

# *In Praise of Specialization*

*Or, No, Not Just Anyone Can Design, Test, Etc.*



## RBCS

TIME TESTED.  
TESTING IMPROVED.

[www.RBCS-US.com](http://www.RBCS-US.com)



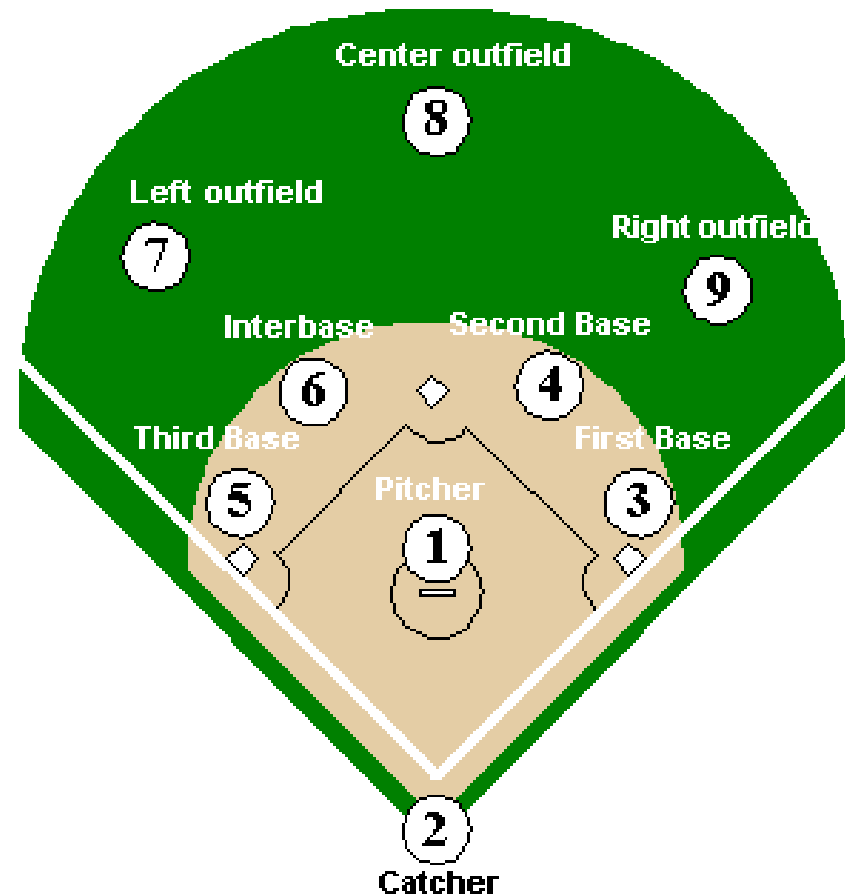
## *Introduction*

- It's hip these days to talk about cross-functional teams
- Hierarchy and specialization are *verbotten*
- As Scrum Grand Poobah Ken Schwaber wrote: "You don't have to be a tester to test, or a designer to design."
- He's not the only one saying that specialization is passé, and that testing especially is something anyone can do
- I couldn't disagree more...



# *Would This Work in Baseball?*

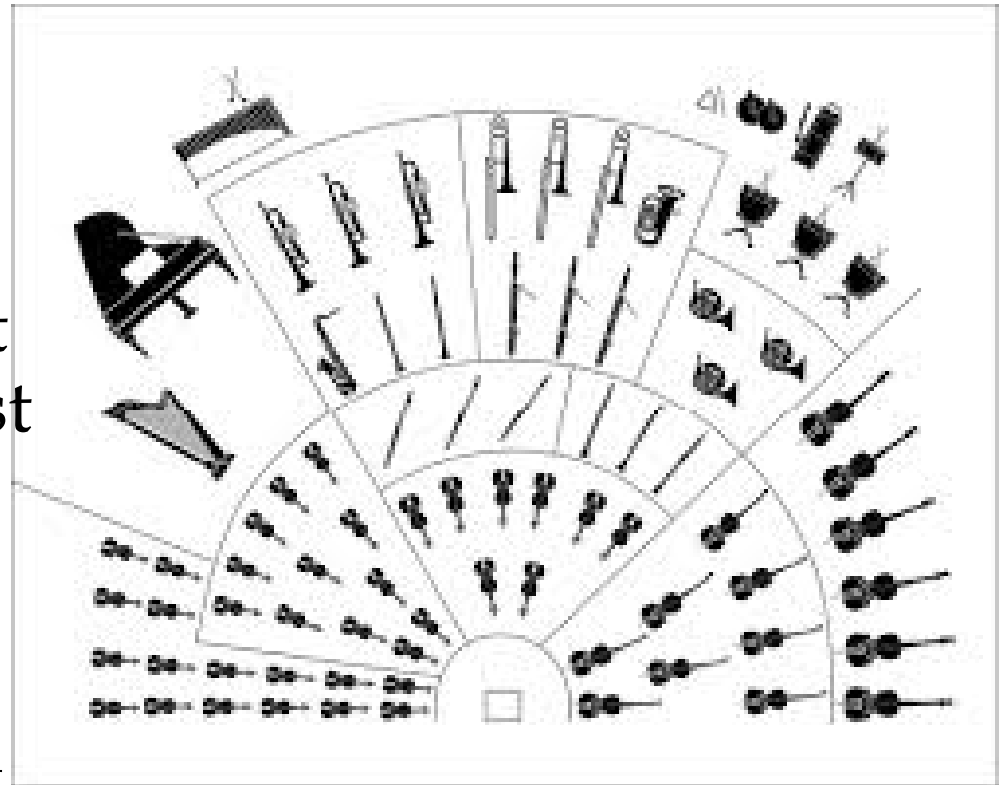
- Consider this paraphrase of Schwaber
- “If you’re a baseball player, you don’t have to be a pitcher to pitch, or a catcher to catch.”
- Would that work in pro baseball?





## *Would This Work in Music?*

- How about this paraphrase of Schwaber?
- “If you’re a musician, you don’t have to be a violinist to play violin, or a pianist to play piano.”
- Would that work in professional music?





## *The 10,000 Hour Rule*

- Gladwell cites the 10,000 hour rule in *Outliers*
- This rule comes from a paper by Anders, et al.
- From the intro: “One thing emerges very clearly from Bloom’s work: All the superb performers he investigated had practiced intensively... Later research building on Bloom’s pioneering study revealed that the amount and quality of practice were key factors in the level of expertise people achieved. Consistently and overwhelmingly, the evidence showed that experts are always made, not born.”
- This rule applies to any intellectual challenging endeavor, of which software testing is certainly one



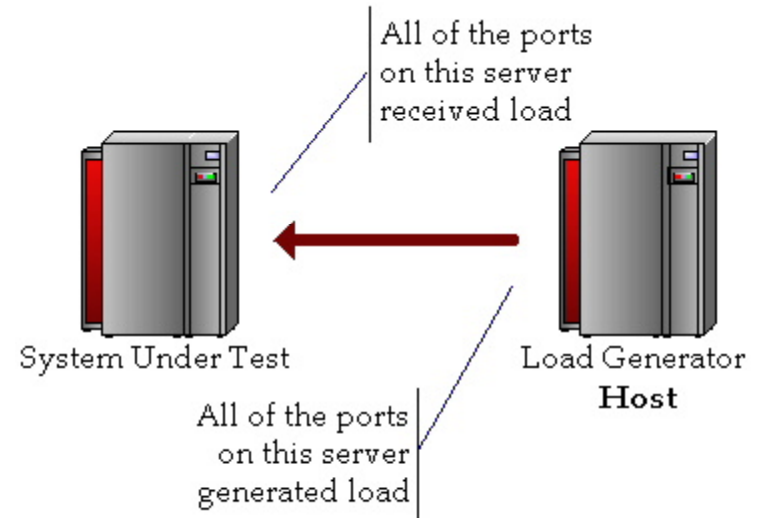
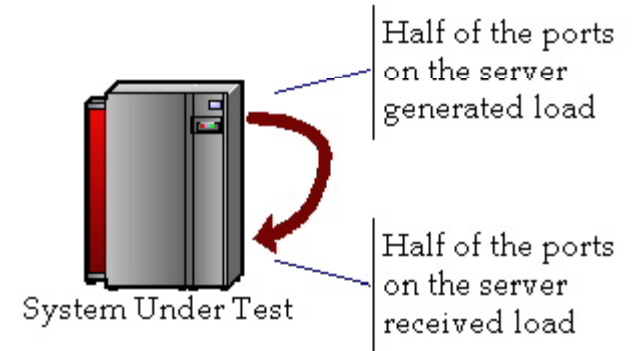
## *Examples of Failure and Success*

- Enough of the metaphors, books, research, and academic papers!
- How about some examples of the 10,000 hour rule in practice?
- I'll give you three situations where a lack of skills, specialization, or experience lead to disaster
- I'll then give you three situations where skills, specialization, and experience prevented unpleasant outcomes
- And then I'll explain what Karl Marx has to do with all of this...



# “Cogeneration is Not Okay, Guys”

- For an IVR server, the developers tested performance using cogeneration of load
- The total load was unrealistically low
- Warnings from test professionals were ignored
- We (the test professionals) then tested the system the right way
- We loaded all ports
- The tests failed...and so did the project, ultimately

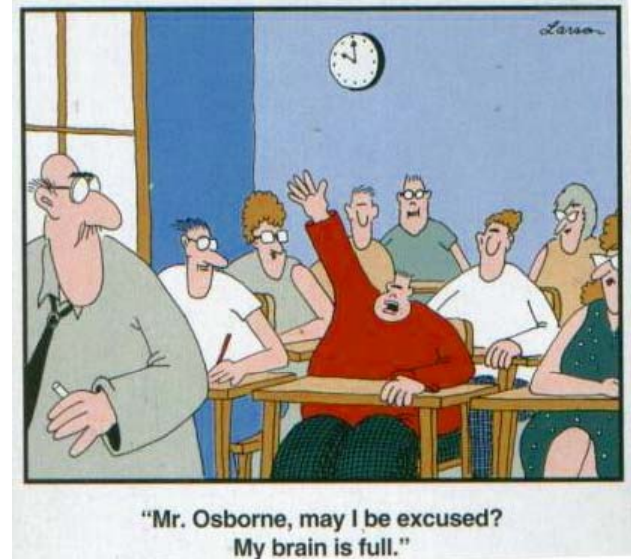






# *“We Are the Designers”*

- A large, complex project to merge insurance systems hit the rocks
- I was invited to do an assessment to see if it could be saved
- The root cause: Two “database designers” with no experience beyond building a few small Access applications had been in charge of the database architecture
- The database did not scale and it was too late to re-design the system
- The project was pronounced a failure and cancelled after three years and hundreds of person-years of effort
- I guess maybe you do have to be a designer to design...at least if you want something to work!







# *Air Disaster*

- Asiana Airlines lost a 777 and (worse) three people on July 6, 2013
- The cause was determined to be pilot error
- According to Wikipedia:
  - Captain Lee Jung Min (co-pilot) filled dual role of check/instructor captain and pilot in command... This was his first flight as an instructor.
  - Captain Lee Kang Guk (pilot) was receiving his initial operating experience (IOE) training and was halfway through Asiana's IOE requirements. This was [his] first landing at SFO in this aircraft type... This was his first flight with Lee Jung Min.
- Could a more experienced team have landed the plane without incident?





## *Whittling Away the Performance Risk*

- “What if we can’t support 50,000 simultaneous devices with our servers?”
- An experienced system architect and my experienced performance testers took on the risk
  - Spreadsheet model of resource requirements
  - Dynamic Monte Carlo simulation of system behavior
  - Performance testing of critical units at the unit level
  - Performance testing of the system at the system level, compared to the simulation results
- Result: no significant performance or load-related problems after release



## *“We Think We Should Test It First”*

- For the largest consumer products recall in history, our client got a bit worried about handling load
- Unfortunately, they called us three weeks before the mandatory go-live date
- A crack team of testers, with more than 100 years of experience in total, took on the task a week after the call
- After ten solid days of 12 hour days, we whipped the product into shape
- Our client’s recall software worked in production
- The other affected company apparently didn’t test as thoroughly, at the expense of their hapless customers





## *Air Miracle*

- A US Airways flight hit a flock of Canada geese, suffering a total loss of power
- The crew, led by Capt. Chesley “Sully” Sullenberger, landed the plane on the Hudson River
- Not one person died, though a few geese were pureed
- NTSB board member Kitty Higgins: “the most successful ditching in aviation history... These people knew what...to do and they did it...”
- Sully: “[F]or 42 years, I’ve been making small, regular deposits in this bank of experience, education, and training. And on January 15 the balance was [enough to] make a very large withdrawal.”
- Sully the specialist saves the day!





## *We've Seen this Messianic Movie Before...*

- “For as soon as the distribution of labor comes into being, each man has a particular exclusive sphere of activity, which is forced upon him and from which he cannot escape. He is a hunter, a fisherman, a shepherd, or a critical critic and must remain so if he does not wish to lose his means of livelihood; while in communist society, where nobody has one exclusive sphere of activity but each can become accomplished in any branch he wishes, society regulates the general production and thus makes it possible for me to do one thing today and another tomorrow, to hunt in the morning, to fish in the afternoon, rear cattle in the evening, criticize after dinner, just as I have in mind, without ever becoming hunter, fisherman, shepherd or critic.” Karl Marx, *The German Ideology*, Vol. 1, Part 1
- How did that Communist thingy end up working out as an organizing principle for society?





RBCS

TIME TESTED. TESTING IMPROVED.

# *Celebrating the Unraveling of a Bad Idea*





## *Conclusions*

- ✚ Okay, so I'm exaggerating
- ✚ Anti-specialization isn't as horrible, soul-crushing, or dangerous as Soviet Communism was
- ✚ However, as I've shown, the lack of skills and specialization can contribute to project disasters
- ✚ The presence of qualified specialists is a common theme in successful endeavors, including in software engineering





## *To Contact RBCS*

For 20 years, RBCS has delivered consulting, outsourcing and training services to clients, helping them with software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS advises its clients, trains their employees, conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit [www.rbc-us.com](http://www.rbc-us.com).

Address: RBCS, Inc.  
31520 Beck Road  
Bulverde, TX 78163-3911  
USA

Phone: +1 (830) 438-4830  
Fax: +1 (830) 438-4831  
E-mail: [info@rbc-us.com](mailto:info@rbc-us.com)  
Web: [www.rbc-us.com](http://www.rbc-us.com)