Testing in a Continuous Deployment Environment

Think, Pair, Share

David Grabel CSM, CSP, SPC, AHAF

Enterprise Agile Coach, Vistaprint

vistaprint

Grabel Consulting Services

GRABEL CONSULTING
Enterprise Agile Coaching

Former VP, Product Development

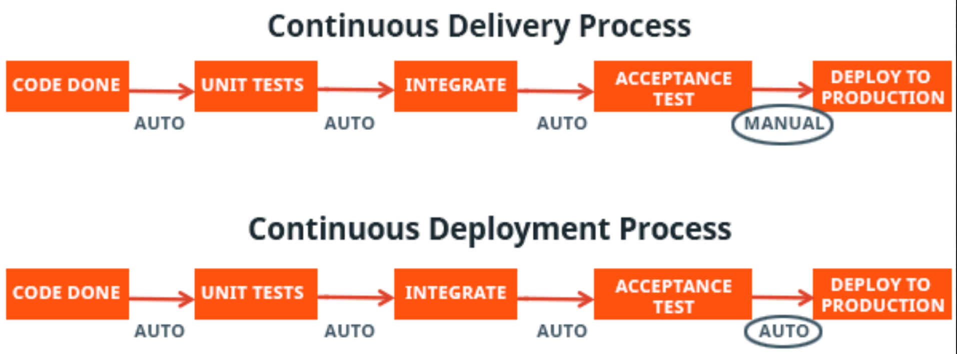
Continuous Integration

Continuous Delivery

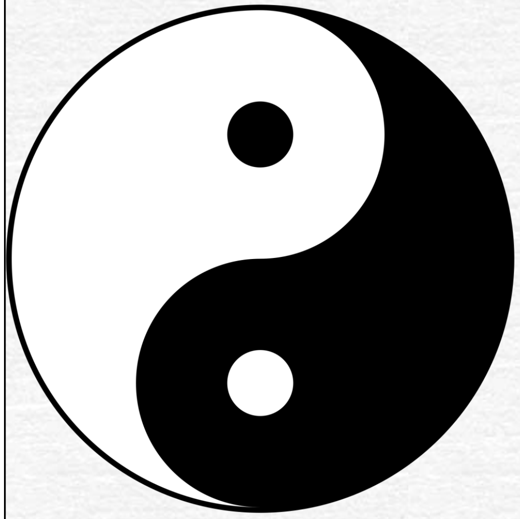
Codebase is always deployable

Continuous Delivery

Codebase is always deployed



Why?

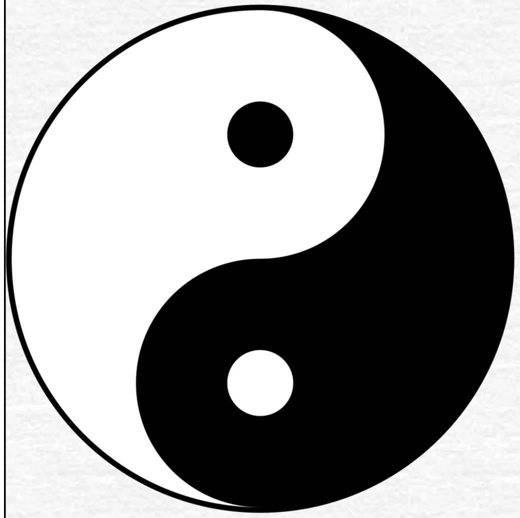


Quality enables Speed

Speed depends on Quality

Speed + Quality = Competitive Advantage

Why Not?

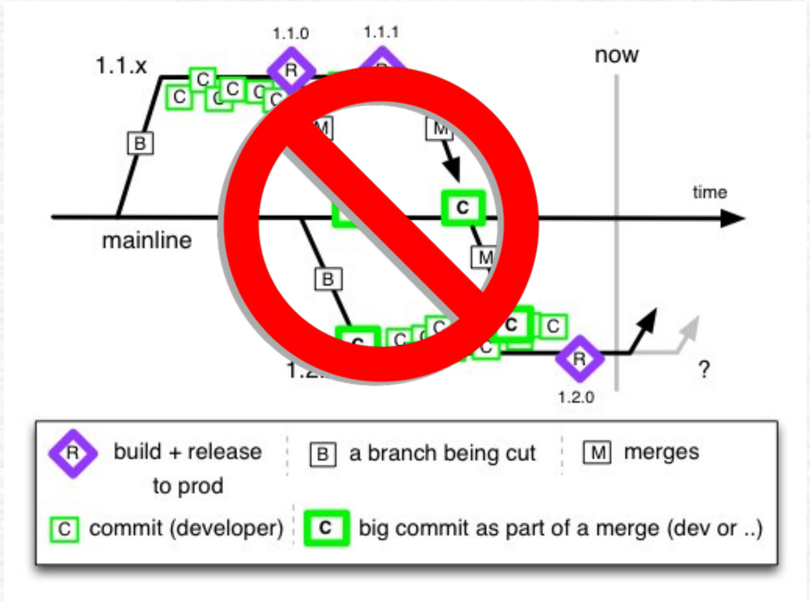


Not for everyone

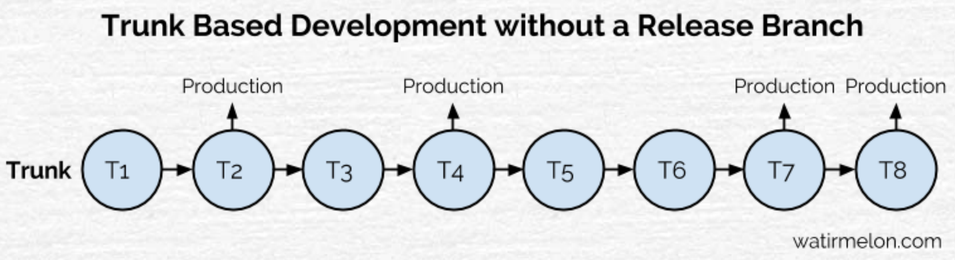
Everyone should develop as if they can deploy continuously

Fosters best practices

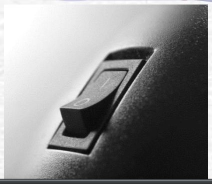
SW Engineering Practices

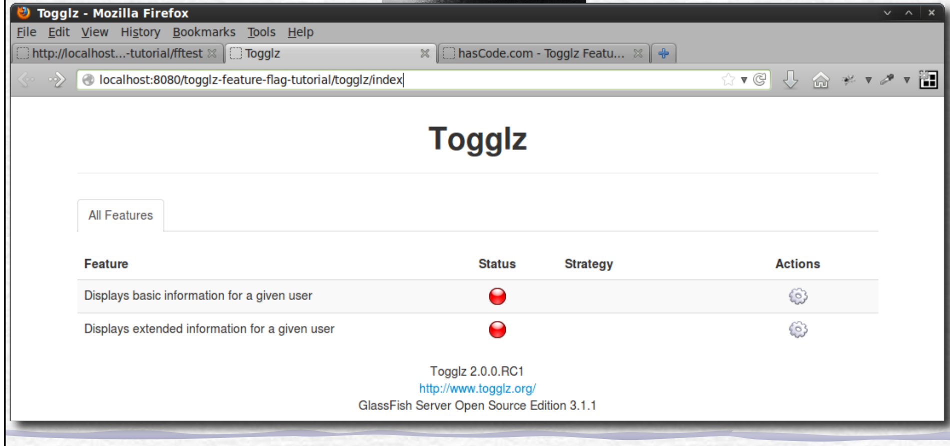


SW Engineering Practices



SW Engineering Practices




Togglz - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost...-tutorial/fftest
Togglz
hasCode.com - Togglz Featu...
localhost:8080/togglz-feature-flag-tutorial/togglz/index

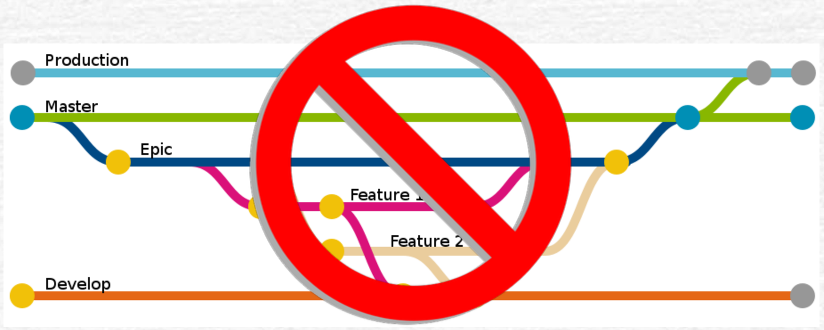
Togglz

All Features

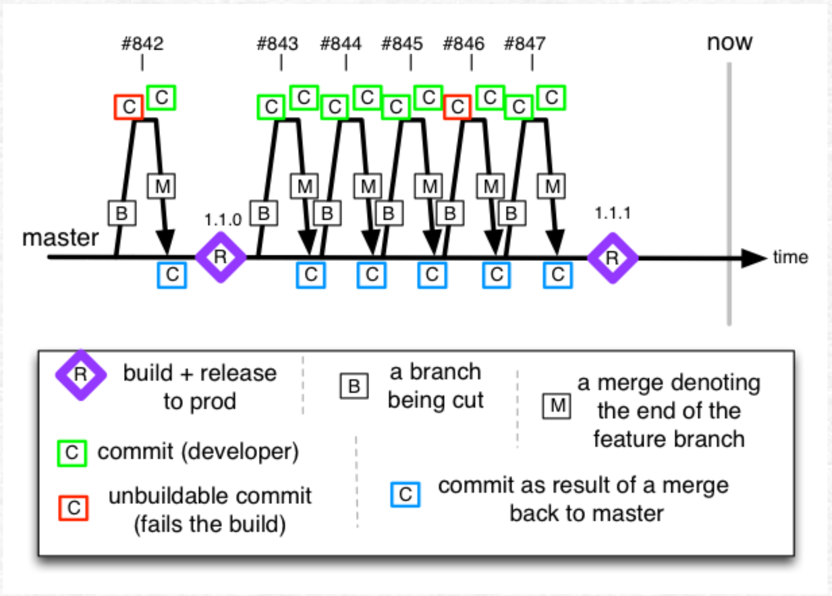
Feature	Status	Strategy	Actions
Displays basic information for a given user	🔴		⚙
Displays extended information for a given user	🔴		⚙

Togglz 2.0.0.RC1
http://www.togglz.org/
GlassFish Server Open Source Edition 3.1.1

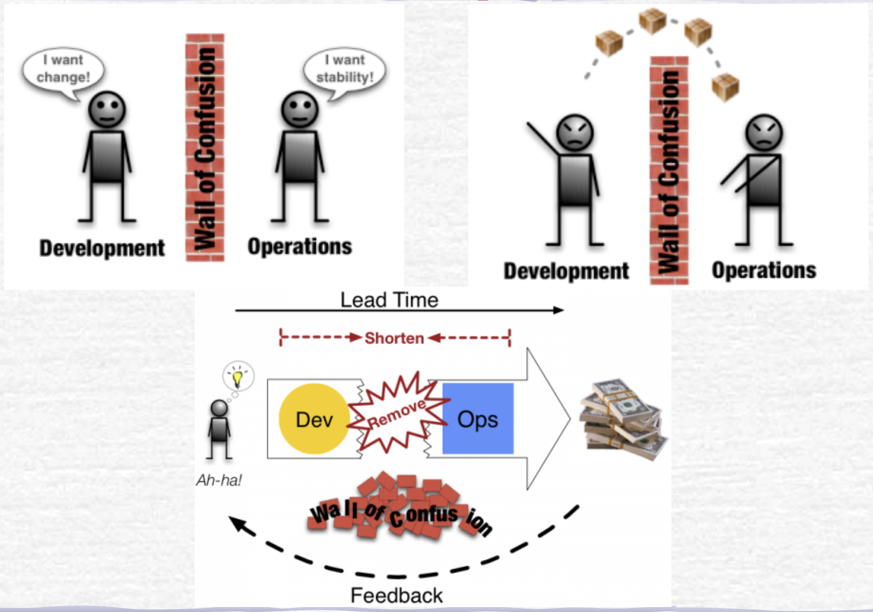
SW Engineering Practices



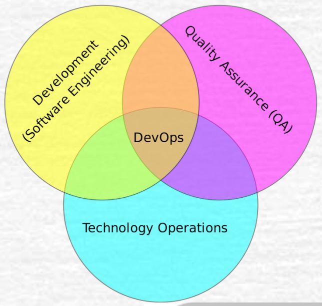
SW Engineering Practices

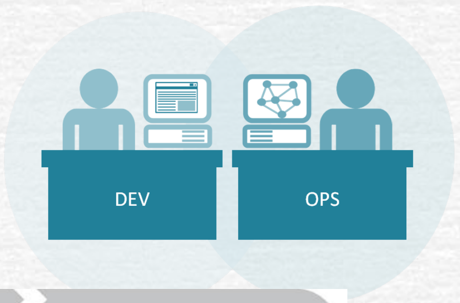

#842
#843
#844
#845
#846
#847
now
master
1.1.0
1.1.1
time
R build + release to prod
B a branch being cut
M a merge denoting the end of the feature branch
C commit (developer)
C unbuildable commit (fails the build)
C commit as result of a merge back to master

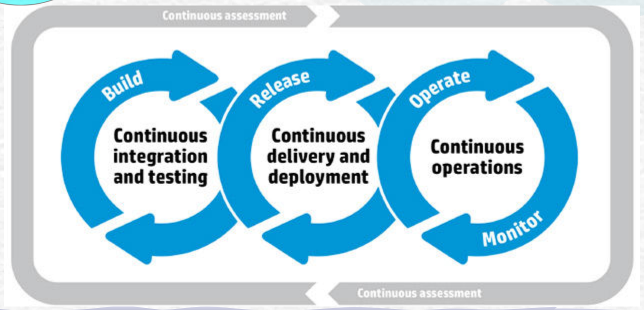
DevOps

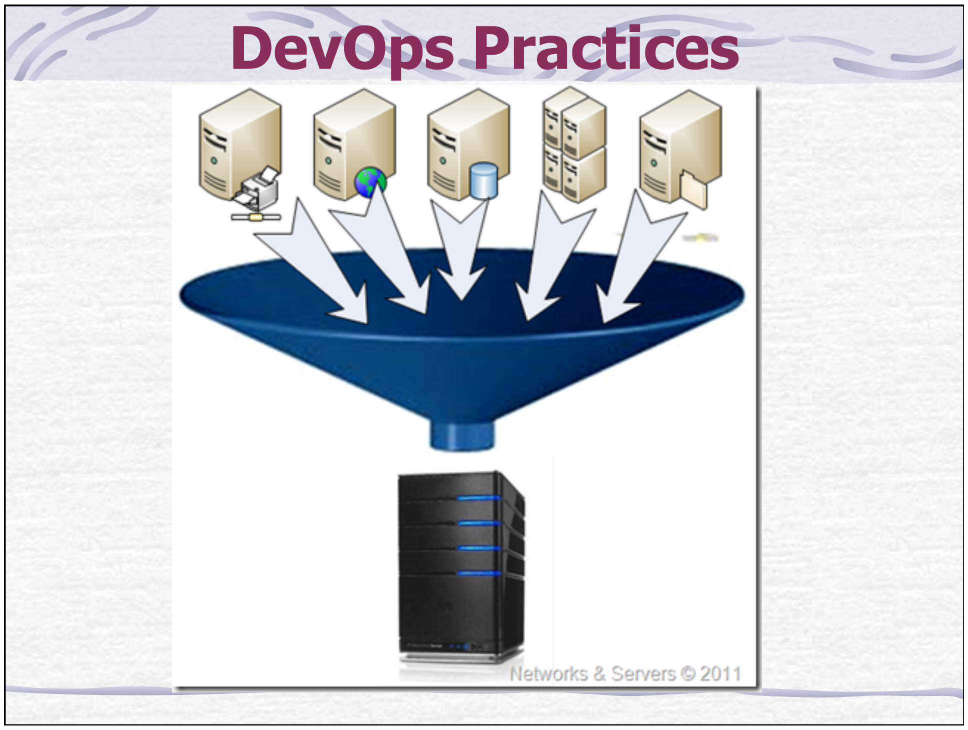


DevOps









DevOps Practices

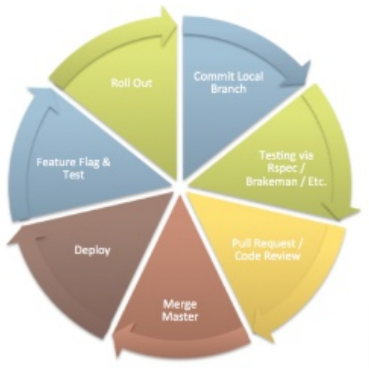
DevOps Practices

DevOps Practices

A SecDevOps Use Case

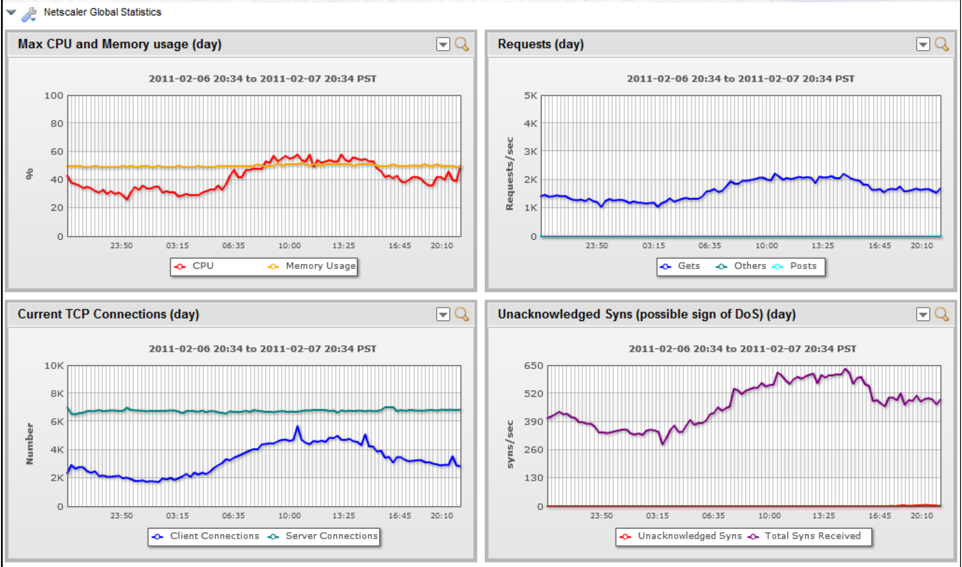
Continuous Security Testing

- Brakeman Static Analysis
- Dynamic Application Scanning
- Paid Penetration Testing Service
- Open Bug Bounty Program
- Continuous Infrastructure Scanning

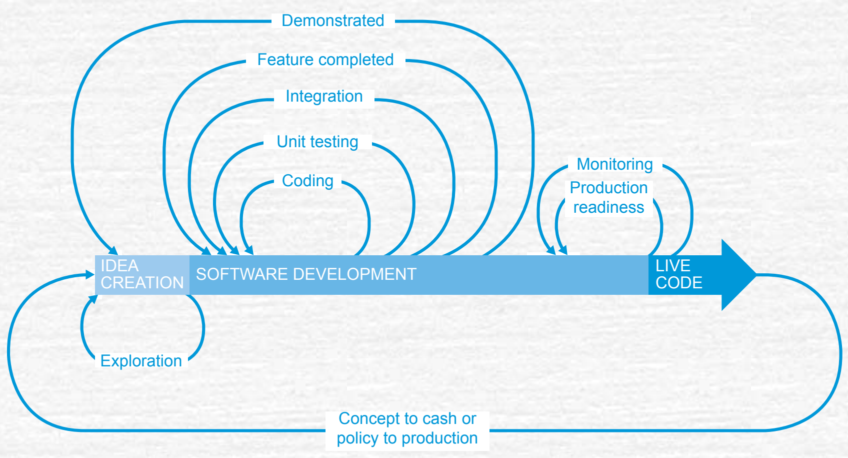


All Results Loaded into Kenna via API

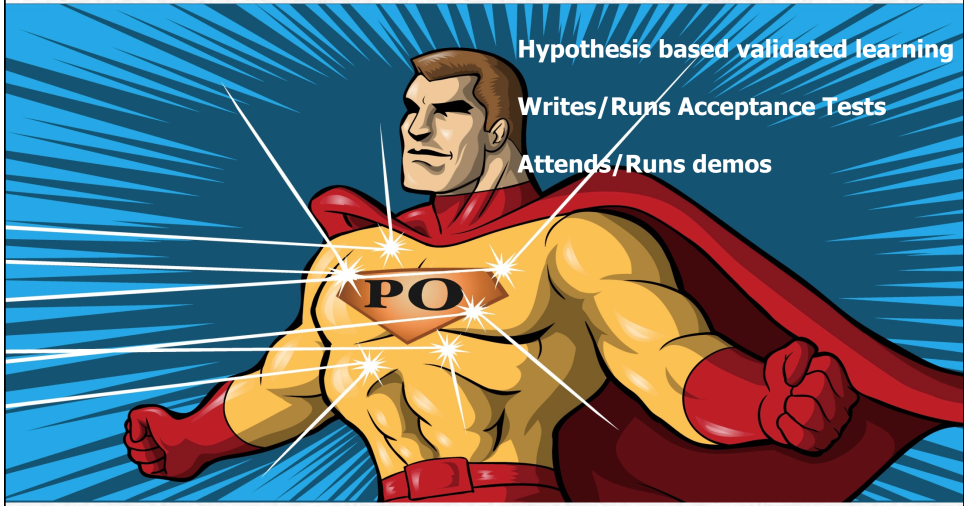
DevOps Practices


Netscaler Global Statistics
Max CPU and Memory usage (day)
2011-02-06 20:34 to 2011-02-07 20:34 PST
%
CPU
Memory Usage
Requests (day)
2011-02-06 20:34 to 2011-02-07 20:34 PST
Requests/sec
Gets
Others
Posts
Current TCP Connections (day)
2011-02-06 20:34 to 2011-02-07 20:34 PST
Number
Client Connections
Server Connections
Unacknowledged Syns (possible sign of DoS) (day)
2011-02-06 20:34 to 2011-02-07 20:34 PST
syns/sec
Unacknowledged Syns
Total Syns Received

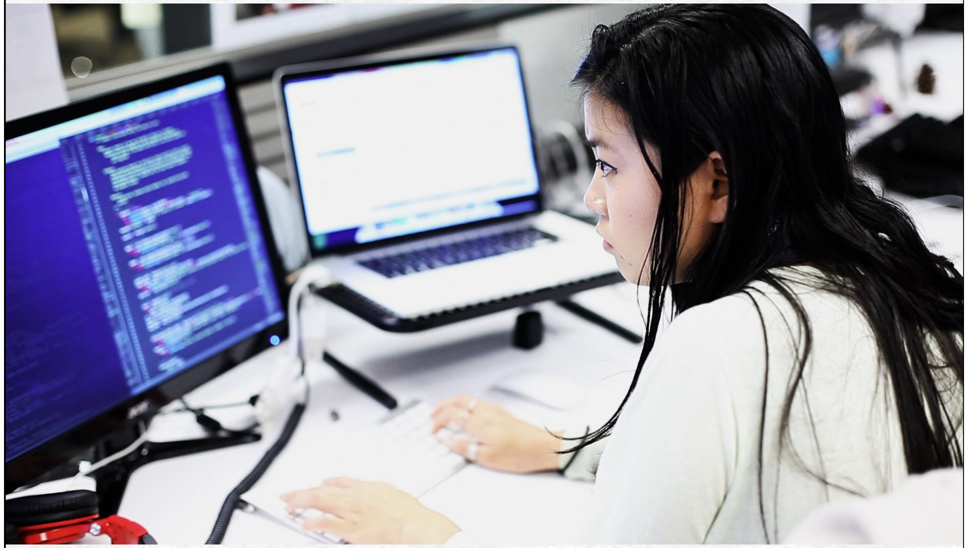
Drive Quality with Fast Feedback



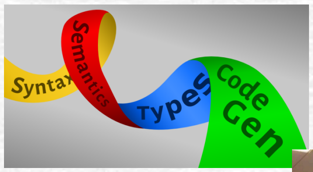
Product Owner is a Tester


Hypothesis based validated learning
Writes/Runs Acceptance Tests
Attends/Runs demos
PO

IDE continuously tests code



Static Analysis



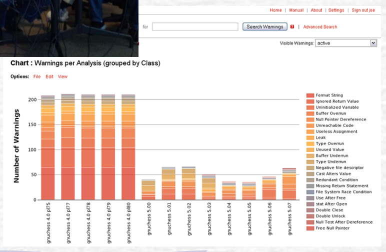
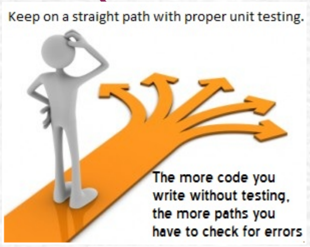
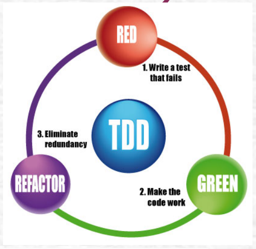


Chart : Warnings per Analysis (grouped by Class)

Design becomes automated testing

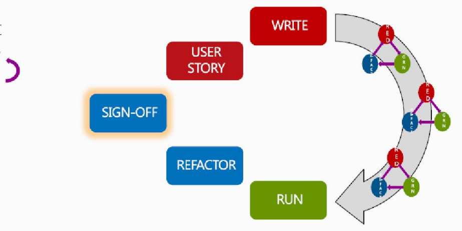
(or did I mean that in reverse)*


Keep on a straight path with proper unit testing.
The more code you write without testing, the more paths you have to check for errors



(Acceptance Test Driven Development)

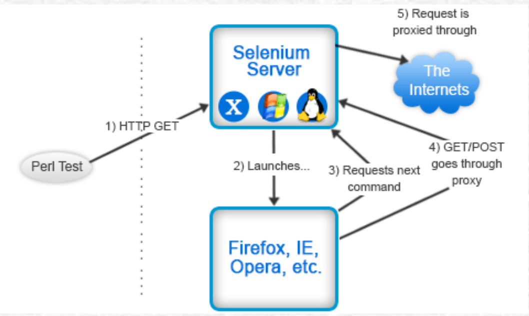
- Select User story
- Write Acceptance Test
- Implement User Story
- Run Acceptance Test
- (Refactor)
- Get Sign-Off



* Steven Sondheim - **Company**

Automated Functional Testing

Domain Specific Language

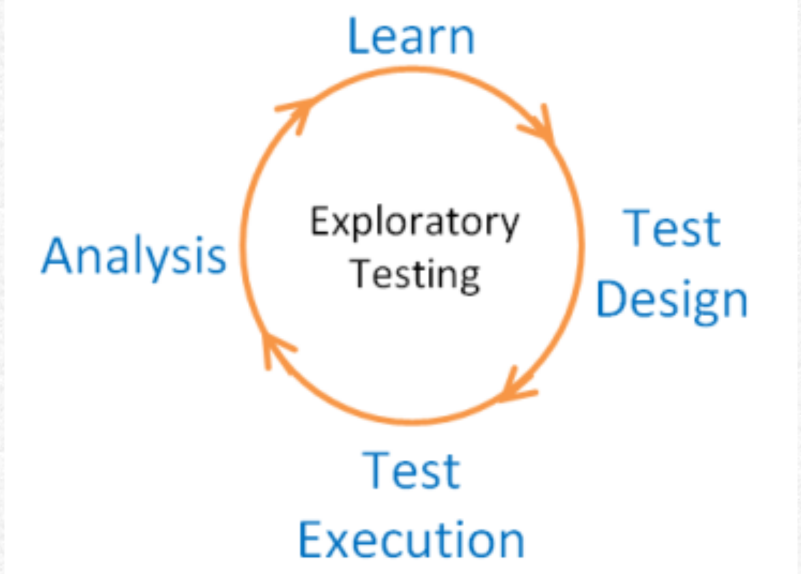


Coded UI Tests

Coded UI Test
Select how to create a coded UI test
Launch the recorder to generate code
Use the UI Control Locator
Use an existing automation strip
Cancel

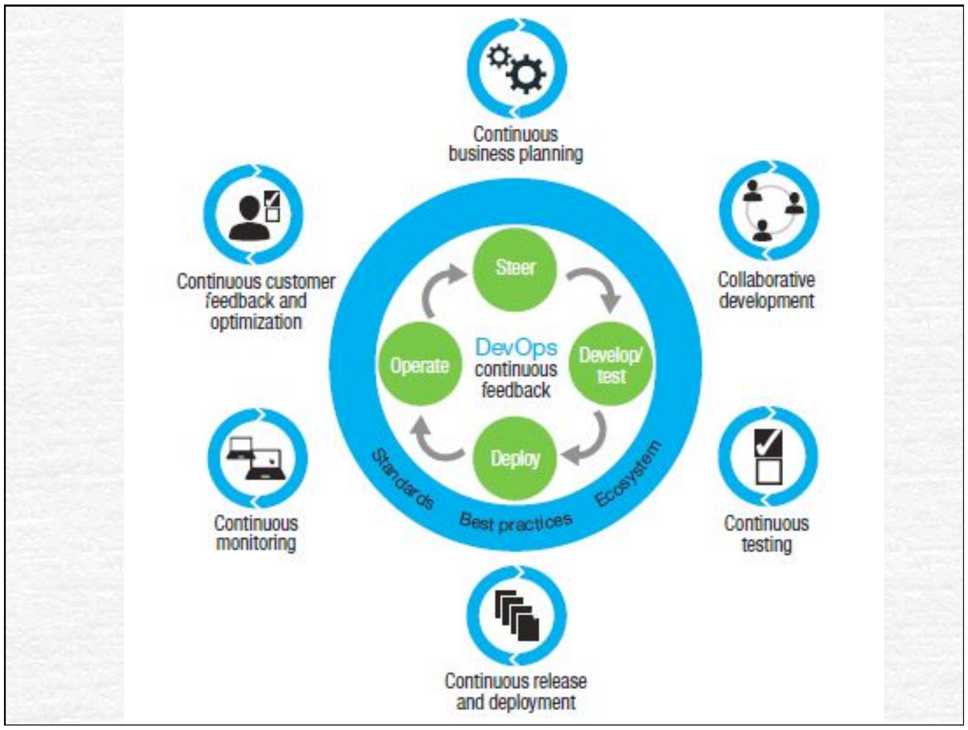
Automated Non-Functional Testing

Manual Testing



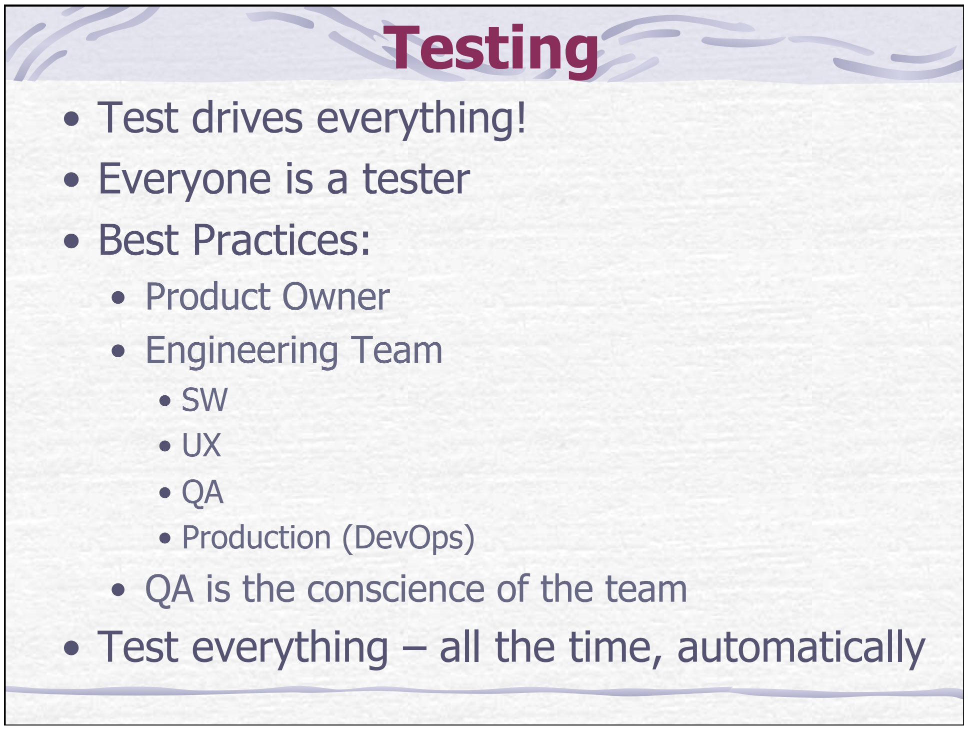
Usability Testing

WE COULD DESIGN THE PRODUCT WITH A SIMPLE POINT-AND-CLICK INTERFACE...
OR WE COULD REQUIRE THE USER TO CHOOSE AMONG THOUSANDS OF POORLY DOCUMENTED COMMANDS, EACH OF WHICH MUST BE TYPED EXACTLY RIGHT ON THE FIRST TRY.
BEAR IN MIND, WE'LL NEVER MEET A CUSTOMER OURSELVES.
MAKE IT SO THEY HAVE TO REBOOT AFTER EVERY TYPO.


Continuous business planning
Continuous customer feedback and optimization
Collaborative development
Steer
Operate
DevOps continuous feedback
Develop/ test
Deploy
Standards
Best practices
Ecosystem
Continuous monitoring
Continuous testing
Continuous release and deployment

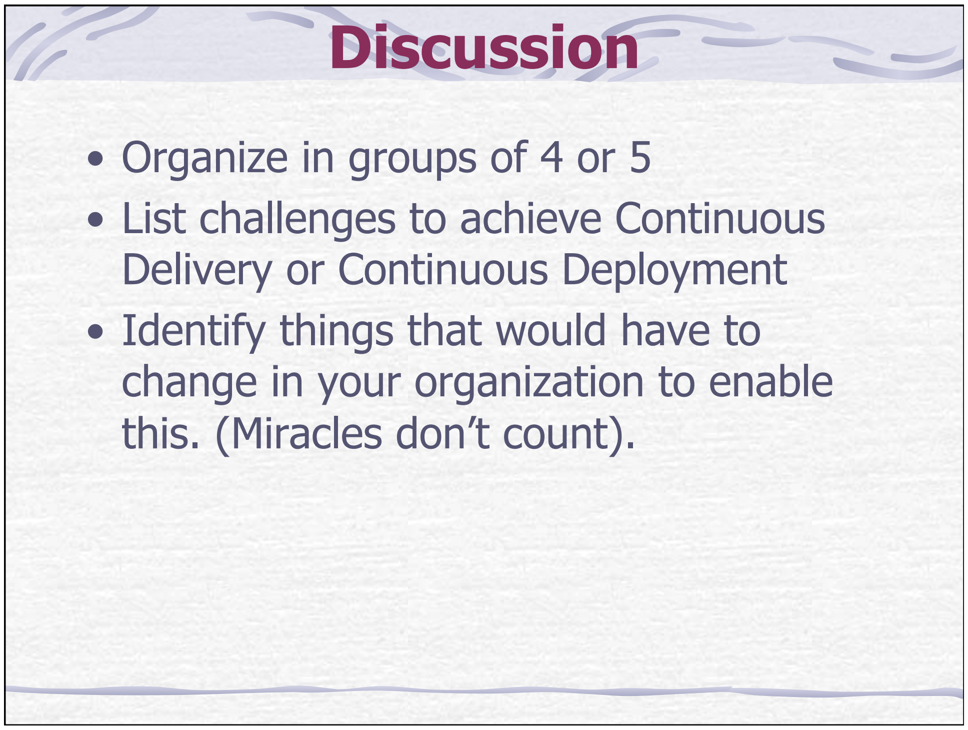
Testing Implications

- Need only one environment before production
- Test in Dev
- Don’t wait for feature complete
- Don’t wait for stable builds
- Definition of Done is critical
- Test key items quickly



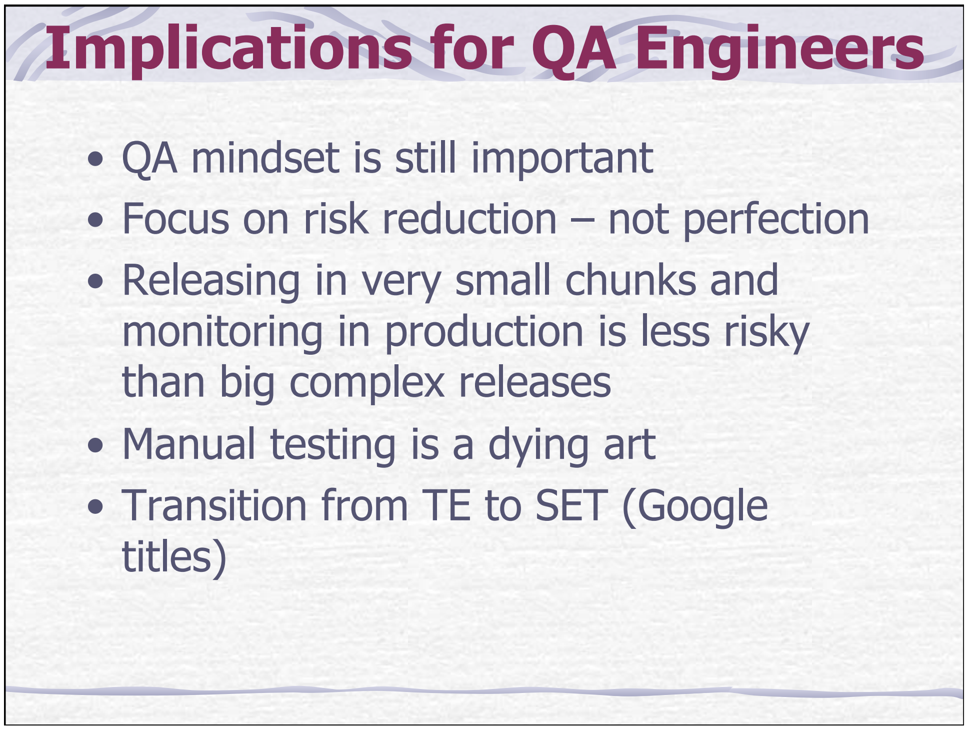
Testing

- Test drives everything!
- Everyone is a tester
- Best Practices:
  - Product Owner
  - Engineering Team
    - SW
    - UX
    - QA
    - Production (DevOps)
  - QA is the conscience of the team
- Test everything – all the time, automatically



Discussion

- Organize in groups of 4 or 5
- List challenges to achieve Continuous Delivery or Continuous Deployment
- Identify things that would have to change in your organization to enable this. (Miracles don’t count).



Implications for QA Engineers

- QA mindset is still important
- Focus on risk reduction – not perfection
- Releasing in very small chunks and monitoring in production is less risky than big complex releases
- Manual testing is a dying art
- Transition from TE to SET (Google titles)