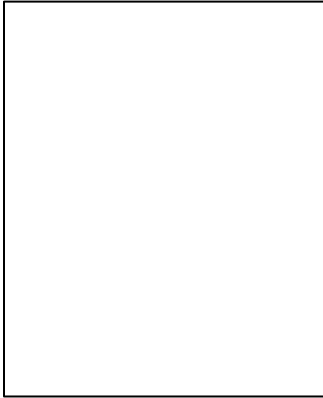


Shift Left and Friends

And What They Mean for Testers



Replica of the shift left persuasion device (i.e., 2x4)
used by a test manager colleague in 1990s





Introduction

- ⊕ All the IT world is abuzz with the latest buzz words
 - ⊞ Shift left
 - ⊞ Continuous integration and continuous delivery (CI/CD)
 - ⊞ Continuous deployment
 - ⊞ DevOps
- ⊕ What is all this and what does it mean for testers?
- ⊕ Are you (or should you be) an SDET?
- ⊕ What's going on with automation?
- ⊕ Are you worried?
 - ⊞ Don't worry
 - ⊞ Be happy
- ⊕ A left-shifted tester life is gonna be fun and exciting



What's Shift Left, and Is It My Friend?

- ✦ Basically, outgrowths of Agile, Lean, Kanban, and open source movements of the last two decades
- ✦ Let's unbuzz the buzzwords
 - ✦ Shift left
 - ✦ Continuous integration/continuous delivery (CI/CD)
 - ✦ Continuous deployment
 - ✦ DevOps
- ✦ Automation (not just test automation) is central to making this stuff work
- ✦ So, what does it mean for you as the tester?



Shift left ain't your granddad's waterfall
It really ain't waterfall at all



What Is an SDET and Am I an SDET?

- ❖ Different names for technical test analysts
 - ❖ SDET: Software Development Engineer in Test
 - ❖ SET: Software Engineer in Test
 - ❖ Also DevOps Engineer or DevSecOps Engineer
- ❖ SDETs
 - ❖ Are highly technical
 - ❖ Use, build, and customized tools
 - ❖ Read and write scripts and code
- ❖ The next step in the long-term trend requiring technical skills for testers
- ❖ What SDETs do...

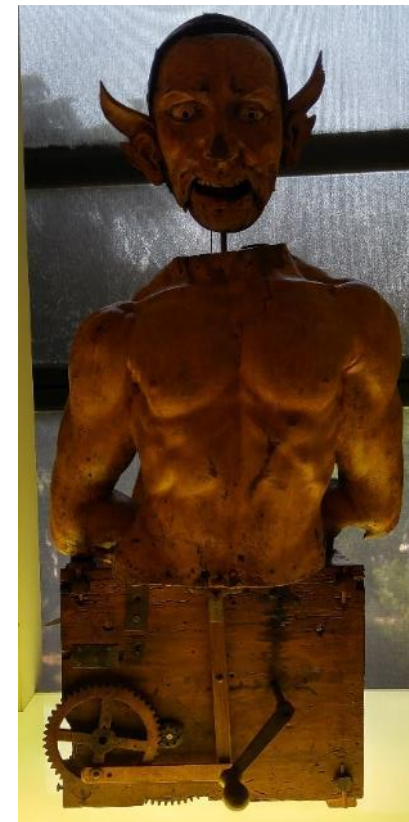


The SDET doesn't just tell time. The SDET knows how the clock works inside and why



SDETs Automate Testing and More

- ✦ Yes, SDETs automate testing
- ✦ Know how to automate through:
 - ▣ GUI
 - ▣ CLIs
 - ▣ APIs
 - ▣ Data layer
 - ▣ Network layer
- ✦ Automate at multiple levels
 - ▣ Unit
 - ▣ Unit integration
 - ▣ System
 - ▣ System integration
- ✦ Augment manual tests with automated tests
- ✦ But there's more than automation...



GUI focused test automation: a wrong turn whose time has come and gone



Unit Test and Code Coverage Coaching

- ☉ SDETs should coach devs
- ☉ Good unit testing practices
 - ☒ White-box techniques
 - ☒ Black-box techniques
 - ☒ Tool support
- ☉ Key code coverage measures
 - ☒ Statement
 - ☒ Branch
 - ☒ MC/DC
- ☉ These techniques can
 - ☒ Contribute to better testing
 - ☒ Lead to false confidence when misused
- ☉ Coach on concepts, not just tools

```
void ObjTree::AddObj(const Obj& w) {  
    // Make sure we want to store it  
    if (!(isReq() && isReq() && (isNort() || (isFlat() && isFreq())))) {  
        return;  
    }  
    // If the tree is currently empty, create a new one  
    if (root == 0) {  
        // Add the first obj.  
        root = new TObjNode(w);  
    } else {  
        TObjNode* branch = root;  
        while (branch != 0) {  
            Obj CurrentObj = branch->TObjNodeDesig();  
            if (w < CurrentObj) {  
                // Obj is new or lies to left of the current node.  
                if (branch->TObjNodeSubtree(LEFT) == 0) {  
                    TObjNode* NewObjNode = new TObjNode(w);  
                    branch->TObjNodeAddSubtree(LEFT, NewObjNode);  
                    break;  
                } else {  
                    branch = branch->TObjNodeSubtree(LEFT);  
                }  
            } else if (CurrentObj < w) {  
                // Obj is new or lies to right of the current node.  
                if (branch->TObjNodeSubtree(RIGHT) == 0) {  
                    TObjNode* NewObjNode = new TObjNode(w);  
                    branch->TObjNodeAddSubtree(RIGHT, NewObjNode);  
                    break;  
                } else {  
                    branch = branch->TObjNodeSubtree(RIGHT);  
                }  
            } else {  
                // Found match, so bump the counter and end the loop.  
                branch->TObjNodeCountIncr();  
                break;  
            }  
        } // while  
    } // if  
    return;  
}
```

SDET: Help devs test this kinda stuff



Get Inside the Matrix

- ✦ Swallow the red pill and wake up to what's really happening while you test
- ✦ Learn how OS options affect app behavior
- ✦ Use dynamic analysis tools to watch
 - ▣ Memory
 - ▣ CPU
 - ▣ Disk
- ✦ Check for memory leaks



Bulletproof your apps from inside
(Image ifc.com)



Little Data, Big Data, All Kinds of Data

- ✦ For certain apps, data is central to what they do, so focus testing there
- ✦ Know how to:
 - ✦ Work with data directly and via tools
 - ✦ Create and use full and partial test oracles on data
 - ✦ Automate directly and indirectly at the data layer
 - ✦ Review metadata, database design (e.g., ER diagrams), etc.
 - ✦ Test replicated and distributed databases
- ✦ Data, data structures, and data tools (SQL, no-SQL, tree, etc.) are highly technical



Data is big. Really big



An SDET Gets Security

- Like data, security testing is highly technical and complex

- Know how to:

- Check security settings on IDS/IPS, firewalls, etc.

- Use static analysis tools

- Sniff network traffic

- Participate in code security and design security reviews

- Evaluate encryption and password implementations

- Test complex authentication and authorization scenarios, including role-based security

- Skills in both app and network security will be in high demand for the foreseeable future

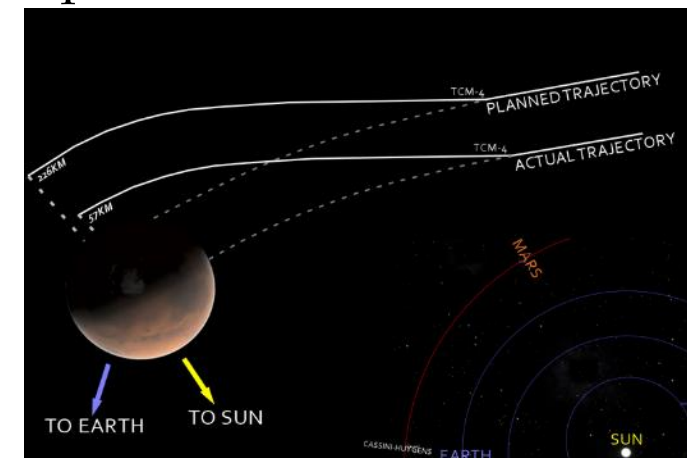


Security: ever more critical in a risky world...



Really, Truly Do Integration Testing

- ✦ Everything talks to everything else (especially in mobile)
 - ✦ Integration testing still under-performed and misunderstood
 - ✦ Some wreckage on Mars can help explain it
- ✦ The places in between need a champion, and the SDET can do it
- ✦ Know how to:
 - ✦ Apply white-box, black-box, and other techniques to design important interoperability and other integration tests
 - ✦ Identify direct (e.g., API) and indirect (e.g., data layer) integration points
 - ✦ Analyze interfaces, data sources, and data sinks as part of integration test coverage analysis
- ✦ Virtualization can help, but false negatives are a risk



By Xession (wikipedia)



Do Stuff at Random (Sort Of)

- ✦ Test design techniques are useful, but not everything is predictable
- ✦ Further, hackers sometimes use chaos
- ✦ The answer: random testing
- ✦ Know how to:
 - ✦ Use fault injection/fuzzing tools on config and/or data files
 - ✦ Create and unleash dumb monkeys
 - ✦ Test for randomness in data sequences
- ✦ Randomness is especially useful in testing reliability, security, and more

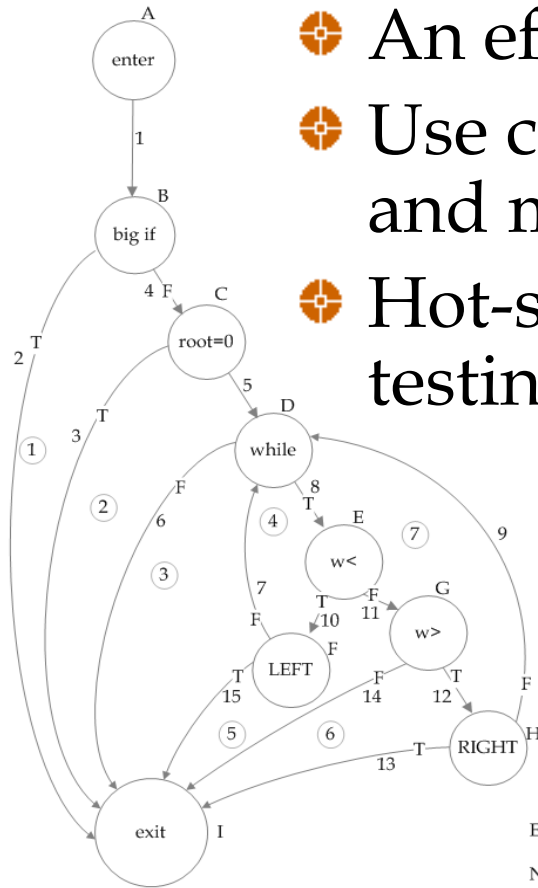


Random. It's your friend.



Predict and Find Bug Clusters

- ☛ An effective bug hunter knows where to hunt bugs
- ☛ Use complexity analysis, defect and incident data, and more...
- ☛ Hot-spot analysis especially useful (e.g., regression testing)

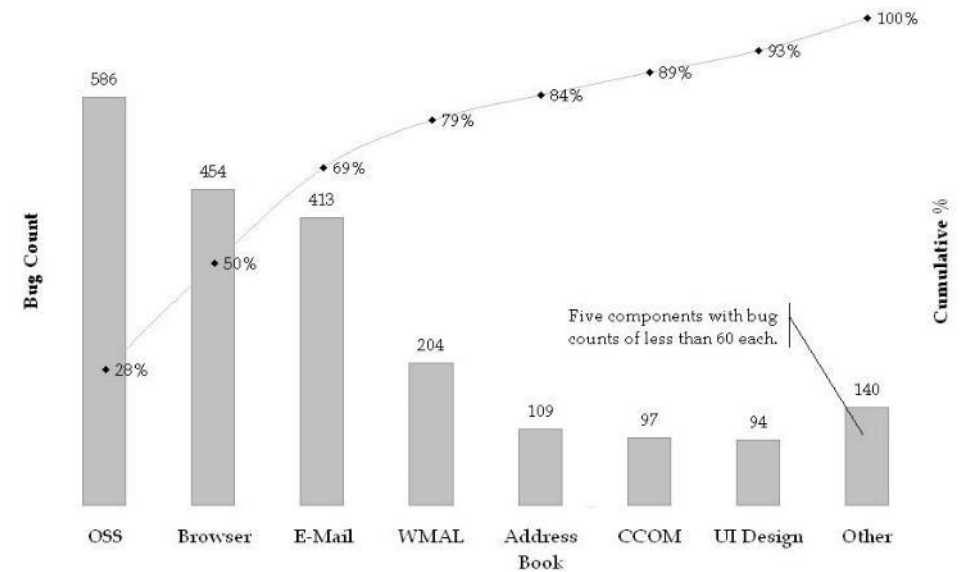


E = 15

N = 9

R = 7

C = 7 + 1 = 15 - 9 + 2 = 8





Learn from Challenger and Columbia

- ✦ Learn from others' mistakes
- ✦ Two more lessons from NASA
- ✦ Challenger
 - ❖ Engineers predicted O-ring problem
 - ❖ Presentation was not convincing
 - ❖ Lesson: Effectively communicate about risk
- ✦ Columbia
 - ❖ Engineers knew about ice strikes
 - ❖ Deferred action since “nothing bad happened on previous launches” and “nothing can be done anyway”
 - ❖ Lessons:
 - Do not extrapolate risk probabilities from small samples
 - Evaluate risks and test results from the user/customer perspective
- ✦ Note: NASA's safety record is amazing: in this most risky of ventures, they have only lost 17 people (never from a software failure)



Video: CNN. GIF: giphy.com



Laugh When People Say Testing is Dead

- ⊕ Testing is **not** dead, though it's provocative to say so
- ⊕ In fact, the SDET is the original tester, now cool again
- ⊕ Beizer and Myers were SDETs before SDETs were cool
- ⊕ I got my start as an SDET, and I still am an SDET (in fact, a Senior DevOps Consultant)
- ⊕ Now, as IT shifts left, it's your time to become an SDET
- ⊕ And, if someone says, "Testing is dead," laugh at them
- ⊕ After all, it's rude to eat them like an undead zombie would
- ⊕ Plus, people who say testing is dead are just wrong, as testing is more important than ever



Devs see you this way? You're doing it wrong
(Film: Dawn of the Dead. GIF: giphy.com)



Conclusion

- ✦ Shift left and friends: best practices that go back to the start of software testing
- ✦ To make them your friends, you need to be an SDET (even if your title is something else)
- ✦ If you take advantage of the technical opportunities open to you, you can thrive in a shift left world
- ✦ If you don't get technical, experience may hit you like my colleague's 2x4, so, as Ren would say...
- ✦ **SHIFT LEFT!!!**



...*Contact RBCS*

For over 25 years, RBCS has delivered consulting, training, and expert 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, provides advice, reports, and testimony for law firms in legal matters, and hires test staff for thousands 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. Find us at the coordinates below:

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

Phone: +1 (830) 438-4830

E-mail: info@rbc-us.com

Web: www.rbc-us.com

Twitter: @RBCS, @LaikaTestDog

Facebook: @TestingImprovedbyRBCS

LinkedIn: <https://www.linkedin.com/in/rex-black>

YouTube: <https://www.youtube.com/user/RBCSINC>

Podcasts: <https://rbc-us.com/resources/podcast/>
www.rbc-us.com