

SOFTWARE TESTING SPECIAL

CIOReview

The Navigator for Enterprise Solutions

OCTOBER-09-2015

CIOREVIEW.COM

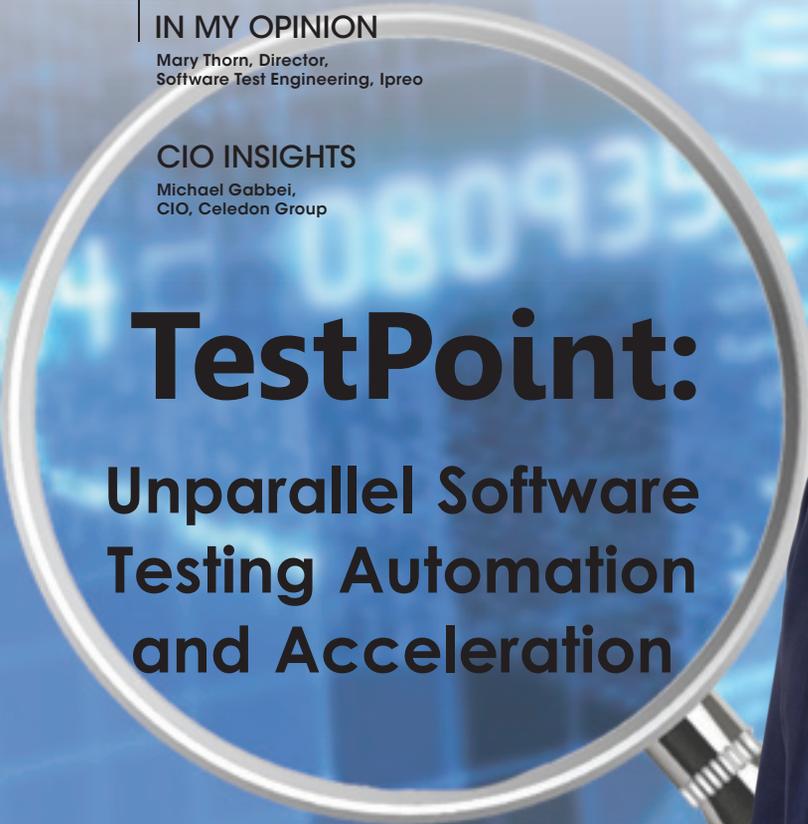
+

IN MY OPINION

Mary Thorn, Director,
Software Test Engineering, Ipreo

CIO INSIGHTS

Michael Gabbei,
CIO, Celedon Group

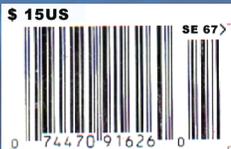


TestPoint:

Unparallel Software Testing Automation and Acceleration



Fadi Hawli, CEO



CIO REVIEW
44790, S Grimmer Blvd.
#202, Fremont, CA-94538

To Really Know How Things are Going, Ask a Tester

By Jeff Dalton, President & CEO,
Broadsword Solutions Corporation



If you ever really want to know what is going on in your product development organization, ask a tester. As consumers of distorted vision, weak requirements, poor design, and sloppy code, they are at a unique position to identify where defects really originate.

Building great technology products is about more than just code. In fact, it's hardly about code at all. Yet most books, articles, and conference speeches about improving software quality focus on testing tools, techniques, and automation. While these are all good and necessary discussions, we have thus far fallen short of reaching the goal line: consistently building high quality products that delight our customers.

Isn't it About Time We Add Something New to the Discussion?

As a CMMI Lead Appraiser and Agile coach, I am

often asked to assess the capabilities of product development teams, and I can usually ascertain strengths and weakness within fifteen minutes—if I start with the test team.

Great software is an ecosystem born from a glimmer in someone's mind, matures into needs, grows into requirements, transforms into designs, manifests itself as code, all the while being validated and verified, until it finally emerges to delight and satisfy our customers.

Instead of starting the discussion with tools, take your first step toward building great products by answering my Twelve Questions of Wildly Successful Agile Teams:

► **What is the Product Vision?**

Product vision exists at multiple levels: company, product, team, and individual. Agree upon the product vision and write it down. You'll need it later when you're pressured to cut corners after everyone forgets why they're building the product.

► **How will we Work?**

Working without an action plan introduces expensive process defects that cause delays, re-work, and

frustration for the entire team. The bigger your project is, the more important it is to write this down.

► What do we Need?

The answers to question number two will drive the need for a resources, facilities, and tools to develop a great product. Lack of proper resources introduces instability in the system, and instability leads to poor quality.

► Who is Responsible?

Building great products requires performing thousands of tasks, and while there is power in the agile self-subscription model, there are plenty of tasks that need a point-person.

► Do We have the Skills?

Great products require more than just great coders, and we'll need strong product managers, product owners, architects, and program managers in addition to our core development team. It's not enough to be great at c# or Java, weaknesses in other critical roles will lead to defects in requirements, user stories, architecture, planning, and estimation.

► Is Intellectual Property Being Captured?

Beyond basic code management, there are data that represent the brainpower behind your product, and it's an investment that needs to be preserved in writing. Anyone who has ever tried to maintain code without proper documentation knows how easy it is to unintentionally inject defects that could lurk untested for years.

► Are the Right People Contributing?

In my work with agile organizations, I've noticed that product owners are rarely customers, and when they are, they barely participate in projects. End-to-end collaboration is critical for uncovering the most insidious and costly defects, and without it, we leave a twenty-something coder to make functionality decisions for a billion dollar company.

► Is our Plan Working?

It is human nature to seek efficiency and "repeatability" which we believe delivers stability, but it also allows recurrent defects. Take advantage of short sprints to capture data until you find a pattern that you know produces results. Keep your plan updated so everyone understands the baseline.

► Are we Using the Plan?

How we work is the most important factor in producing high-quality products, but if it's only a paper exercise, say to satisfy your manager, an ISO auditor, or the federal government, you're missing out the lowest cost, most fundamental action you can take to improve

product quality. Analyze the results of compliance self-evaluations and share them with the team during retrospectives.



Jeff Dalton

► Is Management Informed?

In too many companies product teams are firewalled from management, with little or no interaction between the two. Sometimes management doesn't know, or seem to care, if the teams are taking proactive steps to improve the quality of the company's products, and teams have little motivation to take action. Encourage collaboration with management by using the data produced by answering questions seven through nine, and seek their assistance improving the results and aligning them with the organization's business strategy. Misalignment is the most costly and difficult defect to correct, but it can be solved.



Testing and test automation tools, as critical as they are, are only two tactical pieces of the puzzle on the road to great software

► Do We Know When to Change?

Fine-tuning the plan is like adjusting the knobs on your car's radio, except instead of one volume knob, you have hundreds. If you turn everything up to eleven, it just doesn't sound good (at least not for long), and different musical styles require different settings. Not everything in your plan has to be high-intensity and rigorous. Sometimes a simple code review is called for, and other times a Fagan Inspection is the only thing that will uncover dangerous defects. Define some guidelines so you know what to do, and when to do it.

► Are We Always Improving?

I tell my classes that "those who always improve, win." Build your plan to include the identification of performance defects that resulted in re-work, chaos, confusion, frustration, poor quality, project delays, and more. Bake them back into the plan and share them with the rest of the company.

When I was a developer, I was obsessed about passing the test, and there was nothing I loved more than un-wrapping a shiny new tool. But testing and test automation tools, as critical as they are, are only two tactical pieces of the puzzle on the road to great software, and their effectiveness would be vastly improved by building a performance strategy that eliminated the most serious defects first by answering the Twelve Questions of Wildly Successful Agile Teams. 