BETTER SOFTWARE

The Print Companion to **StickyMinds**.com

Incremental and Iterative Development

HOW they are different and WHY you should be doing both
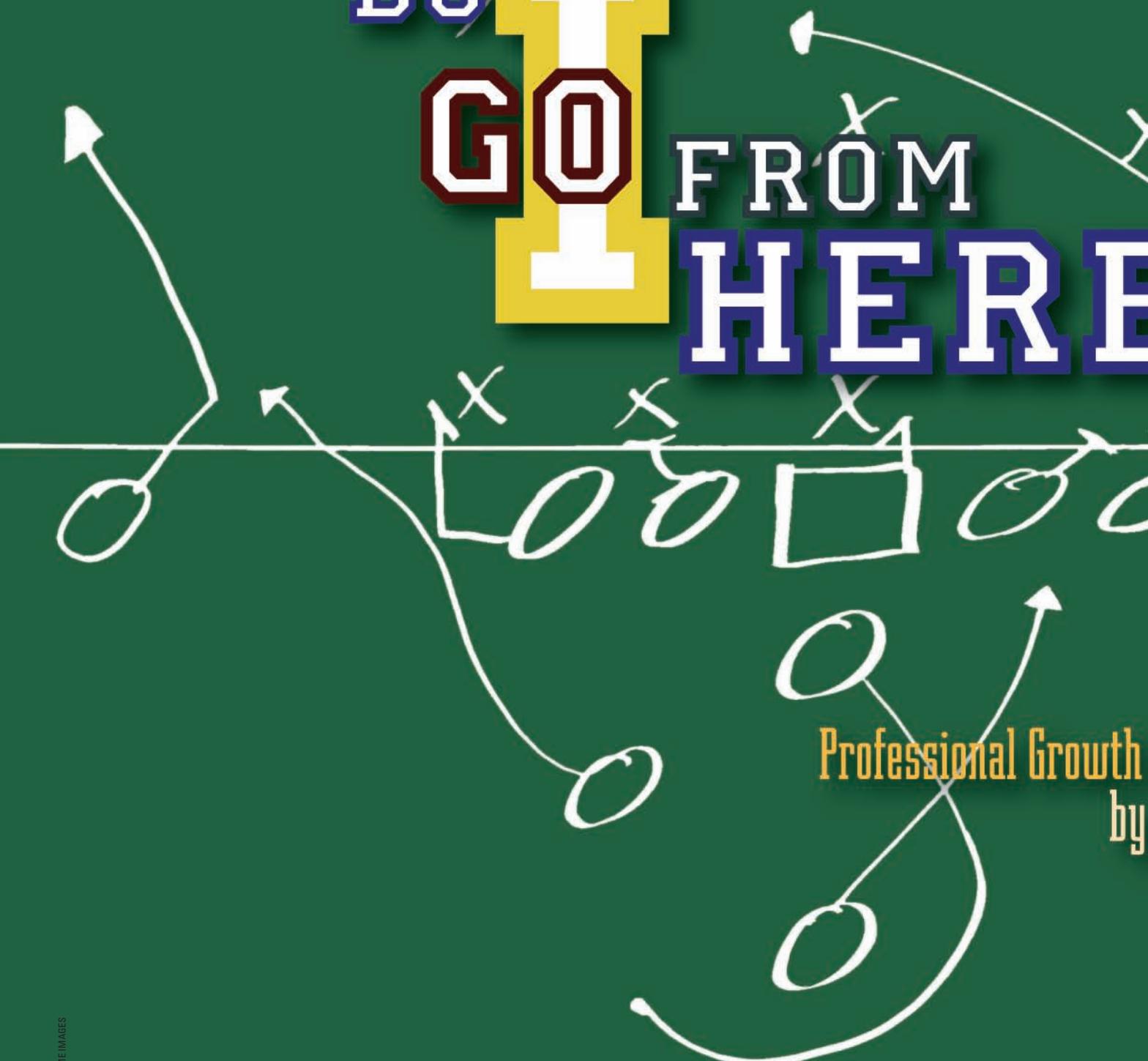
WHERE DO I GO FROM HERE

Professional Growth

by

for Software Testers
Matthew Heusser

I sort of fell into the testing field: I learned testing on the job from some really good people. I used to think that was exceptional, but over the years, I realized that very few people actually plan to be testers. Sometimes people use it as a stepping stone to development, but very few people set out to test as the end goal. Then, once you become a tester, the next step is unclear. The perception often seems to be the choices are stagnate or go into management.

　　　　　　- Ben Simo, Software Tester

Sadly, Ben is not alone in his dilemma. Where accountants and professors have a detailed professional career ladder, in many test organizations the ladder only has one rung. Ben is quick to add, "In my mind, this isn't just a problem for testers, but often for all of IT. Often the IT managers are the least qualified managers in the organization. They chose management as the default, because they saw no where else to go."

Shrini Kulkarni, a popular blog author and software tester, recently pointed out that a management job often leads to dealing with personnel evaluations, invoices, payments, recruiting, budgets, and other aspects of running a business that may not seem interesting to technical people. Many testers prefer solving technology-related problems. Therefore, many testers are more interested in a technical career ladder, but they may not see one with their current employer. If we are responsible for our own professional development, as Alicia Yanik pointed out in the December 2007 issue of *Better Software*, we need to develop our own professional growth strategy. In other words, if your company does not provide a professional development system, one option is to build your own ladder—figure out where you want to be and work with the company to recognize that growth.

ANOTHER WAY TO LOOK AT IT

Software testing is a business, and you are the product. Your goal is to increase the value of the product. There are three common strategies [1] that we can use to accomplish this goal: cost reduction, market focus, and differentiation. In other words: charge less per hour, know your customer better, or be "special."

Charging less per hour may not be feasible, but if you do, then everyone competes in this fashion and the result is a price war. The winner of a price war is whoever has the lowest cost structure.

A market-focused tester limits himself to a particular niche, such as Department of Defense contractors, medical device manufacturers, pharmaceutical companies, and so on. This can work for some contractors and job-hoppers, but if you want to stay in the same company, it doesn't help much. Everyone at your company is a specialist in that field.

The third approach is differentiation. People willingly pay twice as much for a bottle of Coca-Cola than a bottle of the generic stuff—and the same goes for automobiles, laundry detergent, macaroni and cheese, and even hotel rooms. For example, Donald Trump has managed to make his name synonymous with "expensive … and worth every penny." In one episode of his television show, he essentially slapped his name on bottles of water and doubled the price. There are so-called "name brands" in software testing, too—people who have made an effort to be better known than their peers.

This "branding" style of differentiation is reputation and perception, but sometimes the product actually can do something unique and different. For testers, this means specialization, often in a sub-field that your company does not currently have. By making this specialization explicit (defined in our job descriptions), we can demonstrate added value to the company and then build our ladder.

BUT ... WHAT SPECIALTY?

Inventing a completely new specialty can be scary. It not only must be an area in which you are interested but also must be a role that your organization needs filled. But if the organization needs it, why isn't it already being done by someone else?

Jerry Weinberg suggests generating ideas for new specialties by taking two existing things and intersecting them [2]. Figure 1 shows a few examples of roles

Non-Testing Group	Required Testing Expertise	Area of Intersection - A niche to hang on
Network Admin	Zero	Performance Testing
Security Team	Little	Security Testing
Technical Writer	Zero	Documentation for the Test Team
Business Analyst	Little	Analyst/Tester
Developer	Moderate	Automation and Frameworks
Project Manager	Little	Test Estimation and Status Reports
Technical Support	Zero	Triage/Investigation of Support Tickets
DBA	Little	Database Testing

Figure 1: Overlapping roles and possible opportunities

that can be formed by intersections between testing and non-testing groups.

In each of these specialties, we find a real business need that is served by two groups. The non-testing group may not understand or appreciate what your manager needs, or it may not be serving the needs of the testing organization well. Perhaps you just like what that group does and find it interesting. By investing a little bit of time exploring the intersection of testing and other roles, you can find out enough to better fill a need. You may learn enough about the role to fill in when Bob's sick or to have a backup job in case of a layoff.

Once you have identified a specialty, work at it for six months. Get enough experience to have something to list on your annual review. If you really enjoy the work, you might want to think about where that could go in five or ten years and set out some really high-level goals. In order to have a serious ladder, the role will have to carry significant responsibility and be of real value to the company.

FROM CONCEPT TO CASH

Deciding on a field will give you a concrete direction. While having a direc-

tion is great, do you know your long-term goal or destination? Is it clear and concrete or just warm and fuzzy?

One way to develop longer-term career goals is to work backward from an envisioned future—or, in the words of business guru Stephen Covey, "Begin with the end in mind." For example, imagine that some coworkers and friends are discussing you around the water cooler five or ten years from now. What would you like them to say about you? What role are you in? While "Where do you see yourself in five years?" might be a clichéd interview question, "Where do I want to be in five years?" is a much more serious one.

Once you have goals that are specific and realistic, you can time-box them—which is as simple as creating your own deadlines to provide motivation. With time-boxed goals, you can create subgoals, or tactics, that are smaller and smaller until the pieces are manageable enough to accomplish. We do this every day in testing and project management; it really isn't that hard to apply to career growth.

If your goal is to be the go-to guy, you might sit down with your manager

and ask for additional responsibility on a highly visible project. If your goal is to be a developer/tester, the first step might be to go to the library and check out two or three books on development. If your goal is to earn a college degree, the first step might be to find a program that is a possibility, and then take a single college course.

Along the way, your goal might change. You might find that you dislike the pressure of constant delivery, or that college course might lead you to a second career as a journalist. The idea is to do something that moves you in the right direction and then adjust your plan as you go. If you are an "agile" type, you could think of each of these steps as an iteration. Each step provides you with value, feedback, and—because the iterations are frequent—the opportunity to adjust.

LOVE THE ONE YOU'RE WITH

If your company doesn't have a technical career ladder, it is possible to develop one, especially at larger companies with an HR department that employs a compensation analyst. Microsoft, for example, has a defined ladder for every technical role, and the higher levels are well compensated. Alan Page, a test architect in Microsoft's Software Engineering Excellence group, explains that "The split between management and non-management typically happens around the 'senior' level. For example, for software development engineers in test (SDET), we have SDET I, II, senior SDET, and principal SDET. A principal SDET may also go by the title 'test architect.' Thus, our newest managers have several years of testing experience, and people who choose to remain technical are given a few more steps on the ladder to climb."

Many companies can stand to benefit from defining detailed career paths. For example, consulting companies can charge a premium by offering higher-level consultants, who have more perceived value. If you work for a company

like this, you can demonstrate how career pathing can make the company more money. At the same time, a professional ladder will mean more money for those people willing to climb to the top. (Hint: That's you.)

THE CAREER PATH LESS TRAVELED

Without a non-management career ladder, testers have a false dilemma. Again, Ben Simo tells the story: "I knew one excellent tester who was pushed into management, and he hated it. The truth was that he wanted to do excellent testing, not personnel evaluations and staffing with Excel spreadsheets. So, after some time, he convinced the company to create a test architect role."

If your company doesn't have such a path, it is up to you to carve one out for yourself. One integral part of any career management plan is marketing. As odious as that sounds to many technical people, no matter how much you know about software testing, if other people do not realize it, then no one will be knocking down your door asking to promote or hire you.

The StickyNotes for this article are full of ideas that will help you grow professionally and grow your reputation. The challenge of this kind of growth is that you want the improvement to be real, and you want the credit for it. That is the "marketing" part. To get that credit, we may need to improve in areas beyond hard technical skill.

One simple model for this is to think of inside skills (coding, exploratory testing, test automation) and outside skills (verbal or written communications, giving and receiving feedback, or supervision). While we don't have time in this article to explore all these areas, I feel obligated to mention that the gap between your current skills and where you want to be might be technical and heads-down, or it might be how you interact with the world. In sales, management, liaison, and consulting roles, outside skills can far outweigh inside skills. Even the best technician is stuck if he can't work with people or communicate his value.

Getting the credit for our work can be very challenging. When I interviewed John McConda, a performance tester for Mobius Labs, he pointed out that it is very hard, if not impossible, to put "I read a book" on a résumé. Even if you read and applied the ideas in the book, it can be hard to take that credit. John's suggestion is to learn new technology, then find opportunities to use those technologies in the workplace. Or, if you enjoy reading, then form a book club, host a brown-bag session, or champion some other form of distributed learning. Then not only can you claim it on your résumé but you also are building personal relationships, which are the best kind of recommendations.

The biggest, most positive career moves I ever made were in building things. In 1998, I founded the West Michigan Perl User's Group ("Perl Mongers"), and in 2006, I co-organized the first Great Lakes Software Excellence Conference (GLSEC.org). These led directly to jobs and interviews from hiring managers.

Building events and organizations follows the general pattern of giving things away. When I say giving things away, I don't mean a pen or a key chain, but rather ideas and your time. Start a blog or a brown bag. Learn a technique, do it, then bring the rest of your team along. It may take time, but people will notice, and when it's your time to ask for help, they may be in a position to help you. My advice is to pick two to four things that provide the best return on investment and pursue them aggressively.

TOMORROW

If your organization has a professional development ladder, it is probably vague by design. Sit down with your manager and find out specific, measurable goals to demonstrate accomplishment of the next level. If there is no ladder, the first part of the conversation is how to create one. If the organization balks, you can figure out where you want to be, what to do to get there, and how to make sure the effort is noticed. Earlier I mentioned redefining your role in a way that adds unique value to the company. After a few months of effort, you can ask that this new role be designated "tester II," "senior tester," "performance test specialist," or so on. There are several Web sites that recommend how to redefine

NETWORKING

Networking is an extremely common practice, but it amazes me how it is so poorly understood. Yes, you can go to meetings, collect business cards, and aggressively pursue relationships with people who are in a position to help you. It is even possible that this process works! It just feels a little mercenary to me. I have colleagues who "network" this way, and when I answer a call from them, the first question out of my mouth is usually something like "What do you want this time?"—only a little more polite.

When I think of networking, I think of the opposite. Instead of trying to get, I try to give—finding out what I can do to help the other person. For job seekers, that might mean an introduction. For volunteer organizations, that might mean stepping in and filling a role. At conferences, it can be as simple as listening and offering advice or, even easier, listening and offering sympathy.

A few of my colleagues also network in this way. When I receive a call from them, I am quick to answer it, because I know the conversation will be about an opportunity that is to my advantage.

Now, when 90 percent of our phone conversations are to my advantage, and the one time comes when you need my help … that is when the professional network you've built can come to your aid.

your job or role formally in order to advance technically; I am suggesting that you actually do it informally first, so you'll have a leg to stand on.

I would go so far as to suggest that additional work beyond forty hours a week should be invested in you, not project work. Stephen Covey refers to this as "sharpening the saw"—doing personal work to make your professional work go faster. This could be contributing to open source projects, experimenting with new technologies and techniques, or just keeping up with the software testing literature.

This research and experimenting can create some real options beyond going into management, which brings us back to our original dilemma and my colleague Steve. With nearly thirty years of software experience, Steve has worked for a large government agency, and multiple Fortune 500 companies, and earned a master's degree from Johns Hopkins University along the way. In his words, "The choice to stay technical is not a quick and easy way—in fact, you have to build your own way. Staying current is hard. I strive to add something to the résumé every six months, live a thrifty life, save and invest some of my income, maintain relationships, and contribute a little bit to my peers and my community."

Steve is quick to add: "There is nothing wrong with going into management, but my choice has been to avoid it. The problem is when you make the decision by default, or let someone else make it for you." **{end}**

REFERENCES:

1] Porter, Michael J. *Competitive Strategy*. Free Press, 1998.

2] Weinberg, Jerry. *Becoming a Technical Leader*. Dorset House, 1986.

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

- Professional resources

USER GROUPS AND PROFESSIONAL ASSOCIATIONS

Working in technology— especially software testing—can feel isolating, but believe it or not, there are people near you who struggle with the same thing. Every large metropolitan area (and most medium ones!) has formal user groups that meet regularly. User groups typically meet monthly, at lunch or dinner meetings; some may even have an annual conference or other large event. These organizations are always looking for board members, speakers, and attendees.

If you aren't the "toot my own horn" speaker type, there are a number of non-glory jobs like secretary or program chair. People appreciate and notice these contributions, especially when they are done well. Some of my strongest relationships—and a job or two—came through working as a volunteer for my local Perl user group and Extreme Programming group.

Here are a few large, national groups that may have a chapter in your area. If there is no user group in your area, the easiest way to become a recognized "name" in town is to start one.

- The Agile Alliance is an informal group that specializes in agile development. **www.agilealliance.org/show/1641**

- The Association of Information Technology Professionals (AITP) leans more toward hardware and networking, but each chapter has its own flavor. (The AITP was originally the Data Processing Management Association, so it has been around for some time). **www.aitp.org/index.jsp**

- The American Society for Quality (ASQ) has chapters all over the United States, and includes representatives from all aspects of business and government. **www.asq.org/sections/find.html**

- The Quality Assurance Institute (QAI) is a very software-focused version of the ASQ. **www.qaiworldwide.org/chapters/index.html**

- The Association for Software Testing (AST) is dedicated to software testing and just began a local chapters program. **www.associationforSoftwareTesting.org**