Case Study: Beefing Up Software Developer Skills

How Maritz Solved The Developer Skills Problem

by Dave West
with Mike Gilpin and David D'Silva

EXECUTIVE SUMMARY

The skills of the software development team are crucial for project success and can enable or doom the adoption of any new architecture or technology. Maritz understood the value and importance of building the right skills within its development organization as it transitioned to .NET. The company aligned its skills road map with its technical architecture and exploited a new approach to learning provided by InnerWorkings. This approach combined eLearning, community, and competition within the developers' integrated development environment (IDE), making education easy and fun while moving toward clear organizational goals. Application development professionals can learn from this innovative approach when facing a similar need to build new skills within their organization.

SITUATION: MARITZ NEEDED TO EXPLOIT NEW TECHNOLOGY WITHOUT NEW STAFF

Innovation and change drive the technology industry. Yet organizations walk a tightrope when choosing whether to learn and apply new technologies, with their inherent risk and cost, or to maintain the status quo. There are many risks and costs associated with new technology adoption, but one of the most fundamental ones is skills. All technology, new or old, requires practitioners to understand the technology and then effectively apply it to the problem at hand. Maritz, like any company facing this challenge, wanted to exploit new technology without reducing effectiveness or increasing costs to the organization. "We understood the value of our staff and wanted to ensure that we both invested in their development and built an IT organization that was ready for the future," said a manager of the productivity tools and process (PTP) team.

The Traditional Approach To Skills Adoption Was Just Not Working

Acquiring skills is more than training, requiring that theory and practice be combined in the context of motivated staff working in feedback-rich environments. Most IT departments approach skills development by combining classic HR-driven staff development with haphazard individual approaches. HR regularly reviews skills needs to define a set of skills requirements and then finds education materials and vendors to fulfill those needs. This process is lengthy and often focused on *business* or *soft* skills because they are the most widely applicable. The haphazard individual approach happens when one application development professional sees something that he or she considers to be of value and invests time in either learning the materials or building a case to attend a conference or training class. Large technology vendors plan this time and cost into the budget for each engineer, but in most companies, this is neither planned nor budgeted. Maritz wanted to invest in the development of its staff in a planned and structured way, which the manager of the PTP team described as "a systems approach to skills management." But this approach was constrained by:

- **No time for training classes.** As at any organization, immediate needs, projects, and client support get the attention of the development team. There was always something business-critical that would prevent the development resources from attending classes. Furthermore, most classes required a block of time that would not fit into the schedule of project needs.

- **Online sources inconsistent with Maritz standards.** The World Wide Web provides a massive set of resources and materials. Developers armed with a browser and Google can locate large amounts of content. A heavy user of Microsoft technology, Maritz could also exploit the materials of the Microsoft Developer Network (MSDN). But although plentiful, this content varied in its technical depth and applicability. The architectural community at Maritz was also concerned that the approaches described were often contrary to their architectural standards.

- **Different learning speeds and preferences for different people.** Traditional classroom-led training requires practitioners to not only be available for the training but also be suited to that speed and delivery approach. Maritz had many different types of people working within the teams, all with their unique blend of training needs.

People With The Right Skills Were Never Available

Maritz had a significant investment in people and technology. It was important to Maritz to exploit the business and technical knowledge of these resources while bringing all developers up to speed on new technologies and approaches. Effectively balancing the needs of today with the needs of tomorrow was a key challenge that Maritz faced. This challenge was compounded by the following issues:

- **Developers need to use a mix of programming models.** Traditional boundaries between technical platforms or runtime environments were blurring as teams focused on delivering business value to their customers. This meant that Java developers needed to learn .NET or database resources had to help build data access services in C#.

- **Contract resources never know the business.** The use of contract resources to fill the gap between short-term and long-term skills needs seemed like a great solution, but Maritz found that although they were trained in the new technologies, contract resources required lots of help with business knowledge.

- **Bringing in external skills is expensive.** A heavy dependency on external skills also had a negative impact on the budget of the department, as those skills were expensive — and the greater the skill level the more expensive.

INTEGRATED WEB-BASED TRAINING AND ASSESSMENT ENABLED SUCCESS

Maritz decided to move away from a traditional approach to training and the use of contract resources and embarked on an eLearning, community-oriented strategy. The objective was to tie technical strategy to career development and create a program that both moved the company toward a new technical platform and created an environment for innovation and skills development. Maritz worked with InnerWorkings, a provider of eLearning, collaboration, and assessment technology, to build out a program that fulfilled these objectives.[1]

Competition And Judging Led To Low-Impact Assessments

Maritz wanted to start with a good understanding of what skills were in place and what areas needed development but was concerned that traditional approaches to assessment never really provided the detailed feedback about staff. The PTP team manager said, "Assessment is key, but most traditional approaches to assessment don't seem to work." Instead of using surveys and interviews, Maritz looked at the technology provided by InnerWorkings and its ideas of competition and code-judging as a mechanism that could quickly generate an understanding of team capability. By picking competitions that aligned to technical strategy, Martiz would see how equipped the teams were for tomorrow. Maritz picked this approach because:

- **Competition makes assessment fun.** Traditional assessment approaches are survey- or interview-based, asking questions about a practitioner's skills and experiences. To obtain detail requires a large number of questions, which are difficult to answer. For example, asking respondents about their knowledge of .NET framework requires numerous questions to assess each capability and gather experience. Replacing this laborious approach with one based on code competition, integrated into the IDE, makes it possible to assess many aspects from a single code example.

- **Transparent scores increase company knowledge.** Although they are not yet fully deployed, Maritz is excited about the prospect of providing dashboards that show the development organization's capabilities. Maritz hopes that these measures will provide the basis for improved company knowledge and act as a motivator for the teams.

- **Aligning competitions with technical strategy supports change.** By carefully selecting the right competitions, leaders can drive the overall technical strategy as the teams develop skills associated with particular technologies and experience with key design patterns and working practices.

Certification Provides A Road Map To Development

Maritz is not interested in certification for certification's sake but in using certifications as a skills road map for developers. Each certification fits into a broader curriculum, which will move individuals toward the long-term technical goals of the organization. The benefits of certification include:

- **There is no need to develop a skills road map.** Each certification program includes detailed curriculums that incrementally build skills and gather experience. Traditionally, Maritz would have to spend a great deal of time gathering needs, looking up courses, and defining a skills road map. Via InnerWorkings and the certification program, Maritz avoids that work.

- **Developers are more motivated.** As they add these certifications to their résumé, developers will demonstrate skills in a certain technology. These skills are transferable and thus improve the overall marketability of that developer. This is a strong motivator for developers to make time outside of normal work hours to become certified.

- **Maritz becomes a more attractive place to work**. Hiring the right staff will continue to be more challenging as IT continues to grow.[2] Because it provides certification and continued learning, Maritz will stand out from the crowd.

eLearning: Better Than Traditional Models

Traditional approaches of weeklong training classes, self-study, books, and Web resources such as MSDN add value but are hard to integrate into work life or are expensive. eLearning provides a more flexible approach to training.[3] The benefits of eLearning to Maritz include:

- **Location, location, location.** Developers can access the content from their workstation at the office or at home. Because Maritz does not have to send staff away from the office and pay for travel and living expenses, this reduces the cost of training classes. Also, by integrating eLearning into the tools developers use every day, developers can reuse materials and learn in the same tool configuration that they use daily.

- **Flexible timing makes it easier to blend training with work life.** Finding an hour a day is much easier than finding a week. With a flexible training tool, developers can quickly pick up and complete a module when the time suits them.

- **Training in context makes it easier to get help.** By sitting at their own desks, developers can call on all the same resources they use for their normal work, including asking for advice and comments from colleagues. A remote community hosted by InnerWorkings augments this local community. These community members provide expert help and support via instant messaging.

RECOMMENDATIONS

COMBINING eLEARNING, COMMUNITY, AND COMPETITION IS THE FUTURE OF TRAINING

For technology-oriented skills training, the delivery of eLearning within the IDE is a very compelling solution. By adding community and dashboards, Maritz has motivated its developers to actively participate in their own skills development. The result is a group of practitioners who are motivated to learn and an organization that is effectively measuring that capability over time. To emulate Maritz's success:

- **Integrate technical strategy with training strategy.** Maritz considered the skills of its development teams to be a fundamental part of its technical strategy. Combining the rollout of its strategy with development of the associated skills helped ensure a successful implementation of that strategy.
- **Review your current technology and see if integrated eLearning is possible.** Developers love to live in their IDE. By integrating eLearning into the IDE, developers could easily switch to training resources. It also meant that the experience was consistent with their normal development experience, thus allowing a seamless transition from training to work and back again.
- **Think about combining community with education.** Being able to ask questions during and after training adds a lot of value to the education, often providing real-world examples to support the training examples. Social media can amplify that support network by providing access to people who are not local. Software development professionals should add some element of collaboration to both electronic and face-to-face training, supporting attendees during and after the training session.
- **Introduce competitions to encourage participation.** By exploiting the natural competitive nature of human beings, software development professionals can be better motivated to learn new technology.
- **Measure usage and build a dashboard.** If people are a primary resource, you must provide regular statements as to their skills and gaps. Introduce a system-based approach to education so that you can instrument the processes and gather data.

ENDNOTES

[1] InnerWorkings is the vendor used by Maritz to support the rollout of an eLearning toolset. InnerWorkings offers an integrated eLearning tool and associated services to support the optimization of development skills and processes. Details can be found at http://www.innerworkings.com/.

[2] For the challenges of workforce planning and the effect of shifting populations, see the October 26, 2009, "Global Workforce Planning Through 2016: How Population Shifts Will Affect The Supply Of IT Skills" report.

[3] Ruth Clark and Richard Mayer discuss the value of eLearning and whether it is better or worse than traditional approaches in their book, *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. Source: Ruth Colvin Clark and Richard E Mayer, *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*, Pfeiffer, 2007.