



Quardev Monthly

Volume 4, Issue 6

June 2010

Welcome!

Welcome to the June edition of the Quardev Monthly!

We look forward to sharing insights and helpful information in the areas where we know a thing or two - testing, quality assistance, technical writing and documentation, project management, and consulting.

In this month's edition, we have a guest contributor, Geordie Keitt, a tester from Prochain Solutions in Virginia who shares

some thoughts on Elliot Jaques' theories on cognition and how they relate to his daughters and his subordinates.

Also, Jon Bach shares a speech he gave as the commencement speaker at a local technology school graduation.

Enjoy the newsletter with our compliments and please contact us with questions, comments, or article ideas.

-The Quardev Crew!!

Requisite Organization Theory

By Geordie Keitt from [ProChain Solutions](#)

Introduction

Children and their parents can each solve problems, but children cannot solve problems that are nearly as complex. Young adults starting out on their first job, and CEOs of major corporations each solve problems but the CEOs are generally harder.

Some testers can make up very intricate tests or tests that cover great areas of the application, and other testers can imagine tests in a narrower problem space or of less investigative power. One explanation of all these phenomena, according to Elliott Jaques' work on Requisite Organization Theory, is that the cognitive power people bring to their work differs in well-defined ways, grows throughout their life, and affects the amount of work we can expect them to accomplish.

Here is a brief introduction to this aspect of Jaques' work.

Cognitive Levels

If you have observed very young children begin interacting with the world around them you will notice the progression they make in the complexity of their playthings. Initially babies are able to focus on one object at a time and choose which block to play with. Soon they are able to play with two blocks and make them interact. Later they can make a stack out of blocks, creating a structure that relies on work they do over time. Around the time they start becoming verbal they can create a physical system that relies on multiple stacks linked with a capstone, such as a bridge or an arch.

These four stages of play can be described as "Disjunctive" (separate items processed separately), "Cumulative" (separate items used together), "Serial" (items depend on each other to create structures), and "Parallel" (structures rely on one an-

(Continued on page 2)



Requisite Organization Theory, Continued

(Continued from page 1)

other to complete a superstructure). These descriptions map exactly to the four logical operations: OR (A or B), AND (A and B), IF-THEN (If A then B), and IF-AND-ONLY-IF (if A then C, but only if B as well).

As the child gets older the natural process of cognitive growth recurs, with words and symbols instead of concrete objects. She learns many words, ideas and instructions, and expresses one at a time at first. Let's see what this sounds like as she creates an argument: "I want a pony." "Why honey?" "Because I want one. They are pretty. Jenny has one. I see them on TV." The points of the argument are disjunct or separate from one another.

Let's revisit her argument at the next cognitive level, Cumulative processing. Now she groups her arguments atop each other. "I want a pony." "Why honey?" "Because I want to be a vet when I grow up, and having a pony lets me learn about animals so I can be a better vet. I already have a dog and a cat and a hamster and goldfish so it's time I learned about bigger animals." The points of the argument accumulate. (Full disclosure: I have 2 girls.)

Further cognitive growth will allow her to argue using Serial processing. "I want a pony." Why, honey?" "If I get a pony then I will have an interesting activity to put on my college application, which will get me into a better college, so I can earn more money to support you in your old age, Dad." One can see the chain of logic with each link dependent on the one before.

If you are lucky enough to have a daughter ask for a pony using Parallel processing, it might sound like this. "I want a pony, Daddy." "Why, honey?" "Well, I could tell you all about how wonderful it would make me feel to know you love me so much. Or I could tell you it's a great investment since a pony will only lose value if it dies, and since I'm going to Bowdoin on a pre-med scholarship with a concentration in veterinary medicine I have a vested interest in displaying aptitude with livestock. But my real reason for asking is because I know you want to

spend more quality time with me before I leave for college, and we can take care of Betsy together. I'm really just concerned about you, Dad."

Elliott Jaques categorized these various levels of cognition and numbered them: Symbolic Disjunct is Level 1, Symbolic Cumulative is Level 2, Symbolic Serial is Level 3, and Symbolic Parallel is Level 4. Levels 5, 6, 7, and 8 again recapitulate the mind's progression up the logical chain, but in the realm of abstract concepts with no tangible analogs. This is the realm of systems thinking. A Level 5 thinker understands a single complex system. Level 6 understands the intersection of several systems, possibly of global scale. Level 7 imagines new global-scale systems and works to implement them. Level 8 replaces the current systems of the world with ones of their own design. Jaques cites Einstein and Churchill as prototypical Level 8 thinkers.

Presumably there are further levels, but no experimental evidence exists to suggest what they are.

About Geordie Keitt:

Geordie Keitt has been testing for 15 years, and managing testing for 10 years. Keitt says, "I test with my brain, with my hands, with my ears and with my eyes – and I expect any testers I manage to do the same."

Keitt's mission: getting problems fixed so people have a great experience with his software. He has acquired and developed an arsenal of test tools. As a manager, Keitt develops custom test status reporting tools and methods to suit his context. He helped beta-test ChainFrame, a Selenium-based automated Web testing tool. Keitt has a healthy number of friends and mentors in the context-driven, small-"a" agile testing community.

Some Thoughts for Graduates

By Jon Bach, Manager for Corporate Intellect, Quardev, Inc.

Article Note: The following article is an actual speech that Jon Bach gave as the commencement address at a local technical college.

For further reading that may be fitting for graduates, please see two of Jon's papers: [The Expert Game](#) and [When You're Tested](#)—both on the Quardev Website.

—

I'm not sure I should be here.

I mean, what qualifies me to speak to graduates in the technology industry?

I have a journalism degree from the University of Maine. Maine - wayyyyy over on the other side of the country from a state who's mascot is the entree at Anthony's Seafood restaurant.

Does that make me qualified?

Well, because I'm named after the soaring, learning spirit of *Jonathan Livingston Seagull*, I might be qualified. It's a book about a seagull who struggles with passion for flying and a thirst for knowledge despite adversity - an inspiring story to tell in a commencement speech.

But wait, I didn't write it. My dad did. So am I qualified?

Well, like my father, I am also an author. Ok, that's pretty good, the book was a lot of work, I learned about life, some people know my name... great.

But wait, the book I wrote was not technical. It was an autobiography - about when this famous author dad of mine left my family when I was two years old. It's a very personal story with a happy ending - great, wonderful, awesome.

But that's not why they asked me to speak to you today. They asked me because these days, I'm a technical manager for a software testing company in Seattle.

THAT IN ITSELF shouldn't qualify me. I could be a clueless, incompetent manager for all they know!

No, I'm here because someone heard me give a guest lecture to a class here. It's one of my favorite things to do. Someone saw the way I interacted with the students, saw my energy and enthusiasm, saw my sense of humor, my technical presentation ability and thought I'd be a good fit.

But even there, they're taking a chance on me. I never gave them a resume, I've never worked with them on a project.

So they might not know that this past ten years of software testing career, started out as data processing. They might not know that before that, I was a clerk at Borders Bookstore, and before that, I was washing dishes in Tacoma.

And this was AFTER my journalism degree!

(So now that I've totally depressed you about your chances if finding a job in your field...)

No, I'm going to bet that the people who asked me to be here don't care about that stuff.

That data processing job required me to use software, and I started finding problems. And I reported them, and then I got good at it, and then they called me the TESTER, and I said, that would be cool to be paid to do nothing but find problems all day, and someone said, yes, that's a whole career, it's called software testing, and I loved it, and I still do.

So, one day someone convinced me that I was ready for the major leagues - the biggest software development company in the world - Microsoft - right here in our backyard. And I was nervous as hell and I interviewed and I failed. And I failed again, and failed 5 more times in other interviews. And just when I was convinced that I was not good enough, I interviewed one last time.

This time, I told them I had some limitations, but I had solutions for them. I told them not only what I had done in the past, but why I did it and how I failed the first time, and on and on. I even got up from my chair and went to the whiteboard to answer a question graphically. For the first time, I thought like the interviewer. Is this guy

Some Thoughts for Graduates, Continued

(Continued from page 3)

good enough to take a chance on? Does he have what it takes?

And I got in. It's been a career that has enabled me to meet the best minds in my industry and win their respect enough to have them eventually call me a colleague. It's taken me overseas as an invited speaker on the science of exploratory testing - my area of expertise.

How did this happen?

I love finding problems. They say testers love to break software, well I don't break anything. I FIND THE BREAKS THAT ARE ALREADY THERE.

Think about that for a second.

A lot of you in this room, perhaps everybody, has felt like a failure at one point or another. At the very least, we've all felt tested in our lives.

TESTED. It's a word that conjures fear in some, especially in this room. But to me, it's simple. It means, are expectations being met?

In fact, we're all testers. You're testing me right now.

You know from the early part of my speech that I'M testing me right now. Am I good enough to talk to you? Do I have something meaningful to say?

Testing is about whether something meets expectations. It's a measure of sufficient benefits against critical problems and helping stakeholders make a decision about value.

And in this room, today, there is none of that failure stuff. Faculty decided that for you to be here right now, you are good enough, all of you, and you they have called you graduates!

The world calls you graduates! You have GRADUATED!

You have passed the tests others have set out for you.

YOU... HAVE... PASSED THEIR TESTS.

Now I dwell on failure, like I said, so I confess, I'm a little out of my element in this room filled with so many symbols of SUCCESS today. The graduation cap and gown is an icon - a message to the world that you have made it through. No one can take that away from you.

Yes, there are those who will tell you that the TESTS don't stop. And, actually, let me be one of them now. But don't despair, because I have a formula that might help.

You will meet someone like me soon, if you're looking for a job. Because I am also a hiring manager for the testing company I work for. In fact, in my fifteen-year career I have interviewed over 500 people, many were right out of school.

But when you get to the next stage, whenever and whatever you decide that is, even if you're already working - you have problems to solve - you will be tested. And ladies and gentlemen, this is LIFE. You know this already.

But I'm here to say that I think I have an antidote, or at least, a way to cope with the stress of FEELING tested.

It goes like this. It's like a program. (Some of you have studied software development, I've seen some of your resumes, so I know...)

Anyway, in programming there's a concept known as an If / Then statement. If condition A is met, THEN do instruction B. The code is compiled, it runs, it looks for the condition, and executes some kind of algorithm when it finds the right condition.

But in my career as a test manager, I've learned that humans are programs, too. We are only as good as our programming. We get tired, nervous, moody when certain conditions are met....

So I offer this snippet of code to add to your existing programming in case those conditions occur. This is that If-Then statement:

Some Thoughts for Graduates, continued

(Continued from page 4)

IF:

you are able to get important things done

you are seen learning things on your own

you are seen trying to do things even though
you don't know how

you don't hide your ignorance, but also don't
rest on it

you share freely the things that you know

you honor what other people know

you know more often than not how to find out
what you don't know

you know how to ask for help

you offer help to people ON THEIR OWN
TERMS

THEN

no one will care whether you succeed by learn-
ing or succeed by already knowing,

no one will care if you mess up occasionally be-
cause they assume you learn from it,

no one will care if you forget (or don't know) any
given fact or method at any given time,

and you will be treated as if you're smart and
useful, even though everyone knows you have a
lot to learn.

This is a formula that has been true for me, a
journalism major, author's son, and autobiogra-
pher philosopher who is now an internationally
recognized expert in exploratory software test-
ing.

I don't pretend to know it all, I just know what's
true for me. And as my father said, if you speak
what's true for you, you give others a chance to
agree. But what he didn't say, and what I think
he meant by saying it, is that, in that agreement,
we find a measure of our success. We find com-
munity. And more importantly, we can feel good
that we will have passed the tests that others
set out for us, including ourselves.

One day soon, you may be giving tests as an
interview-ER on the other side of that confer-
ence table. You will be the one with the clip-
board, interviewing a graduate.

And IF so, THEN do me this favor. IF you see
them doing some of those IFs, THEN give them
a chance to show you what they can do.

IF you do that, THEN I will be honored to call
you colleagues of mine, just as I am honored to
stand before you today on this great occasion.

Thank you for listening, and congratulations!