

Enterprise Computing Community Conference 2011

Career Paths in Enterprise System Testing



Authors

Tara Astigarraga – IBM Tucson, AZ

Michael Browne – IBM Poughkeepsie, NY

Eli M. Dow – IBM Poughkeepsie, NY

Christina Lara – IBM Tucson, AZ

Michael S. Phay – UMES Princess Anne, MD

Dorian Thomas – UMES Princess Anne, MD

Maria R. Ward – IBM Austin, TX

Nicholas Wiggins – UMES Princess Anne, MD

Developing T-Shaped People



Economics and Social Sciences

Business Anthropology

Organizational Change & Learning

Business and Management

Science and Engineering

Industrial and Systems Engineering

Computer Science & Info. Systems

Math and Operations Research

What is Enterprise Computing and System Testing

- The computing systems that make the world run
 - All major industries and sectors have enterprise computing at their core
 - It is not just the mainframe but many systems, devices, software and business processes
 - Working as a system of systems to accomplish a common goal
- System Testing is the testing of all those things together and evaluating them on their enterprise readiness
 - A key attribute of an enterprise service is that it does not fail to deliver the service
 - Testing at this level is proving that it cannot be broken or the service cannot be delivered
 - Complexity levels of the systems and the testing are extreme
 - The opportunity for errors is significantly increased
 - A mix of science, mathematical analysis and systems thinking are required
- Cost of errors can be large in terms of money, lives and opportunities.

Systems Thinking Considerations in Testing

- Conventional thinking on testing is around proving something works
 - This is straightforward science based mathematical thinking

- The use of systems thinking concepts are key
 - Part of the thinking is around proving it can fail until it no longer does
 - Evaluation of the various systems and projecting the second or third order effects
 - Understanding of interrelationships and interactions across the systems
 - Being able to identify subtle behaviors as problems
 - Being able to recognize anti-patterns

- This requires broad and deep knowledge
 - Key components need to be understood including their edge conditions
 - How they interact and the affects multiple components away from the interaction

- Debugging large complex problems is required
 - Often a multitude of problems and symptoms
 - Distillation of facts and symptoms and proof points to back up your assertions

The I & T shaped model

- I-shaped skill models are very deep in a particular discipline
 - They usually do not have much exposure to other components or cross system areas

- T-shaped skill models have depth and breadth
 - Often they end up with depth in multiple areas

- When solving complex problems having many I-shaped engineers doesn't solve the problem
 - Often this hides identification of the problem or problems

 - Have you ever been bounced from one call queue to another and end up back where you started ?
 - I-Shaped thinking is at play here.

Example of a T-Shaped Enterprise System Test Engineer

Operating Systems	Web Technology	Enterprise Systems	Networking	Retail Industry	IT Architecture	Systems Management
-------------------	----------------	--------------------	------------	-----------------	-----------------	--------------------

Breadth:

- Range of differing experiences
- ‘Big picturing’ ability
- Super generalist
- Generally sought out for opinions
- Keeps up to date across the topics
- Knows how to learn and filter
- Business and commercially aware
- Good personal networking

IP Networking

Network Security

Test and debug

Depth:

- Deep knowledge of a few relevant topics
- Connected to the ‘leading edge’
- Level 4-5 in skills terms
- IT Industry connected in the topic
- Leader in ‘networks’ in their community

Qualities of Exemplary System Test Engineers

- Inherent ability to find defects
- Understand and use automation but still think and use your head
- Find defects in the lab, field and in design
- Debug skills
- Design knowledge of products you are testing
- Ability to design tests and products
- Customer use knowledge
- Ability to cheat to prove something works or does not break
- Performance analysis skills

Typical Career Tracks in Systems Testing

- Tester to Developer

- Tester to Quality Assurance (QA) Manager/Director

- Tester to Test Engineer and Technical Executive

– These are not absolute or mutually exclusive

Tester to Developer

- Typical developer new hire path
 - One almost never starts out changing the key algorithms of a complex product

- Typically an I-shaped skill model
 - Starts usually with functional testing
 - Potentially loaned out to system test at times
 - Will eventually code a small component and test other components
 - Then move on to full development or stay in formal function test

- Not uncommon to move to system testing if they find they like testing over developing

- Academic training is all about being a “developer” so students tend to seek out these positions

Tester to Quality Assurance (QA) Manager/Director

- Most often chosen path for career test engineers
- Typically a T-Shaped skill model
- Starts usually with test execution and moves to various components
- Customer involvement is usually indirect
 - Recreating a field problem not solving the problem at the customer
- There are plateau points in this path
 - One usually becomes a test architect and is involved in designing broader test plans and sizing new work items or products
 - Typically involved in the test planning stage of the testing
 - Or one take the project management path and does the managing of the development product once it is in test.
 - Defect prioritization
 - Resource allocation and movement
 - Execution ordering and optimization
- Further growth is usually obtained by entering management of test and then executive management

Tester to Test Engineer to Technical Executive

- Very small percentage of engineers take this path
- A heavy T-Shaped skill model
 - Significant depth usually in multiple areas
 - Significant breadth and a fair amount of depth in breadth areas
- Starts usually with test execution and moves to various components and products
- Customer involvement is usually direct with long lasting relationships
- Usually a short stay as test architect and back to deep test execution in many areas within any given period.
- Considerable time spent with development on design correctness items
- Raises the bar for test engineering at all levels continually
 - The test expert as well as a multi-domain expert
- Takes on executive decision making in technical disciplines for an organization
- Ultimate arbiter for technical issues in test and the product especially on ship decisions
- Industry and Academic influence is required and common
- Innovation and invention in your industry as well as publication are critical
- IBM STG has had this model for the past decade –
 - IBM SWG and other IT companies are emerging with this model
 - Enterprise computing and disciplines are a key enabler for this

System Testing Pre-Hire to Retire (What students should consider)

- Pre-Hire Computer Science Major
 - During your junior year start building diversified skills
 - Take a testing class if offered – You will be doing this !
 - Take courses that provide cross platform experience
 - Take some classes or get exposure to non-IT industry sectors
 - Show you are more than just an “IT person”
- Pre-Hire Accounting Major
 - During your junior year take some computer science classes
 - Given you need 150 credits to sit for your CPA exam
 - Take on a full two semester load of enterprise computing classes for 5th year
 - IT is a significant portion of any enterprise and understanding it is an asset
 - You are more competitive in the accounting and IT job market
 - You can system test or implement at the technology or application levels
- Pre-Hire Finance Major
 - During the latter part of your junior year diversify with computer science classes
 - In your senior year add in enterprise computing classes
 - Diversification will allow you to be competitive in the financial industry
 - An Enterprise IT knowledgeable financial analyst can use technology understanding as a key to analyzing most companies or testing or providing technology products and services to the financial industry
- Use co-op assignments and guest lecture opportunities to build your technical network

System Testing Pre-Hire to Retire (Once you have the job)

- Entry System Test Engineer (0-5 years)
 - Enterprise level expertise does not show up in a few weeks
 - Learn your product domain
 - You can not test something you don't know
 - Learn how to construct the test environment – don't just use what's there !
 - Follow the defects you find to resolution and keep doing it
 - Debugging is key to understanding what is going on
 - Build your technical network and don't be afraid to ask questions
- Mid-Career Test Engineer (6-20 years)
 - Focus on building out your T-Shape along with going deeper or adding additional depth areas
 - Take on stretch assignments outside of your comfort zone
 - Become the recognized expert in as many areas as you can
 - Ensure you stay aware of the outside world
 - Build customer knowledge and foster long term relationships
 - Don't plateau and keep learning
 - Learn how to be productive at sizing new work and don't spend all day doing it
 - Build your technical network across divisions and companies

System Testing Pre-Hire to Retire (Once you have the job)

- Late Career Test Engineer (20-30+ years)
 - Use the cyclical nature of technology as a productivity leverage point
 - Be smarter about doing it on the next cycle
 - Develop completely new depth areas (Original areas might be getting a little stale)
 - This is a key differentiator and keeps you from being bored
 - Focus on working with and teaching new test engineers to raise the bar
 - Things are more complex and we have less time to run up the learning curve
 - Create collaborative learning environments for new engineers
 - Don't be that grouchy old person that says "I have seen it before this is nothing new"
 - Listen to new ideas and change things even if it has never been done that way
 - It helps the newer engineers build confidence