

# Defining Testing Strategies for Modern Times



Joel Montvelisky  
Chief Architect & Co-founder - PractiTest

# About Me!

## QA / Testing

(last 24 years)

- Tester
- QA Manager
- Blogger / Podcaster
- Speaker
- Consultant
- Solution Architect



**Joel Montvelisky**  
Follow: @joelmonte

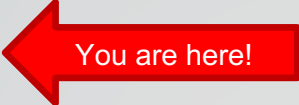
## PractiTest

- Co-Founder
- Chief Solution Architect

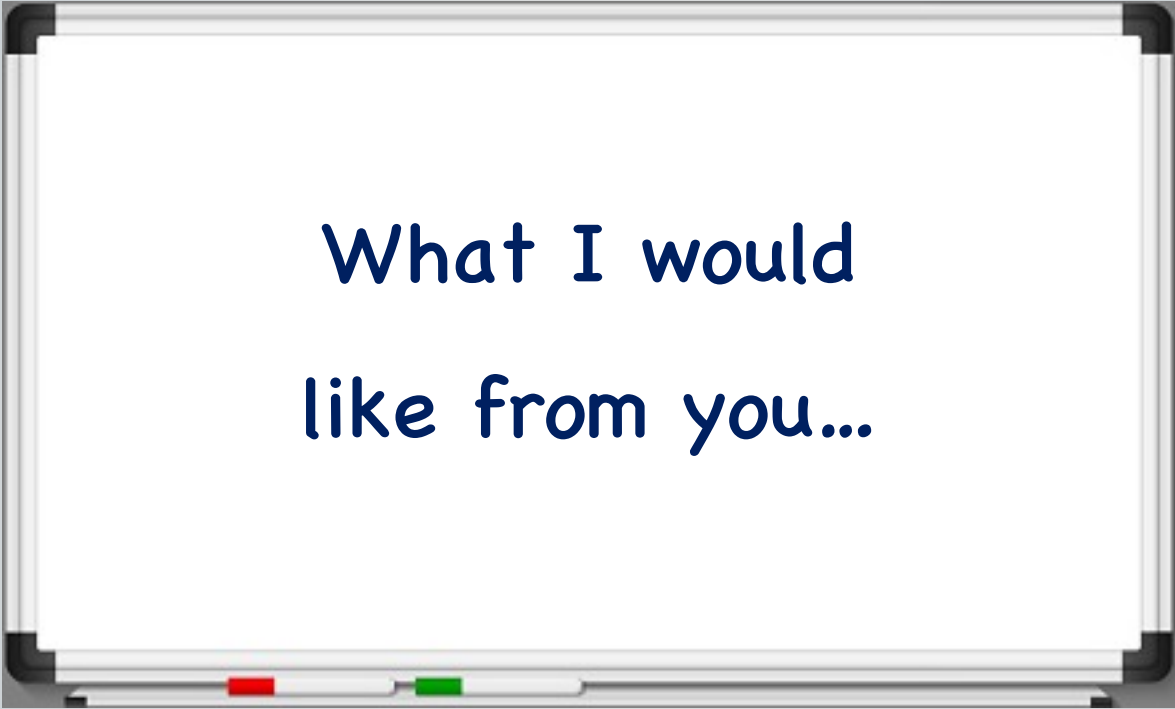
## Other Stuff

- State of Testing™
- Online Test Conference

# Agenda

- Quick introduction 
- Mutual expectations
- “Modern Times”?
  - Break -
- Evolution is good for you!
- Modern Times Testing Strategies

What do you want  
to get out of this session?



What I would  
like from you...

# Let's talk about “Modern Times”

# Question:

*What has changed for you in the last 5-10 years?*

- The way your company conducts business
- The tools you are using
- The structure of your teams
- The work culture of your company
- Your responsibilities as a tester

# A lot has changed in the last decade.

- ✓ Wide adoption of Agile, and now DevOps
- ✓ Wider range and availability of Testing Tools
- ✓ Shifting Left & Right

And more



# Methodological Migration from WF to Agile to DevOps

# WF to Agile to DevOps:

## The Business Perspective

- More and faster changes
- Users can leave your service quickly and easily
- Users are more tolerating of some issues – e.g. non-critical crashes while they are less tolerating of other issues – e.g. slow response times

# WF to Agile to DevOps:

## Our Production Environments

- Support for a single backend
- Easier scalability
- Responsibility for IT processes (e.g. Disaster Recovery)



# WF to Agile to DevOps:

## Deployment Strategies

- More control how we deploy and to whom
- We can use deployments as part of our Exploratory Testing approach
- Options to rollback problematic deployments

# WF to Agile to DevOps:

## Visualization

- We can see what is happening LIVE.  
Find and fix issues before they are reported by users!
- Analyze real and complete usage patterns.
- Access to real user data



# Tools for Testing

# Tools in Testing:

## Specific Non-Testing Tools

- CI – not only for Development
- Deployment
- Monitoring
- AI (?)

# Tools in Testing:

## Many More Testing Tools

- Better & cheaper tools
- Cheaper tools
- Lower learning curve (at least in theory)



# Tools in Testing:

## A Tool Revolution

- Organizations have between 4 and 10 different tools.
- Switch from Single-Vendor to Best of Breed.
- Isolated execution & reporting >> Siloed tools.

# Shifting Left & Right — Closing the Process Loop

# Shift Left & Right:

## Un-engineered Chaos

- Agile is anything but ordered and simple
- Smaller teams, but in the context of larger organizations
- Political & communication challenges
- Did I mention “Global Teams” as well as WFH?



# Shift Left & Right:

## Starting Earlier

- Be fast at catching inconsistencies
- Preparation time is “really short”
- Full Understanding & Prioritization is critical
- Be ready and prepared to react quickly



# Shift Left & Right:

## Bringing feedback from Production

- Who is working with what features? What is not in use? Interesting patterns?
- Are there bugs / exceptions / unexpected behavior?
- Real time evaluation of usage assumptions / environments / integrations.

# Shift Left & Right:

## Evaluate Quality post-Deployment

- Real stats of Escaping defects
- Response time & other live metrics production metrics
- Other stats: NPS, Conversions, Churn, Completion of tasks

# Summing Up

-

**We're not in the early  
2000's or 2010's anymore**

# Not your father's testing team...

- ✓ We're part of the Development Team
- ✓ Or job is still to Provide Visibility on the Quality of the Product  
as well as Projections on the Project Completion timelines
- ✓ We are working on a Chaotic environment
- ✓ You have more responsibility, power and influence – if you want them



# BREAK

# **EVOLUTION:**

**Adapt  
or  
Risk Become Irrelevant**

# I was in Dubai (UAE) last week



# I was in Dubai (UAE) last week

1. Nomad Bedouins
2. Pearl farmers
3. Oil
4. Tourism
5. Technology



# Testers have evolved over the years too

1. End users playing with the system
2. Professional button clickers
3. Functional Testing Professionals
4. Quality Architects

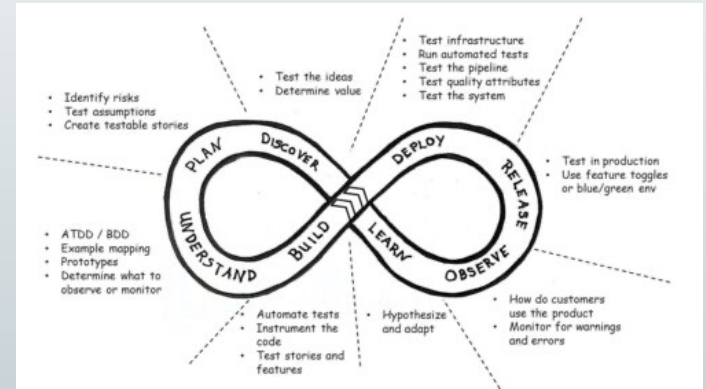
**How has your testing  
evolved over the years?**

# Testing Strategies & Approaches for Our Modern Times (my partial list)

# Holistic Quality

## The Role of the Quality Architect

- ❑ Janet Gregory & Lisa Crispin
- ❑ Look at all your Operation, understand what quality aspects match each process
- ❑ It has to be adapted to your process and constraints



<https://janetgregory.ca/>



# Methodic Shift Left Testing

## Work with User Stories & Epics

- All points covered? No inconsistencies? Deployment considerations?
- User Story Value Hypothesis / MVP Validation criteria
- Feature instrumentation and other production concerns

# Risk-based analysis of testing

Because you can't test everything

- What to test? / What not to test? / What to test in production?
- When to test?
- How to test?
- Who to test?



# Testing while developing

Start as early as possible

- To be done while development is still in process
- Paired testing & discovery sessions
- Internal Acceptance Criteria – to avoid surprises
- Make sure what is being developed matches what's on the User Story

# Testing mentoring & coaching

Testing is everyone's responsibility

- ❑ While testing with devs, explain what and why you are doing stuff
- ❑ Paired testing sessions
- ❑ Share testing artifacts
- ❑ Formally assign testing tasks to Devs



# Rapid-fire testing

Focused testing covering more areas

- ❑ Define quick testing objectives on important user stories
- ❑ Be focused and go deep
- ❑ Document activities to avoid duplication



# Test Orchestration

Coordinate and account for all testing results

- Account for all testing: Unit, Integration, Functional, UAT, Monitoring, etc
- Unified and normalized reports for stakeholders
- Align efforts from all teams and testing artifacts
- Find Over-Covered and Under-Covered areas in the product

# Quality Dashboards

If a tree falls in an empty forest...

- ❑ Why test if you do not make anything with the results?
- ❑ Clear visibility for your stakeholders' concrete needs / their Burning Questions
- ❑ Focus on Safe, Risky and Problematic areas



# Go / No Go for User Stories

## Clear decision making information

- Support decisions for your team around stories and releases, based on clear and simple criteria
- Capture evolving issues and criteria
- Incorporate inputs from as many sources





# Monitoring 1 / 5 / 30 / 90 min

## Different issues at different stages

- Never “push the button and run” / deploy at 6 PM
- Understand what issues are immediate and which take time to brew
- Define checkpoints
- Keep track of issues and metrics

# Alerts & user behavior

## Part of testing in production

- Check what is going on life!
- Find exceptions and fix bugs based on production data
- Learn about how people are working from examining the data

# Escaping defects

## Analyze where to improve

- Understand how you are doing as a testing team
- Examine your risk assumptions
- Find areas to improve



**And this is only my partial list...**

**What are you taking  
with you from  
today's session?**

# Thank you!

[joel@practitest.com](mailto:joel@practitest.com)