3 Keys to Performance Testing at the Speed of Agile

Matching the imperative of high-performance with the pace of modern software delivery

Neotys

Housekeeping

1. This session is being recorded
2. Discussion in chat panel
3. Q & A at the end of session
4. Follow-up email with links to presentation

Neotys

What We'll Cover

1. Agile Challenges and Anti-patterns
2. The Goal: ~~Potentially~~ Confidently Shippable Product
3. 3 Keys to Efficient Performance Testing in Sprints
4. Q&A Roundtable

Neotys

Today's Presenter

Paul Bruce
Sr. Performance Engineer, Neotys USA
@paulsbruce @neotys
linkedin.com/in/paulsbruce

Quick Poll: When do you measure system performance?

Neotys

[multiple choice]

- Just before MAJOR releases
- Just before MINOR releases
- A few times, throughout sprints
- As part of a Continuous Delivery pipeline
- No schedule, project-based, as-needed

Neotys

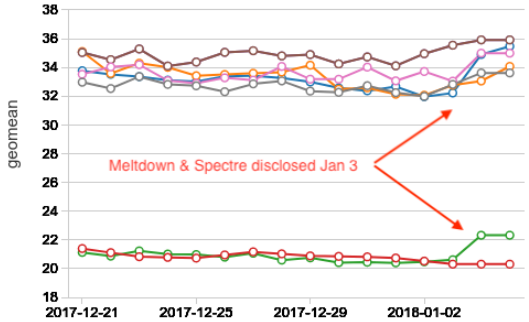
System performance is imperative to all business models

Neotys

max — 14%

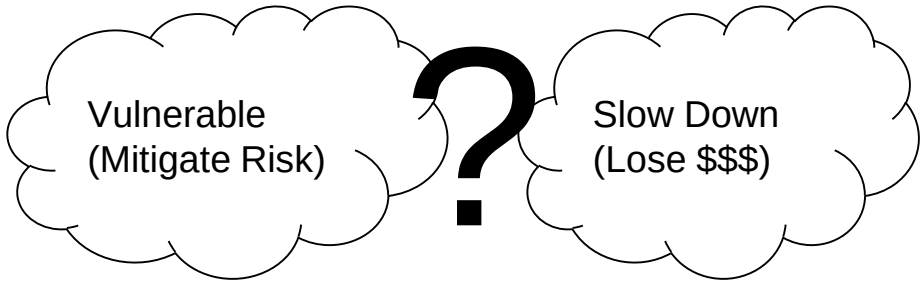
avg — 6%

min — 2%



1 of every 3 organizations
take longer than 30 days
to patch/resolve security issue.

Qualys - bit.ly/2D6OJWU



Intel Newsroom - intel.ly/2B4DWL3

Databricks -bit.ly/2Dd51h4

Neotys

What *kind* of performance do customers expect?

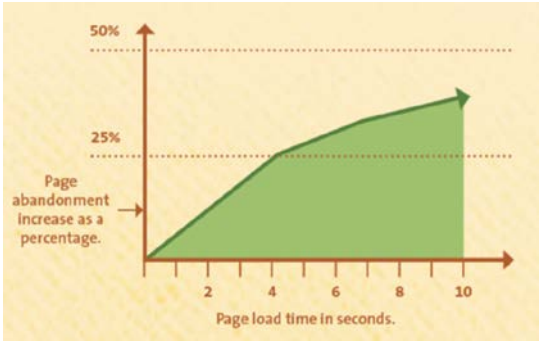
Consistency: across all digital properties (web, mobile, API…

Fast activity completion: checkout, subscribe, approve

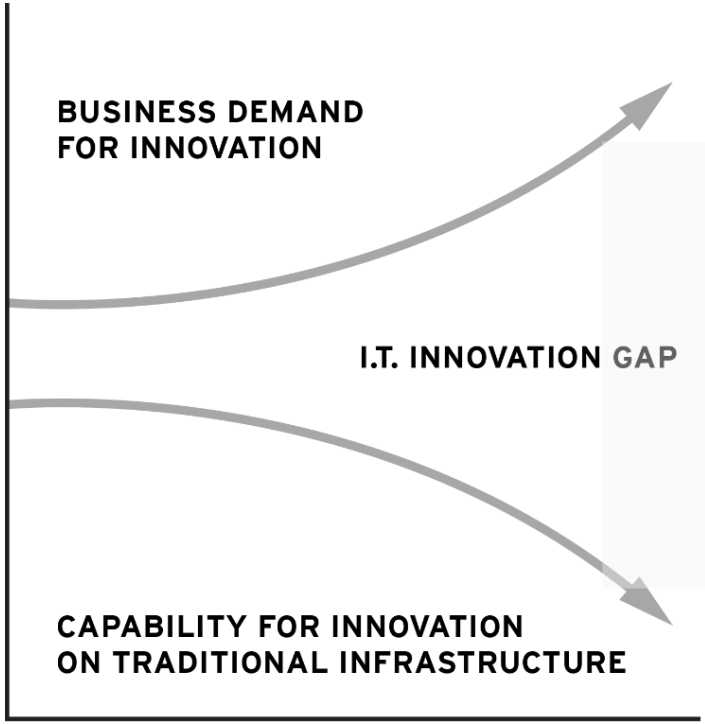
Abandonment: page or mobile app uninstalls

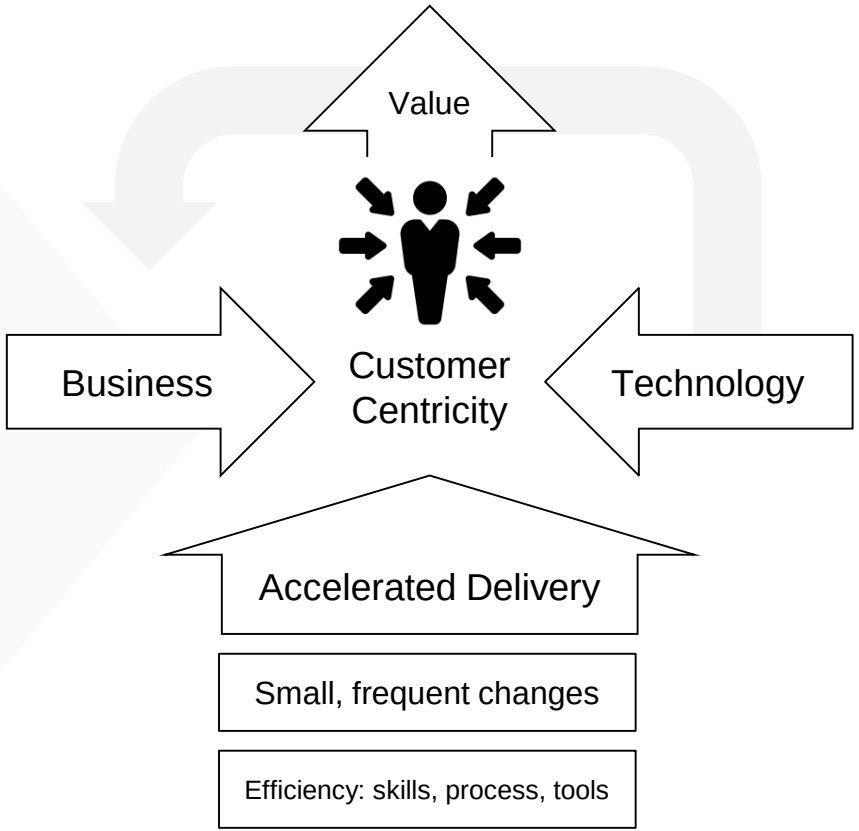
Ubiquitous connectivity: WiFi, mobile, broadband…

Average load time for all elements on the page



“Faster” Isn’t Good Enough Anymore





Agile “faster theater” anti-patterns

Neotys

“We didn’t bother with scalability because they wanted an MVP ASAP…”

“The stakeholders didn’t say they want to integrate this with [that other service]…”

“We’ll deliver documentation…if there’s time at the end of the sprint.”

“Security is for a hardening stage…which as usual, got eaten up by bug fixing…”

“We’ve been doing stand-ups for weeks, and no one mentioned performance criteria!”

“That isn’t in the definition of done, so the customer shouldn’t expect it in the product.”

“That’s someone else’s area of expertise…I’ll assign the task to them next.”

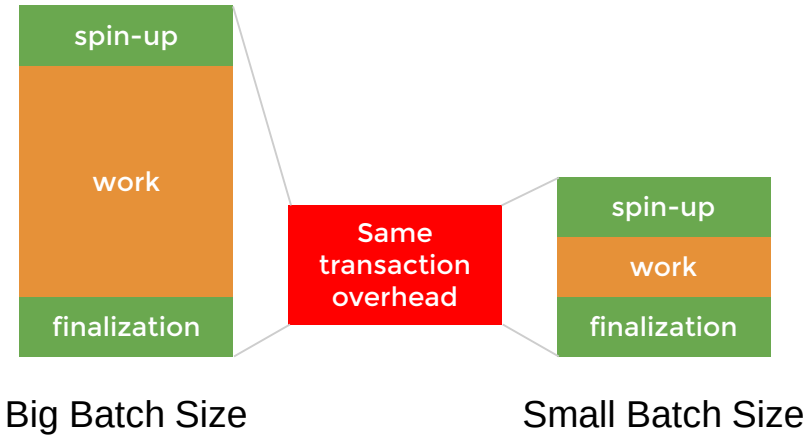
Neotys

Real Work Is Lumpy, but Still Necessary

Neotys

Finely-tuned batch sizes and automation solve many (but not all) problems.

- Resource intensive human-only tasks
  - Planning
  - Estimation
  - Prototyping
  - Code reviews
- Resource intensive automation tasks
  - Regression testing
  - Migrations / upgrades
  - Performance testing
- Trend-driven analysis
  - Build and release metrics
  - Production actuals (e.g. A/B testing)
  - Customer feedback / reviews / NPS



"Right fit" is the biggest challenge.

Neotys

The Goal: Confidently Shippable Product

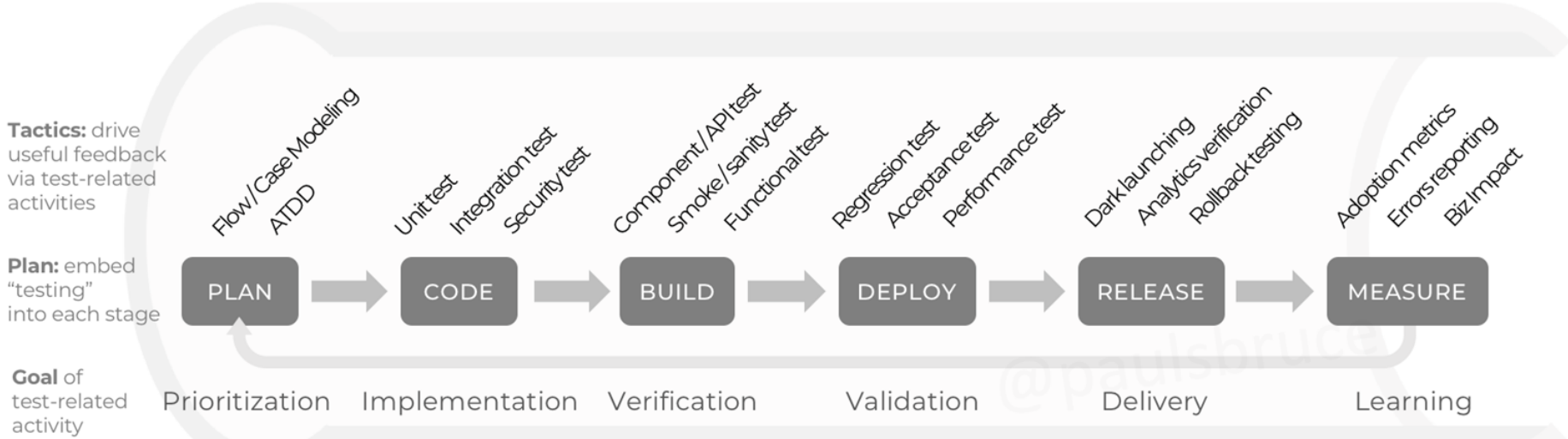
1. What is "Progressive Testing"?
2. What needs to be performance tested (and when)?
3. The "right" performance test for the job
4. Example triggers and schedules

Neotys

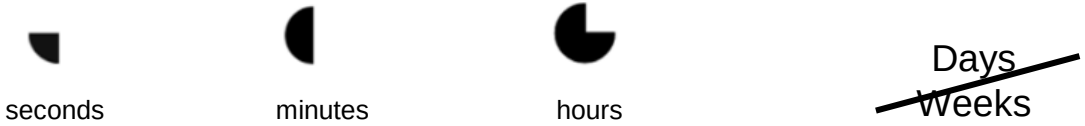
Progressive Testing Across the Delivery Pipeline

Neotys

For: Feature X… API update Y… Patch N…



Time budget for testing:

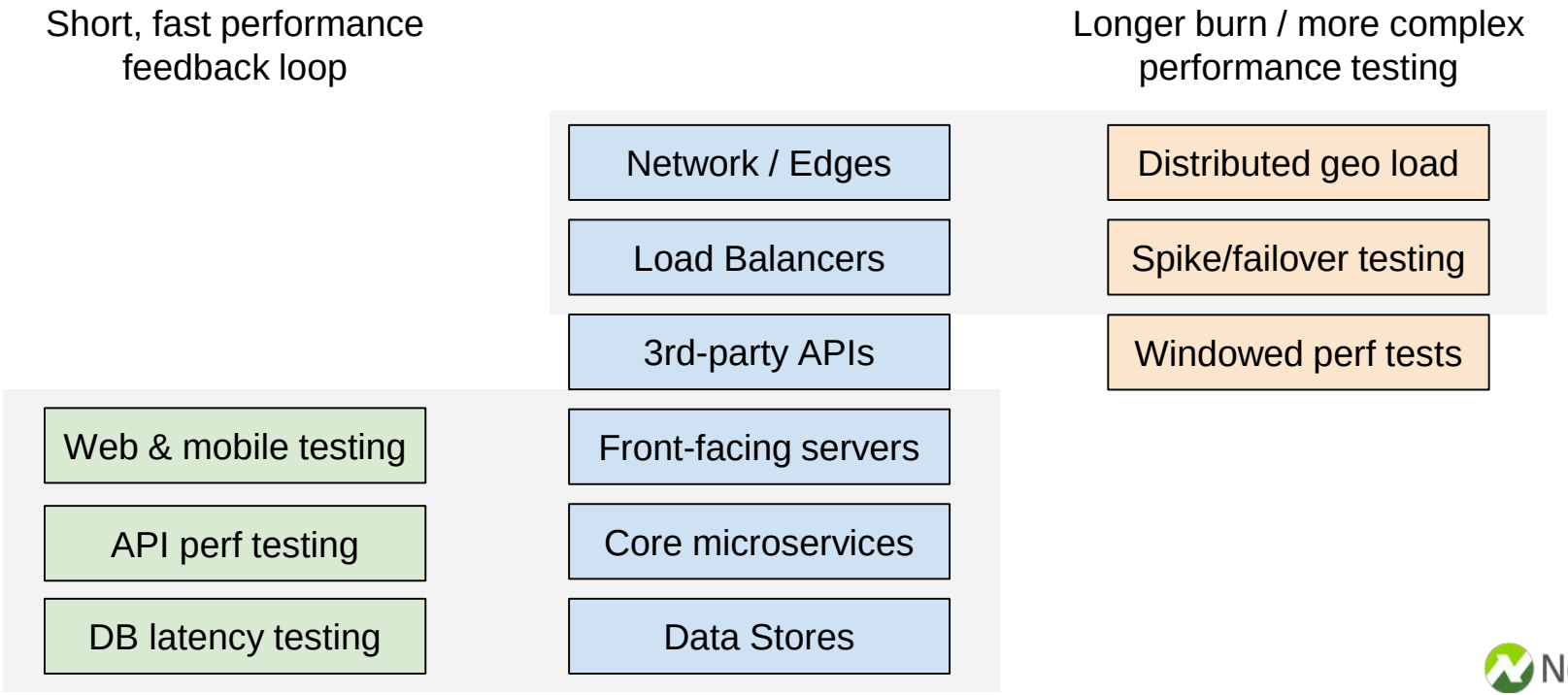


Neotys

What needs to be load tested?

Neotys

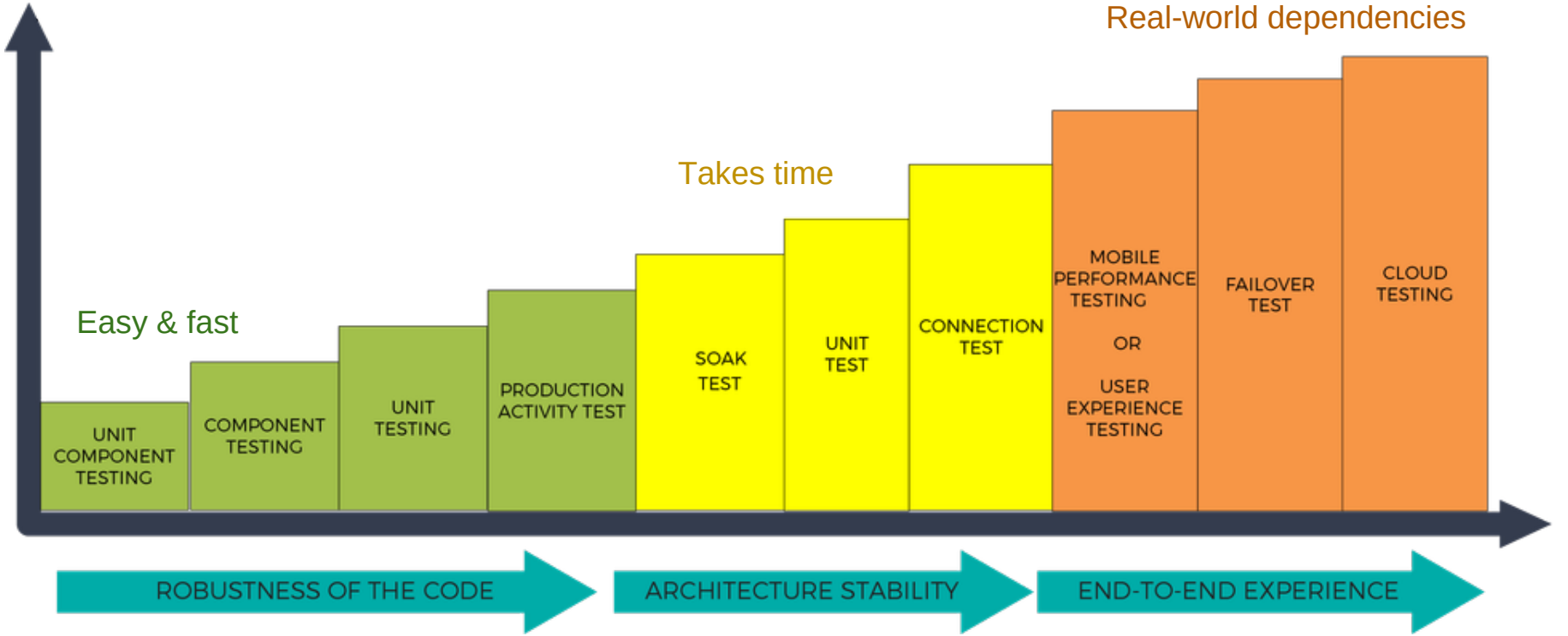
Each layer must be tested for performance:. API, database, web/mobile, networks, 3rd parties, cloud infrastructure, failover/recovery behaviors.



Neotys

The “right” performance test for the job

Neotys



Neotys

The “right” performance test for the job

Neotys

Development and Architecture

- **Load** (does it handle N users?)
  - Cloud migration baselines
  - Build-over-build performance trends
  - Meeting contractual SLA obligations
- **Soak** (are there leaks or degradations over time?)
  - Server RAM, CPU, disk
- **Spike** (how does it handle dramatic changes?)
  - Load balancing
  - Rate throttling
  - Garbage collection

Operations and Change Management

- **Stress** (where is the max?)
  - Capacity planning
  - EUX degradation and ajax timeout handling
- **Configuration** (is this tweak better or detrimental?)
  - Routing, SSL termination, database tuning
  - Logging, retention policies
- **Network** (what latencies are in the pipes?)
  - 3rd-party API SLAs
  - Service distribution (architectural latency)
  - Multi-cloud deployments

Example: Triggers and schedules for performance testing

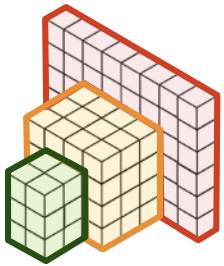
Neotys

Event	Max CI duration	DB	API	Web & mobile	3rdParty	Infra	Network	Regression
Code check-in	10-15 min	Sm	Sm	Sm				
Merge	15-30 min	Med	Med	Med				
Overnight	1-4hrs	Med/Lg	Med/Lg	Med/Lg				Sm
Weekly		Lg	Lg	Lg	Sm	Med	Med	Med
Pre- release		Lg	Lg	Lg	Lg	Lg	Lg	Lg

Large

Medium

Small



Test Coverage (Sm / Med / Lg):

- Volume (VUs)
- Scope (# flows)
- Conditions (geo, wifi/LTE)

"Right fit" is the biggest challenge.

Neotys

3 Keys to Efficient Performance Testing

1. Integrate performance into planning phase
2. Establish “right-sized” feedback loops
3. Reduce waste in scripting and analysis

Neotys

Key #1: Integrate performance into planning phase

Neotys

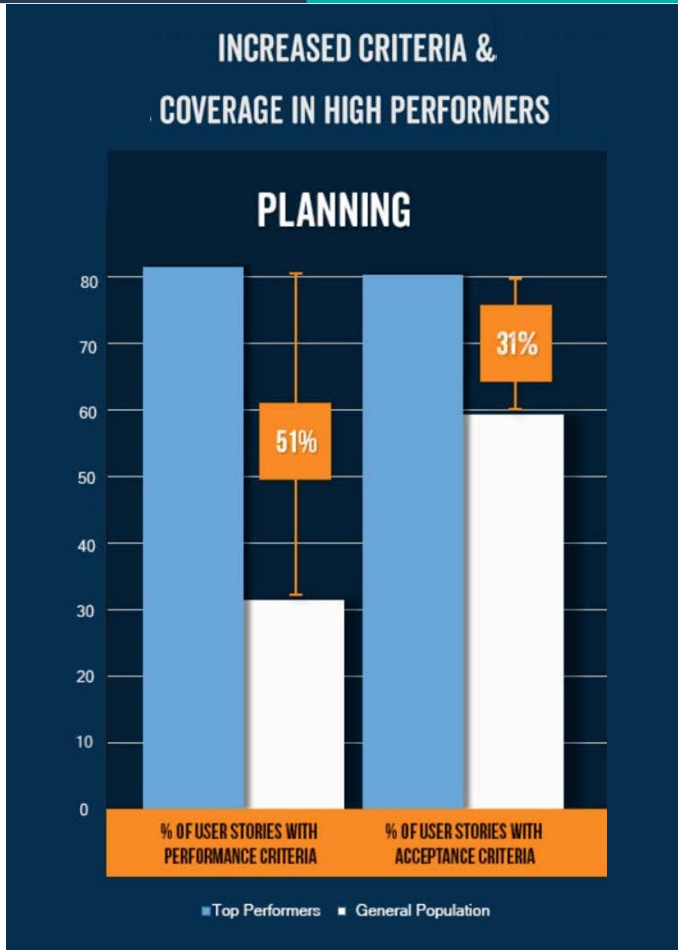
2017 study:

bit.ly/performance-feature

- Release every other day; 2x user stories w/ performance criteria

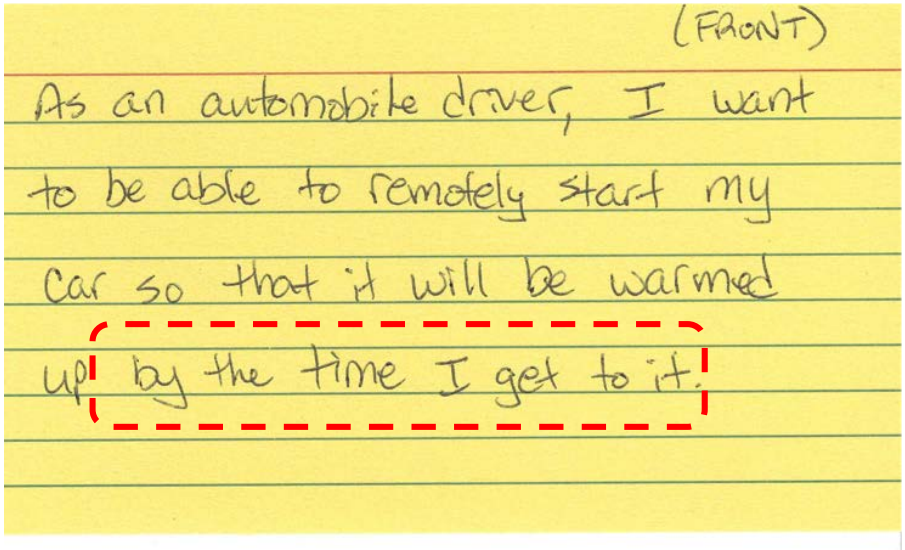
How to “bake performance in” to user stories?

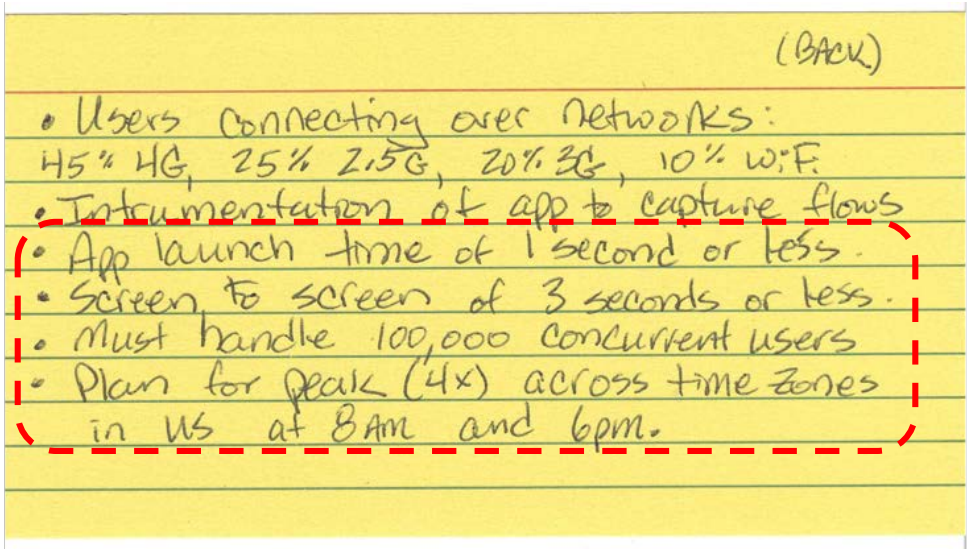
- “Back of card” / DoD - performance criteria
- “Think 4th dimensionally” → Time
- Concerns and guidelines, not testing performed
- Performance-specific:
  - Concurrency: “how many users at the same time?”
  - Conditions: “which devices from where?”
  - Capacity: “what resources will this take?”



Key #1: Integrate performance into planning phase

techbeacon.com/how-build-performance-your-user-stories





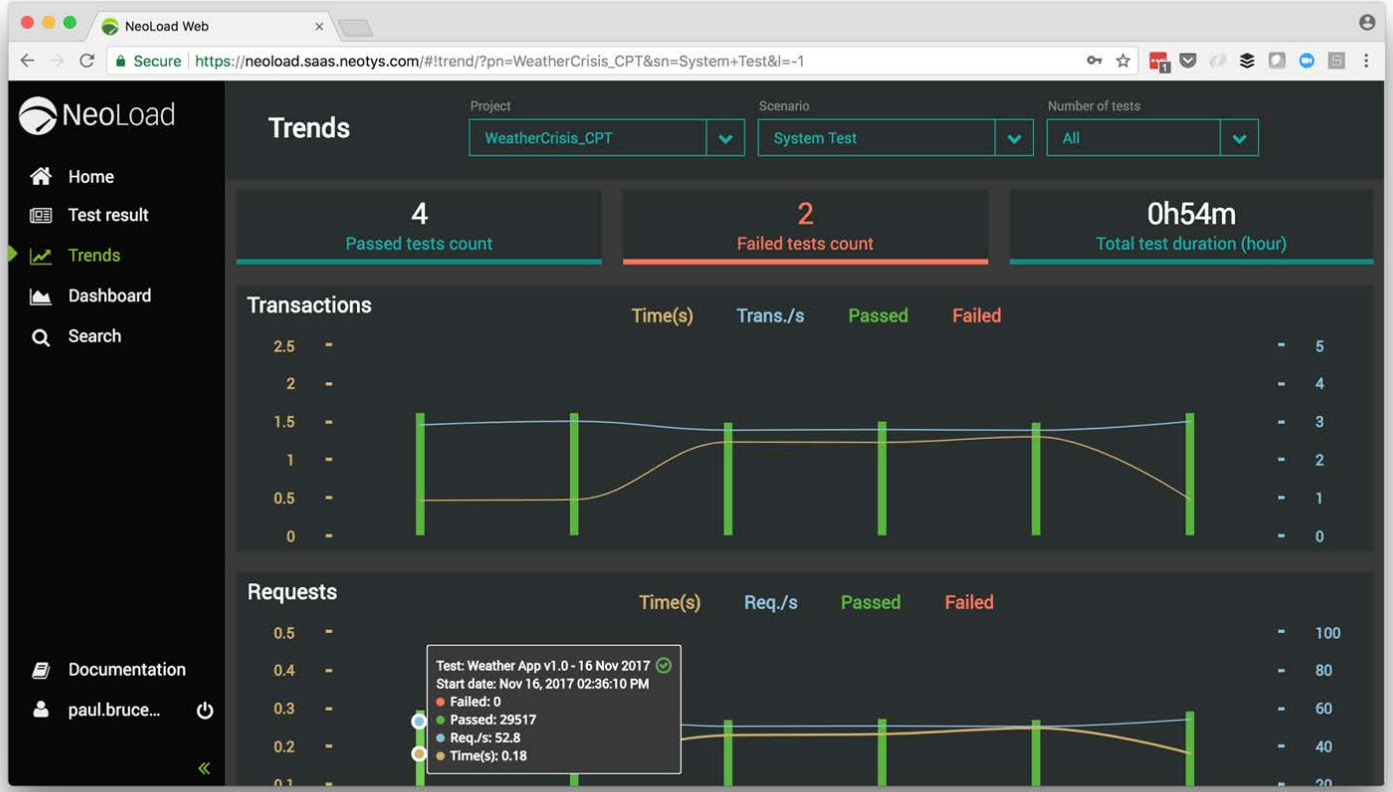
Todd DeCapua - @appperfeng

Key #2: Establish Right-Sized Feedback Loops

- Small load test on new features → **Value vs. time cost**
- Critical-path performance regression → **Risk Prevention**
- Baselines and comparisons accessible to everyone → **Early Detection**
- Including metrics that tell the complete story: → **End-to-end visibility**
  - The “efficacy” of the test (load health and bias)
  - The “impact” side of the story (server monitoring)
  - The “customer” side of the story (end-user experience)

Neotys

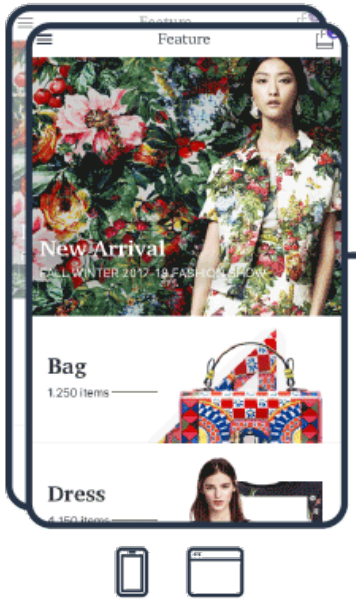
Key #2: Establish Right-Sized Feedback Loops



Key #3: Reduce Waste in Scripting Time

- Record through a browser, low-code approaches → **Easy & Intuitive**
- Carry customizations during re-scripting / recording → **Efficient**
- Build library of reusable test assets → **Consistent**
- Reuse existing functional test assets ("Self-healing" load scripts) → **Realistic**
- Share test assets across teams and engineers → **Collaborative**

Key #3: Reduce Waste in Scripting Time



Recording

- YourWorkflow
  - Init
  - Actions
    - Navigate To: /views/map
    - Click: Weather alert v1.2
    - Click: SEARCH
      - /search
        - /search
    - Click: SUBMIT
    - Click: svg
  - End

3 Keys to Efficient Performance Testing

1. Integrate performance into planning phase
2. Establish “right-sized” feedback loops
3. Reduce waste in scripting and analysis

Neotys

What We've Covered

1. Agile Challenges and Anti-patterns
2. The Goal: Confidently Shippable Product
3. 3 Keys to Efficient Performance Testing in Sprints

Neotys

Q&A Roundtable

Paul Bruce
Sr. Performance Engineer, Neotys USA
@paulsbruce @neotys
linkedin.com/in/paulsbruce

Neotys