to come here?
2. What do you expect will happen here?
3. What do you hope to accomplish here?

She also says, “Notice that the first question elicits information about events from the past; the second, the present; and the third, the future. All three questions provide a starting point to help you determine what’s important to the person or group with whom you’re trying to communicate.” These questions represent starting points for learning the territory.

The responses to the questions indicate how safe the person feels. Short, simple answers may be a tip that the person isn’t feeling safe. Perhaps there’s a blaming corporate culture and whoever’s holding the blame when the music stops gets fired. It could be personality conflict on the team. Unresolved conflicting management agendas can cause a “CYA” environment. If the environment isn’t safe for the employees, it’s not going to be safe for the smokejumper, either.

The key to success and survival hinges on the smokejumper’s ability to ferret out information about why the person doesn’t feel safe. If I haven’t created a trusting relationship, I

won’t get the information I need to learn the territory. I talk to as many people as possible, which allows me to draw a more accurate map to help me navigate the territory.

Do the answers’ content and delivery style agree with each other? I remember one project manager yelling at me about how well he had done on a previous project and how this project wouldn’t fail. I wondered why he chose to do this project differently, but decided to keep my mouth shut. He was partially correct. The project didn’t fail, but he and his company (the management team) were terminated six weeks later. It took us another year of development to complete the estimated twelve-week project.

## Gather Information

“As a general rule, the most successful man in life is the man who has the best information.”

— *Benjamin Disraeli*

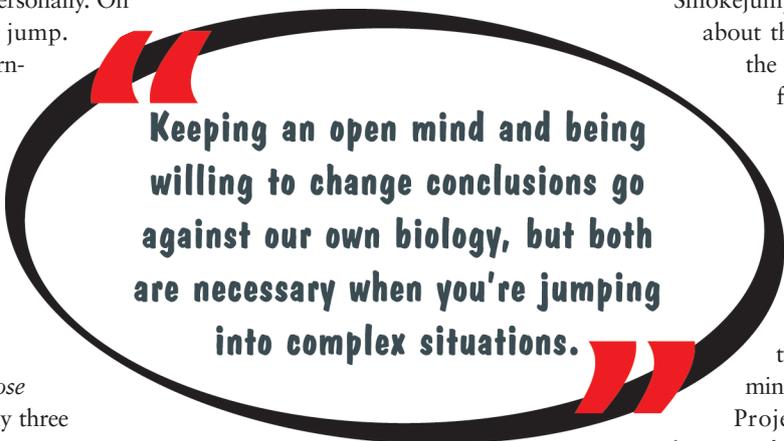
Smokejumpers receive information about the fire before they board the airplane. How big is the fire? What are the weather conditions? Which way is the fire headed? Are there obstacles to overcome? Where are the safe zones? As soon as they land, their first action is to verify the information and determine if anything has changed. Project smokejumpers start gathering information during the first conversation with the client. What’s the nature of

the problem? How long has it been going on? What already has been tried to solve the problem? You need to gather information about the technical fire you’ve been asked to put out.

I once jumped to solve a “three systems quit working” problem. After listening to the problem description, I thought about the symptoms and several possible cause/effect scenarios came to mind based on other successful jumps. I continued to ask questions, and one by one ruled out the possibilities. When I jumped, all I knew for sure was a problem existed. (For the curious, the software quit working because someone created separate IP subnets, and the computers couldn’t talk to each other because the inexpensive routers couldn’t bridge the subnets.)

Project smokejumpers compare this information against their past experiences. What appears to be the same? Is something new or novel? This provides the project smokejumper with an initial problem assessment. This is both good and bad.

The difficulty with this initial assessment comes when it leads us to ignore new data. Since we have an idea of what the problem is, we may believe we have the answer. As Lee Copeland



**Keeping an open mind and being willing to change conclusions go against our own biology, but both are necessary when you're jumping into complex situations.**

pointed out in “Believing Is Seeing” (*Better Software* magazine, December 2006), the Bruner-Postman experiment shows that our experience can blind us to reality. Keeping an open mind and being willing to change conclusions go against our own biology, but both are necessary when you’re jumping into complex situations.

Try to find both positive support and negative indicators for the problem you’re trying to solve. In my career as an emergency medical technician, I was taught to evaluate data and revise my understanding using the following checklist:

- I expect to see something, and I do.
- I don’t expect to see something, and I don’t.
- I expect to see something, but I don’t.
- I don’t expect to see something, but I do.

We can use this new data to modify our problem assessment. This forces a rethink and possible restructuring of our problem assessment. It takes time but opens the door to a better assessment and solution. The other choice is to ignore the information or modify it to fit our problem assessment. This doesn’t require rethinking and restructuring our assessment. It also opens the door for high-impact learning when the information we ignore comes back to burn us.

The technical problem could be something as simple as finding that the computers are on different subnets and thus cannot communicate. Maybe it’s helping the team come to grips with the “process du jour.” Perhaps the team’s engineering practices need modification to achieve the project goals. Whatever the technical problem may be, expect that it will be difficult to solve. If the problem had been easy, the team most likely would have solved it already.

## Do Your Job

“For every complex problem, there is an answer that is clear, simple, and wrong.”

— *Attributed to H.L. Mencken*

Smokejumpers use different tactics to extinguish fires. If the fire is small enough, they may directly confront it. For other fires, they may use an indirect approach of control lines and backfires to deprive the main fire of fuel. When the fire really gets going, they may have to wait for something to change—the type of fuel, the weather, the topography—before they resume fighting the fire.

As a project smokejumper, how you attack the problem is affected by your role in the project, your ability to build trust, your understanding of the territory, and the information you’ve gathered together with the problem’s complexity.

A brain/brawn role and a straightforward problem generally lead to direct action. This is how I dealt with a “software quit working” jump. I sat down at the computer and started checking settings, properties, and configurations. When I discovered two network cards, I started investigating more, and voilà! There were the two non-mappable subnets.

Mentoring or complex problems often require indirect ap-

proaches. I once spent three months helping a hardware team as it worked to get a hardened mobile router into beta production. Since management complained about not knowing where the team stood, I created a burn-up chart in the engineering space so everyone could see what remained to be accomplished and when we anticipated that it would be completed. The semi-weekly status meetings asked the basic Scrum questions: What have you done since our last meeting? What are you going to work on? What problems are you having? I made sure I had the necessary equipment, so if there was a question, I could go in and work with the team. We made the target date with a few days to spare.

Each style of doing things has natural consequences. If I decide to take control and work directly, I’ll miss an opportunity to let others learn through experience. If I let others learn through experience, what happens to putting the fire out? I like to use the following questions to help me understand the implications of my (mentally proposed) actions:

- What will happen if I do?
- What will happen if I don’t?
- What won’t happen if I do?
- What won’t happen if I don’t?

The process of understanding your role through doing your job goes on the entire time you’re fighting the fire. It’s a continuous refinement process as the project smokejumper learns more about the technical problem, the team, and the interactions between them. And it’s not a linear sequence. Other than starting with determining your role, the rest of the activities happen randomly, simultaneously, and continuously.

## Declare Victory

“The Lone Ranger Fantasy: When the clients don’t show their appreciation, pretend that they’re stunned by your performance—but never forget that it’s your fantasy, not theirs.”

— *Gerald M. Weinberg*

After they ensure the fire is out, smokejumpers head back to base, clean up, and repack their gear, getting ready for the next jump.

Prior to heading out, project smokejumpers need agreement from their sponsors that the fire is out. This task combines defining a specific goal at the jump’s start and keeping progress visible. If you don’t know what you’re shooting for and when you need to hit it, you’re probably going to miss the target. Hitting the target gets you praised and paid.

Declaring victory creates the opening to review your contributions to the project. Project smokejumpers often make technical contributions on a project. These are generally obvious, and most people would agree to them. Often more important are the less-obvious personal contributions. No one but the smokejumper may ever know the many little bumps, nudges, and guidance provided during the jump.

I've just finished working with a sponsor who a week ago said, "Again, today, the reports did not get sent out. I guess all of our work was for nothing." I reminded him that the problem involved two different systems, we had only corrected one, and that things would get better when we solved the problem with the second system. Today I heard the happiness in his voice as the second system came on line.

## The Smokejumper's Life

"You live and learn. Or you don't live long."

— Robert Heinlein

The smokejumper's life consists of:

1. Qualify for smokejumping.
2. Train.
3. Go to the fire.
4. Put out the fire.
5. Go to 2.

Over years of jumping, the training will change as the jumper becomes more practiced at current skills and learns new skills. Smokejumpers use all their skills, all the time. Being able to call on their training when they need to can mean the difference between an extinguished fire and an unhappy outcome.

Project smokejumpers follow the same pattern. Somehow, somewhere, you start solving problems, and then someone asks

you to jump in to help them.

Project smokejumpers need to train continuously. Your technical skills may get you started. It's your people skills that help you solve the problem and keep it solved. In addition to reading magazines and books, I recommend attending experiential conferences or training courses where you'll be able to learn new skills and practice them in a supportive environment.

Jumping isn't for everyone. Over the years I've missed birthdays and anniversaries. I once left a weeklong vacation after only two days to make a jump. Fortunately, my family loves me. It occasionally gets tense, so a sense of humor works to my advantage. Being an adrenaline junkie helps too. And it's all worth it when a client says:

"You know, Don, a couple years ago I watched you 'join a team' and help them work together better when your charter was actually to get something shipped. You weren't there to 'fix them.' But, you ended up helping that team and another team be better together."

That still gives me goose bumps. {end}

*Don Gray (www.donaldegray.com) has more than twenty-five years of experience solving machine and process automation problems, but he now focuses more time and energy on organizational issues. Don is a principal at nth Order Systems where he works on integrating people, projects, and processes. He is also a host at the Amplifying Your Effectiveness (AYE) conference (www.AYEconference.com).*

## Traveling for Training Got You Down?



Travel from your desktop with Software Quality Engineering's eLearning course, eMastering Test Design. Experience classroom value with the convenience of self-paced instruction on the Web. Because people learn in different ways and everyone learns best with multiple learning modes, we've combined audio, video, text, graphics, examples, questions, exercises, and additional learning resources for the best Web-based delivery possible.

## eMastering Test Design

### Classroom Value with the Convenience of Self-paced Instruction

- Instructed and designed by two of our most experienced instructors, Lee Copeland and Rex Black
- Same valuable information as the 2-day classroom course

### The Perfect Solution for Test Practitioners with Travel and Time Constraints

- Complete an eLearning course from your own desktop
- Expert mentors provide answers to your specific questions

### Course Content

- Superior lesson material developed and delivered by testing experts
- Tutorials that place content into real-world situations
- Exercises that immediately apply your new learning
- Assessment questions that help you evaluate your learning
- Questions linked to content to reinforce your learning
- Video and audio clips enhance your learning experience
- Web access to an extensive list of additional resources
- Hyperlinks to a glossary of terms used in the course



www.sqe.com

TAKE A FREE DEMO TODAY! VISIT [WWW.SQE.COM/EMTD](http://WWW.SQE.COM/EMTD) FOR MORE INFORMATION.



*You can find a number of my articles published at Better Software magazine, StickyMinds.com, AgileConnection.com and techwell.com.*

## Don Gray

### **DELIVERING CUSTOMER VALUE**

MENTORING AND TRAINING  
for EXECUTIVES, MANAGERS & TEAMS

don@donaldegrey.com  
336.414.4645  
www.donaldegrey.com