# Baubak Gandomi

## Mutational Tests

PACIFIC NW SOFTWARE QUALITY CONFERENCE

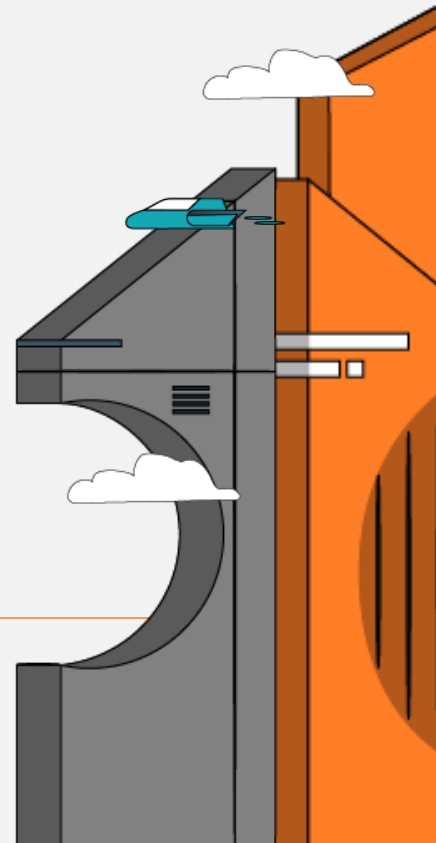
THE FUTURE IS NOW

PNSQC.ORG

OCTOBER 14-16 2024

# Let’s Gather their Interest

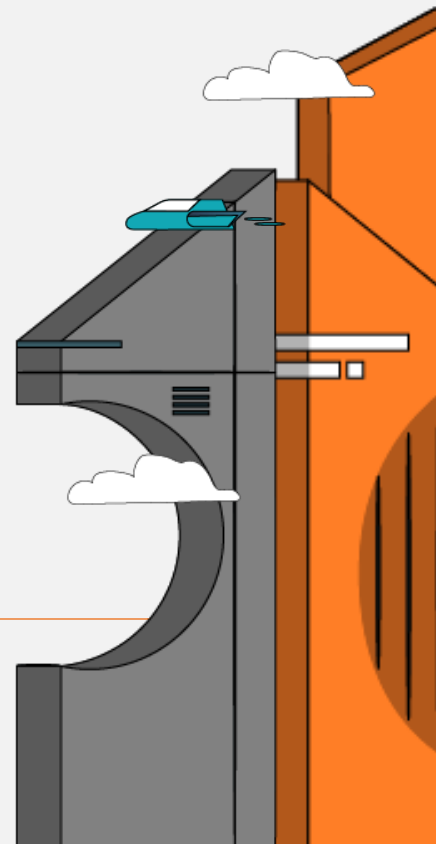
- Why bother in the first place? Why Should you Bother?
- The Cool things we did
- We Mutated
- Look at our super powers
  - Data
  - Events - Chaos
  - More Chaos



## Agenda

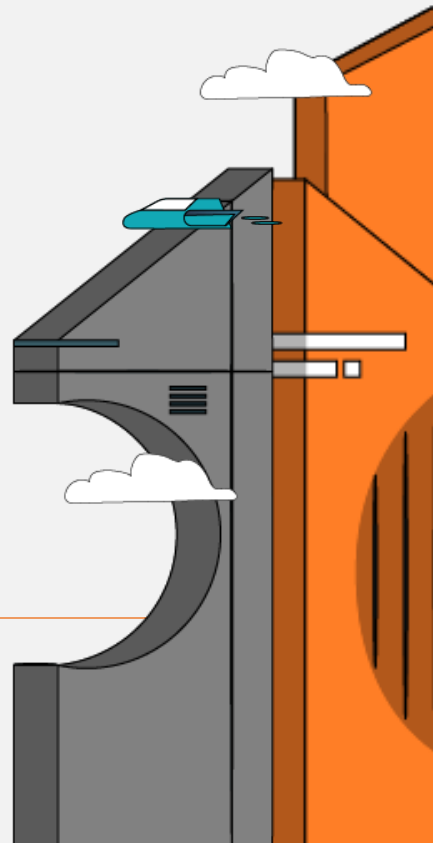
# Who the @&#$ is Presenting This?

- Too many years in the Industry
- 18 as a QE
- 11 at Adobe as a QE Architect
  - Testing complex problems
  - Cross Team and Technology Testing
  - Programing as Literature
- 2nd Participation at PNSQC



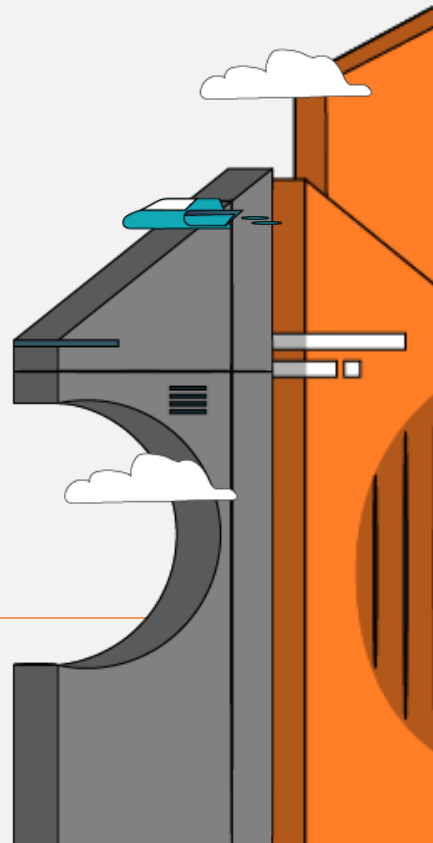
# A New Vision for Scenarios

- A template
- Change and adapt when needed
- The test could be much more than the scenario

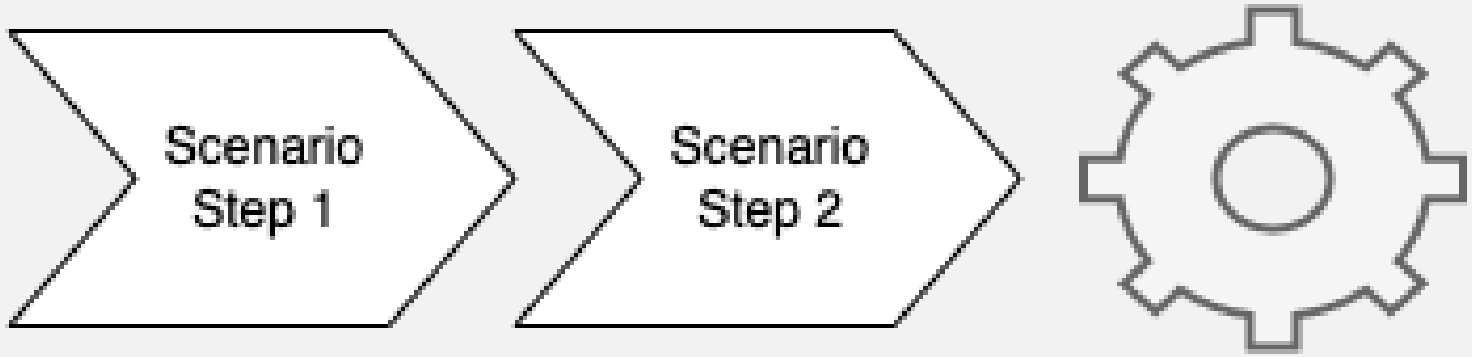


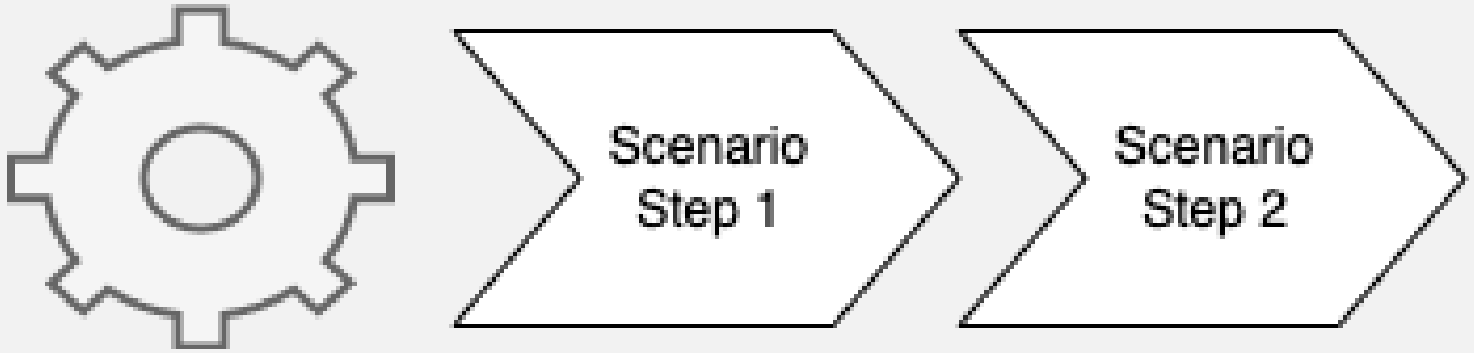
# Once Upon a Time

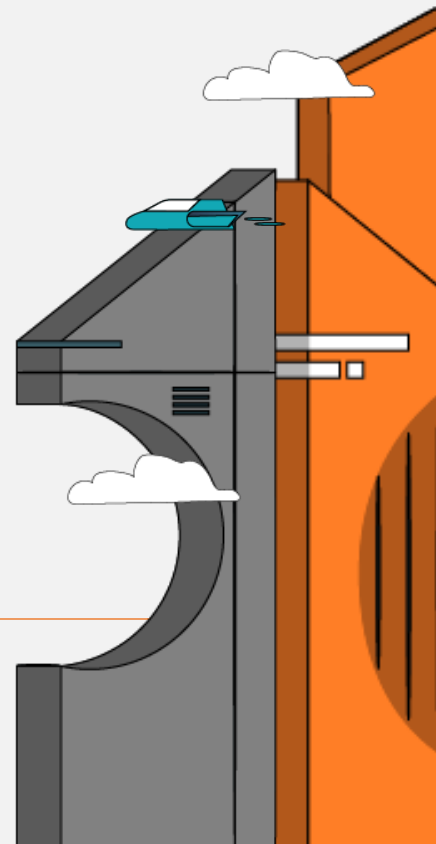
- A Problem Testing Upgrades



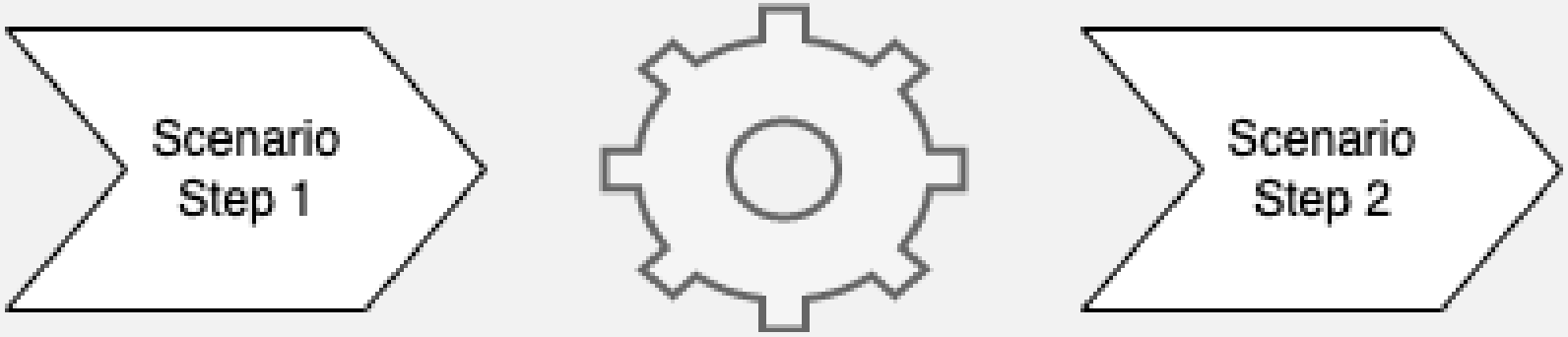
# Common Approaches

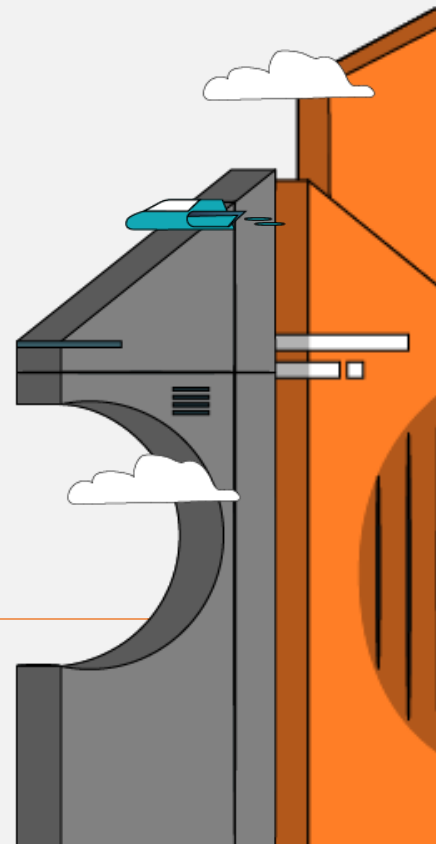


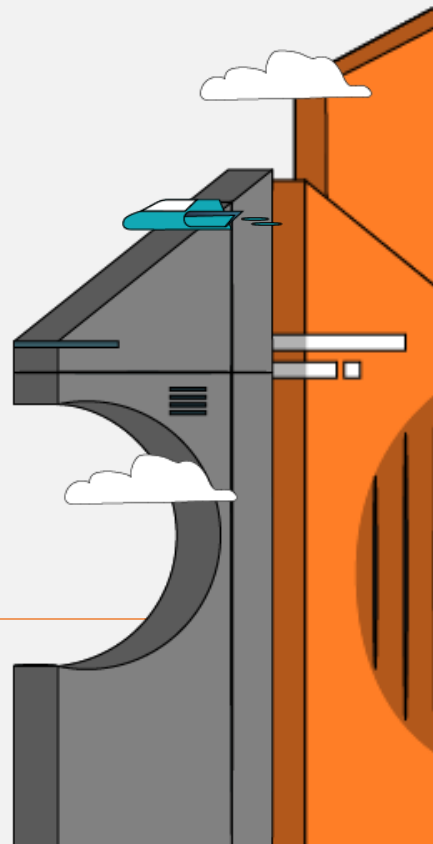
# A Manual Approach





# Data is a Problem

- Data changes between versions
- Enriching data for scenarios
- How do we update this data?

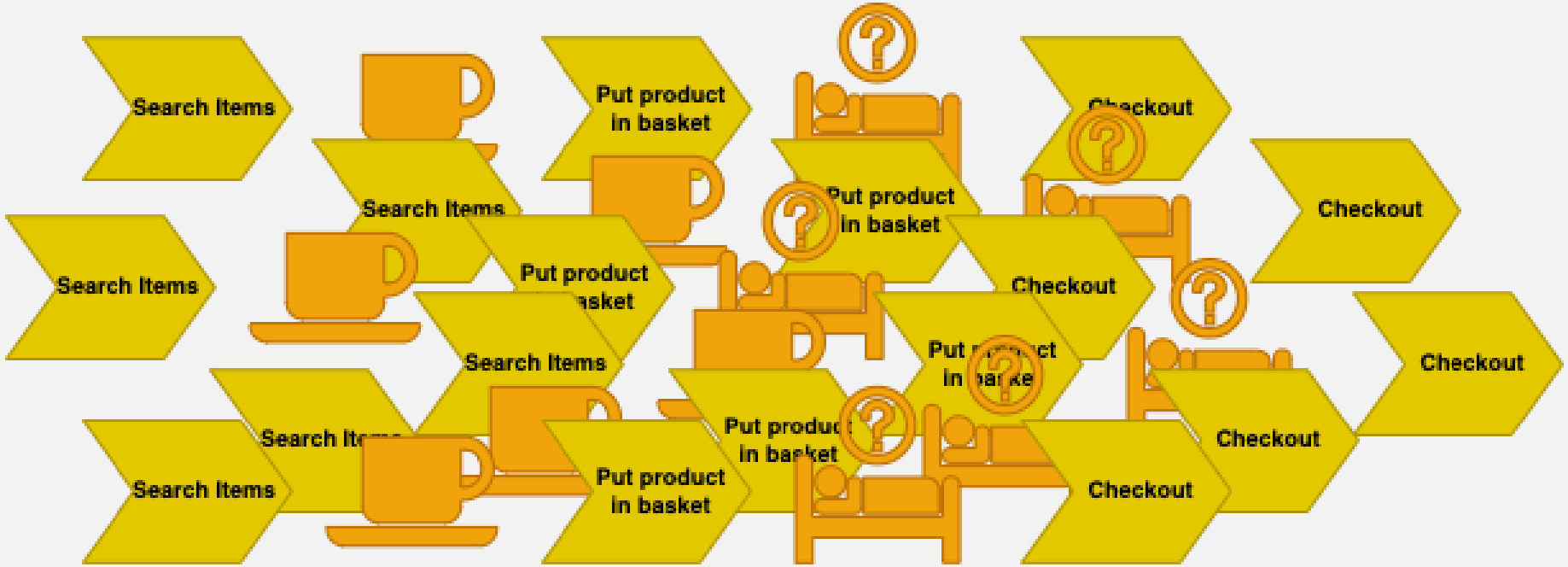


# Time is Problematic

- Scenarios are, in reality, not Atomic
- You are not alone!

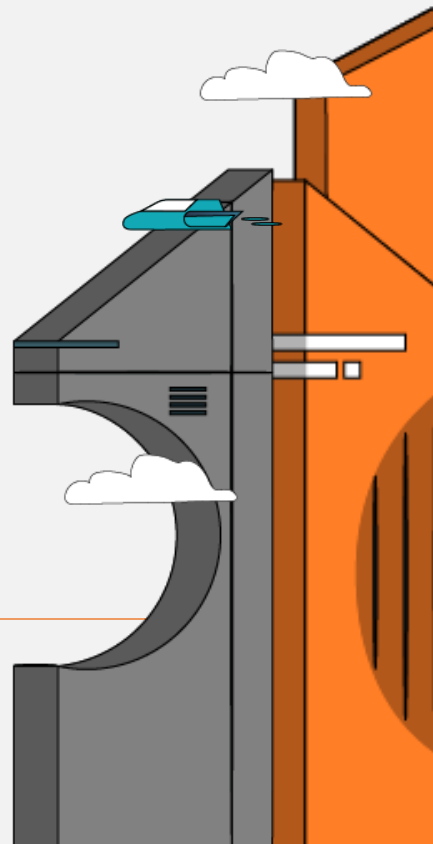
# Time is Problematic

- Scenarios are, in reality, not Atomic
- You are not alone!

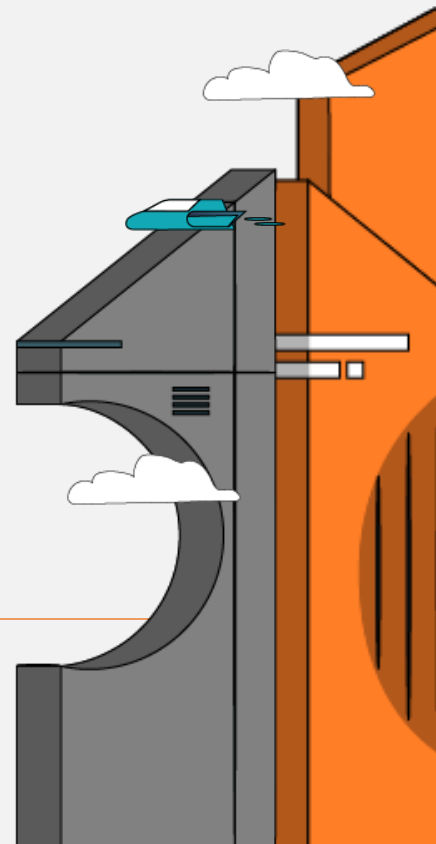


# The Cool Things We Did

- Enter Phased Tests
- Paving the way for mutations

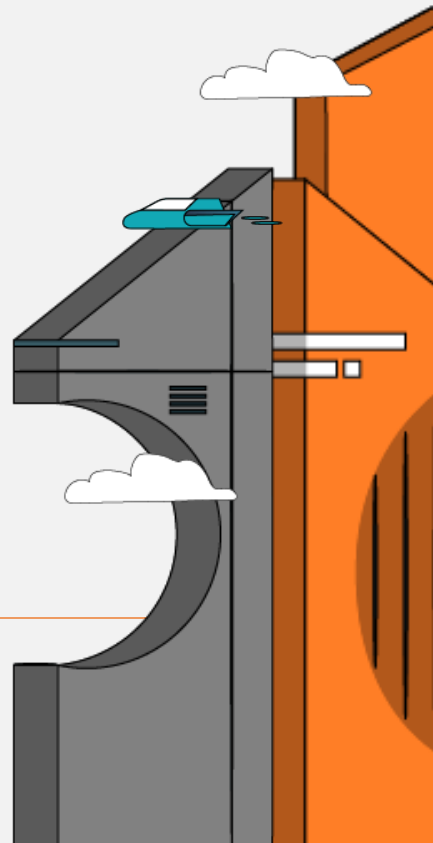


# Phased Tests

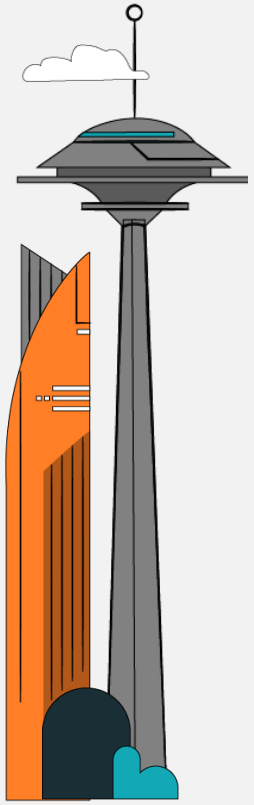


# Phased Testing Requirements

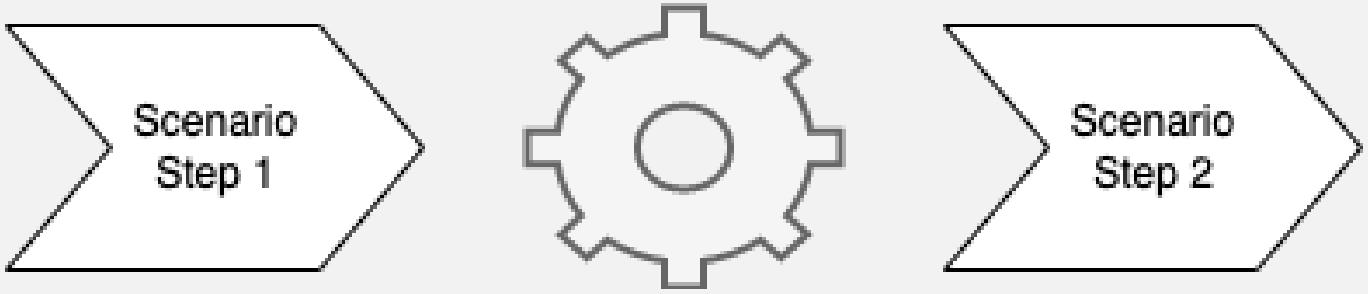
- Address the issues we highlighted
- No dedicated test scenarios



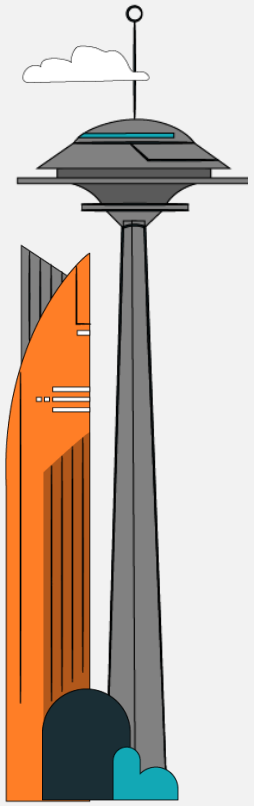
# Phased Tests



- A Scenario that can be interrupted
- Shuffling
- Managing Context
- A Testing Design



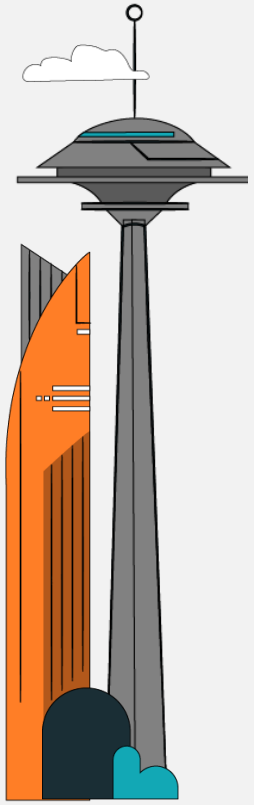
# Phased Tests



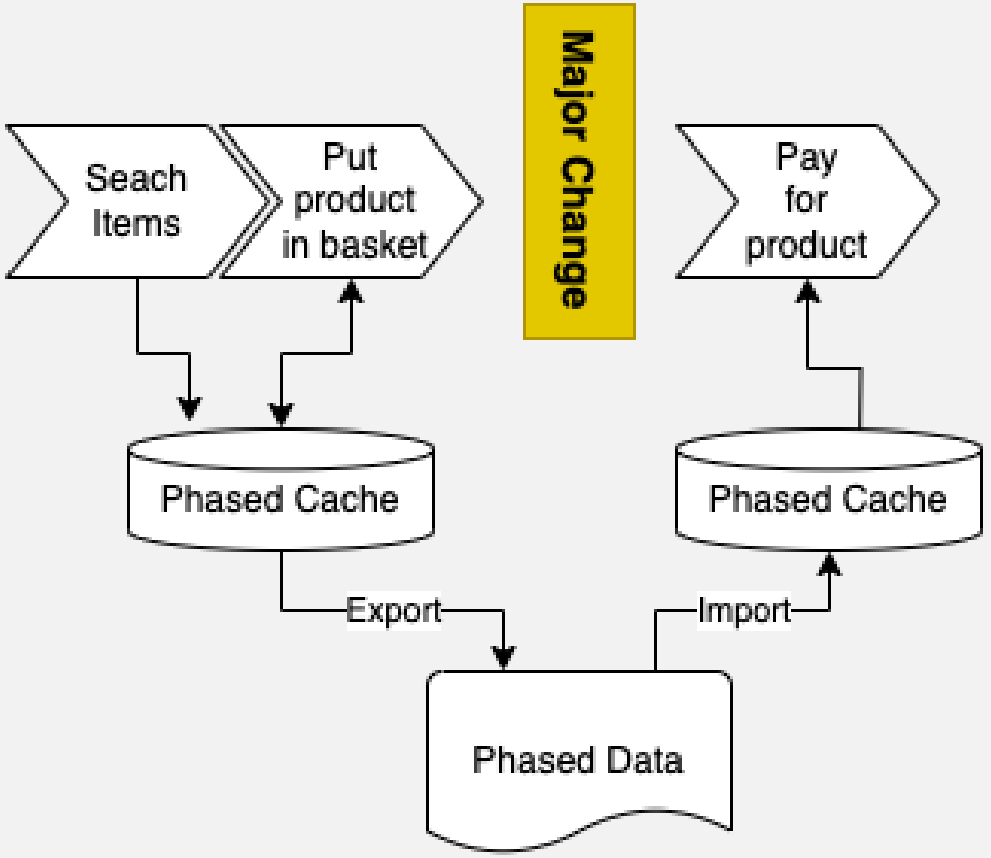
- A Scenario that can be interrupted
- Shuffling
- Managing Context
- A Testing Design

| | PHASE 1 (PRODUCER) | Major Change | PHASE 2 (CONSUMER) |
|---|---|---|---|
| ShuffleGroup_3_0 | Seach Items → Put product in basket → Pay for product | | |
| ShuffleGroup_2_1 | Seach Items → Put product in basket | | Pay for product |
| ShuffleGroup_1_2 | Seach Items | | Put product in basket → Pay for product |
| ShuffleGroup_0_3 | | | Seach Items → Put product in basket → Pay for product |

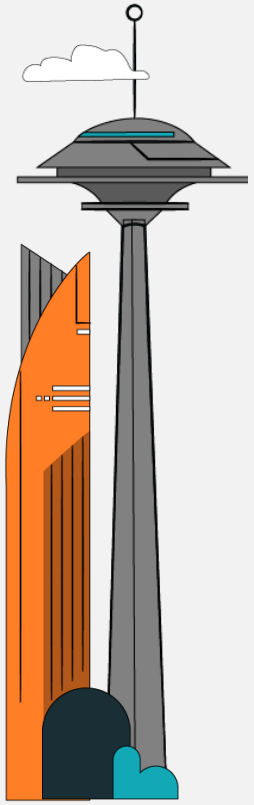
# Phased Tests



- A Scenario that can be interrupted
- Shuffling
- Managing Context
- A Testing Design



# Phased Tests

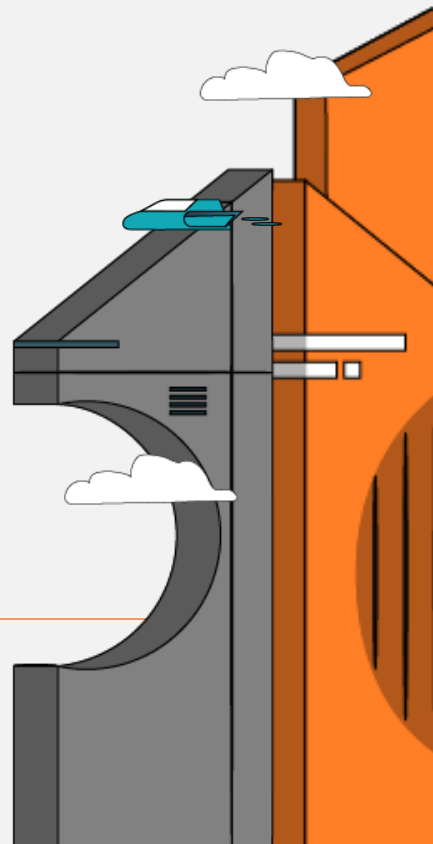


- A Scenario that can be interrupted
- Shuffling
- Managing Context
- A Testing Design

**Clear boundaries and dependencies for steps in practice.**

# We Mutated

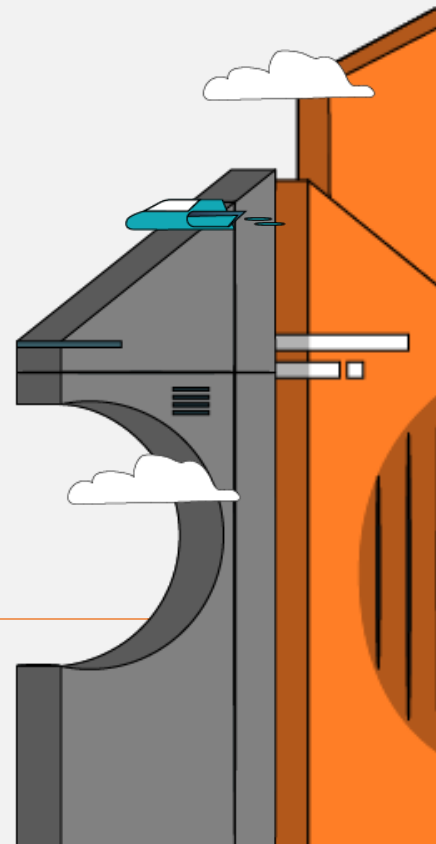
- Lessons learned from the Phased Tests
- Can we apply this beyond upgrades?
- How much further can we go?



# We Mutated

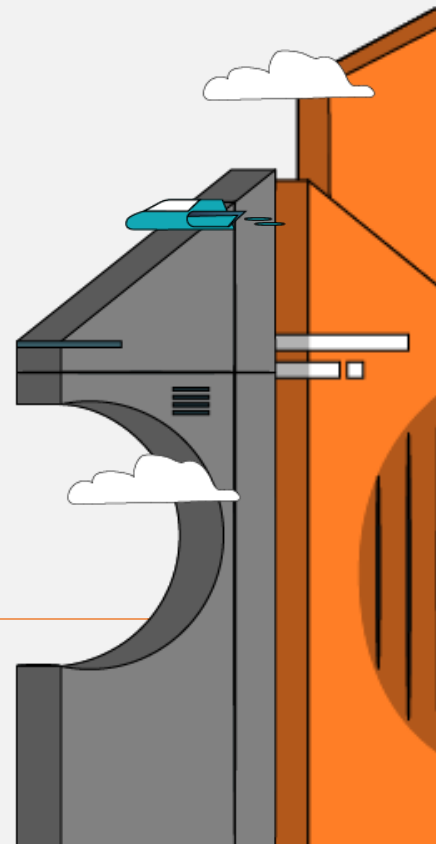
- Lessons learned from the Phased Tests
- Can we apply this beyond upgrades?
- How much further can we go?

MUCH FURTHER!

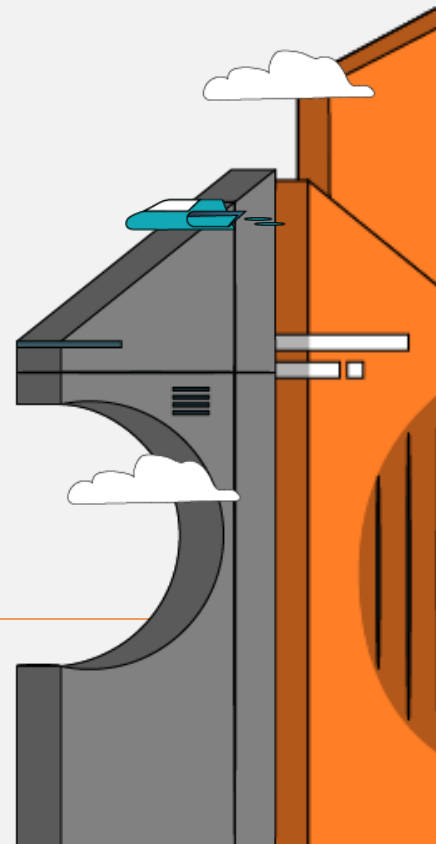


# Mutation Powers

- Data
- Events
- Order

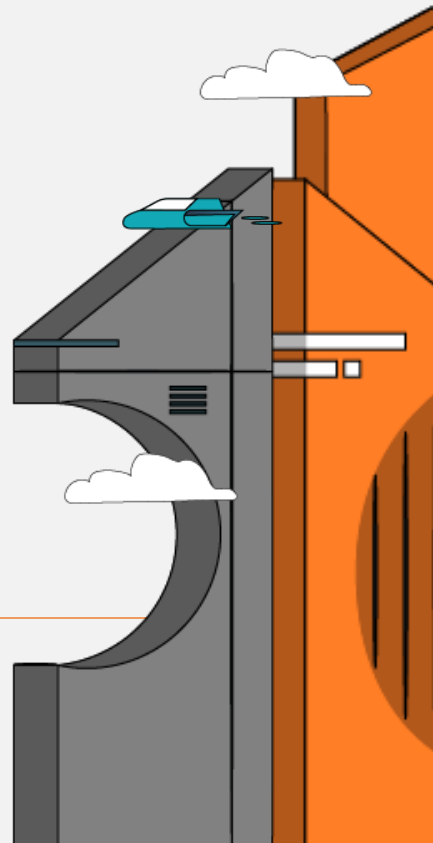


# Data Mutations

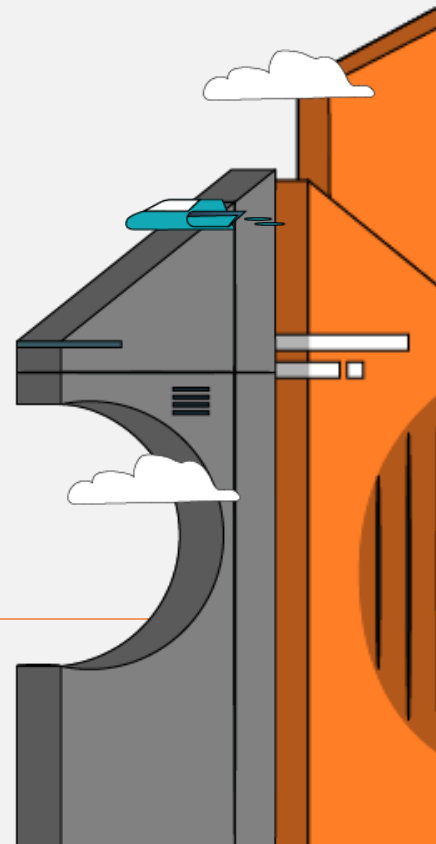


# Data Mutations

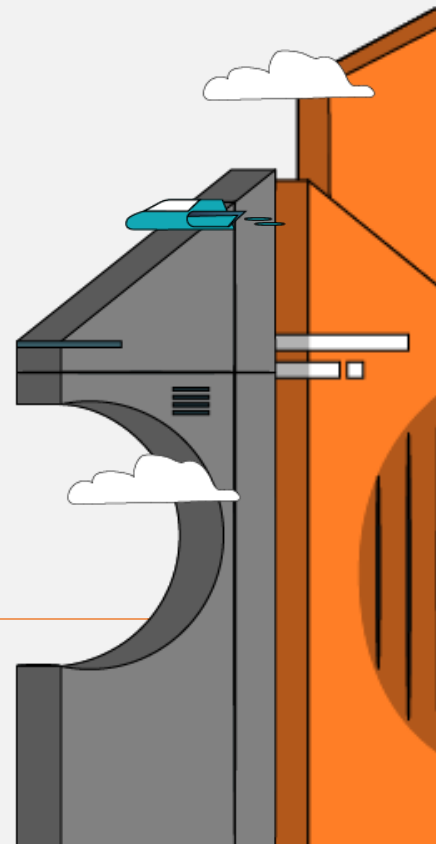
- We are probably doing it already
- Extracting test data



# Demo Preparation

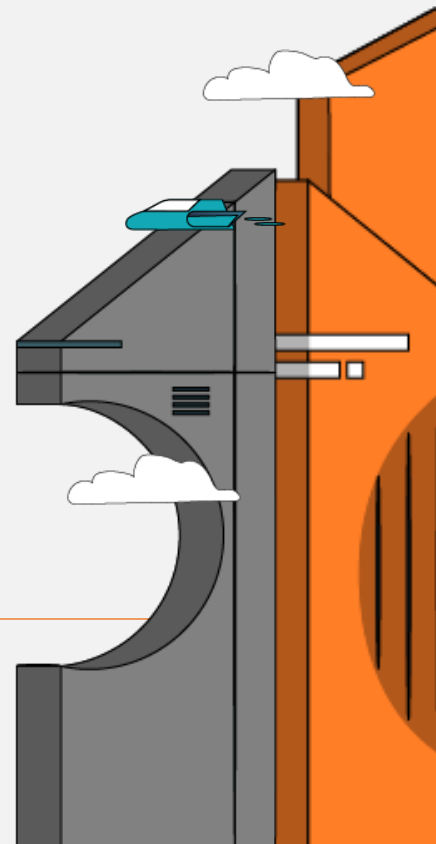


# Event Based Mutations



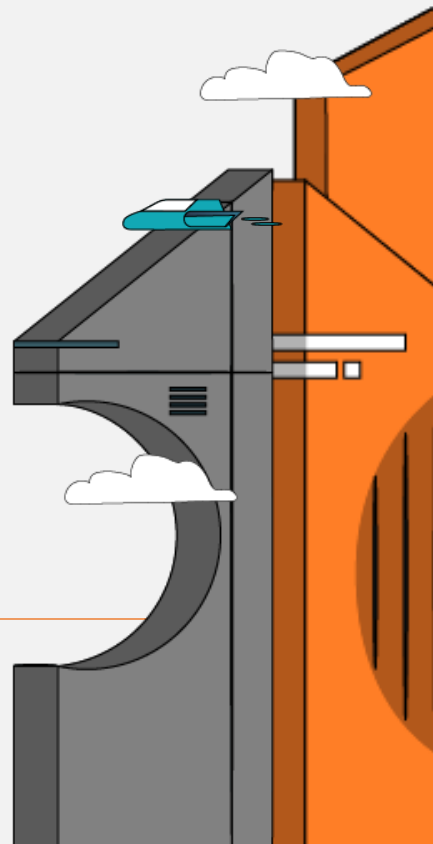
# Event-Based Mutations

- Reacting to external events
  - Planned
    - Upgrades
    - Migrations
  - Un-Planned
    - Network problems
    - System Crashes
    - Restarts
    - Data surges



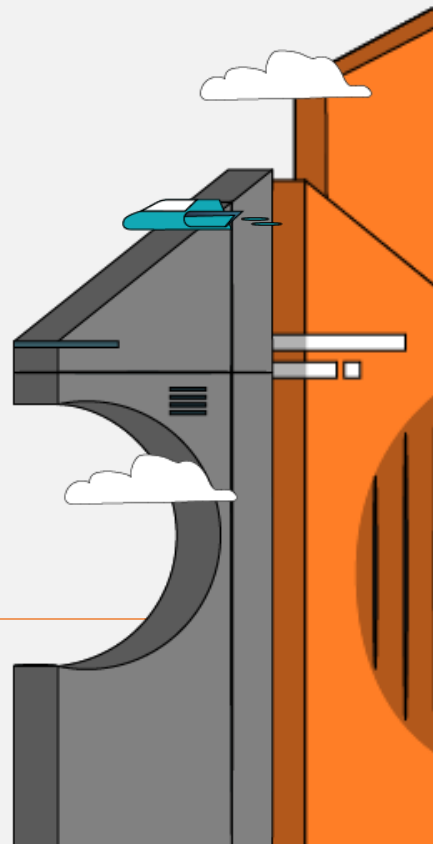
# Event-Based Mutations

- Interruptive
- Non-Interruptive



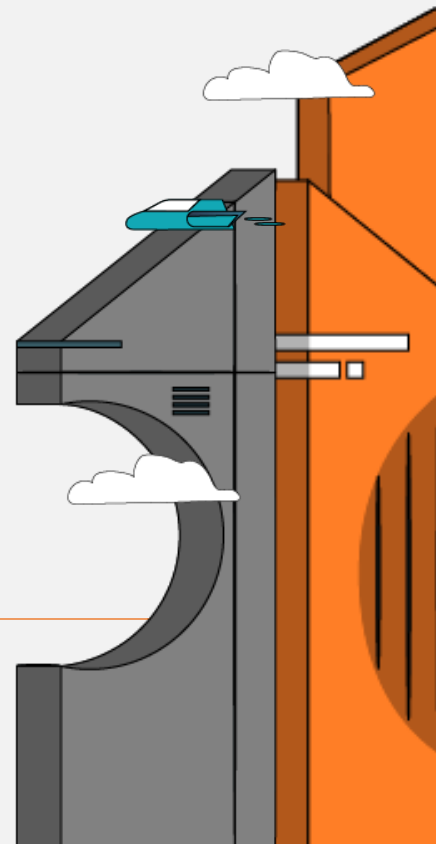
# Interruptive Events

- Stop everything you're doing
- Migrations
- Some upgrades



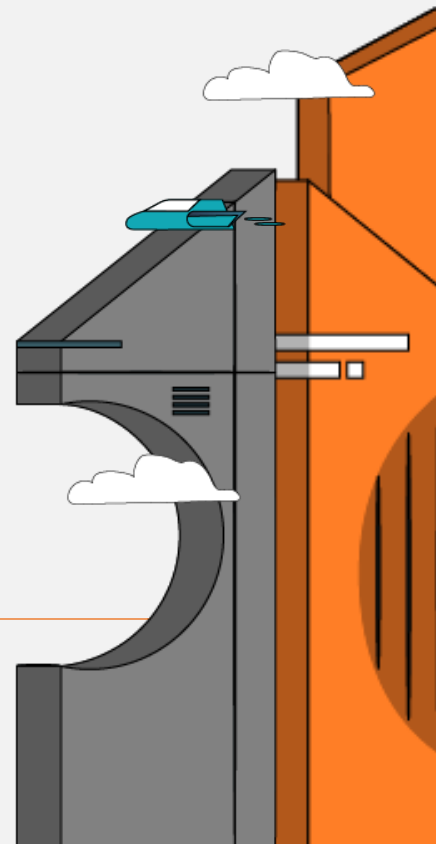
# Interruptive Events: Challenges

- Carry on your scenarios after the interruptions



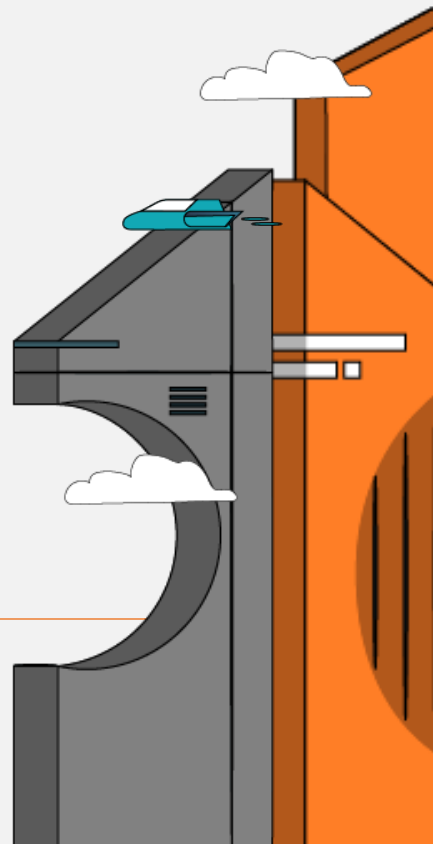
# Shuffling : Execute with all possible Interruptions

| | PHASE 1 (PRODUCER) | Major Change | PHASE 2 (CONSUMER) |
|---|---|---|---|
| **ShuffleGroup_3_0** | Seach Items → Put product in basket → Pay for product | | |
| **ShuffleGroup_2_1** | Seach Items → Put product in basket | | Pay for product |
| **ShuffleGroup_1_2** | Seach Items | | Put product in basket → Pay for product |
| **ShuffleGroup_0_3** | | | Seach Items → Put product in basket → Pay for product |



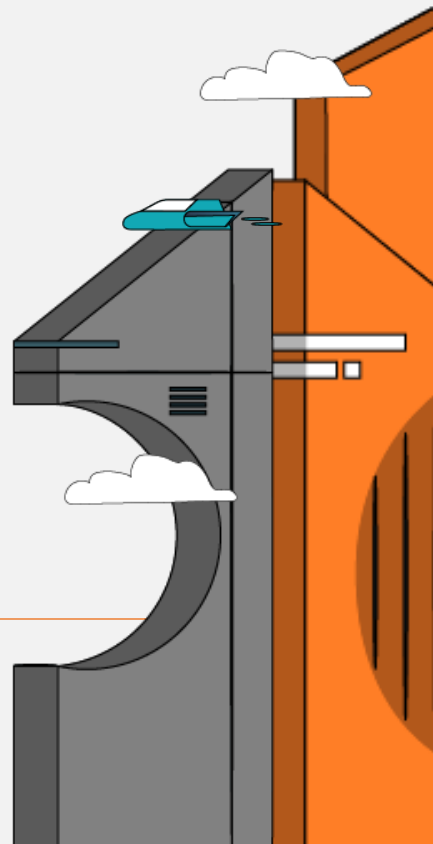
# Non-Interruptive Events

- Often Unexpected
- Chaos Testing
- But not always chaotic



# Non-Interruptive Events: Challenges

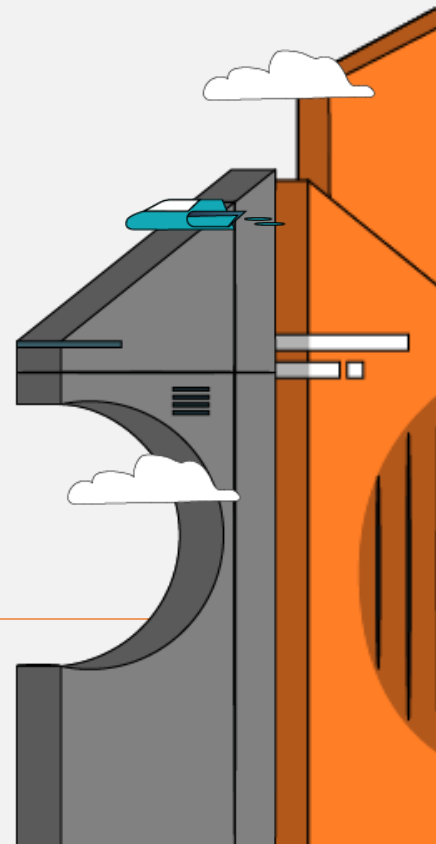
- Wrapping event around the step
- Wrapping the scenario around the event



# Non-Interruptive Events: Shuffling

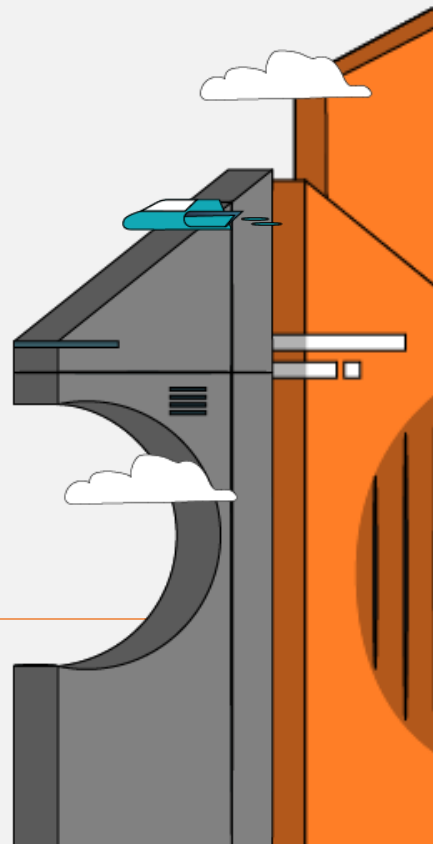
| | Non-Interruptive |
|---|---|
| ShuffleGroupNIE_1 | |
| ShuffleGroupNIE_2 | |
| ShuffleGroupNIE_3 | |

# Order Mutations

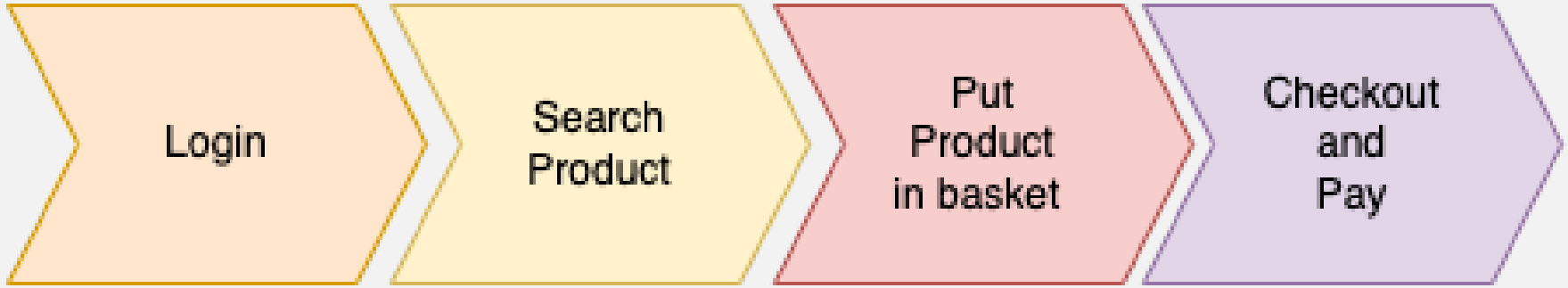


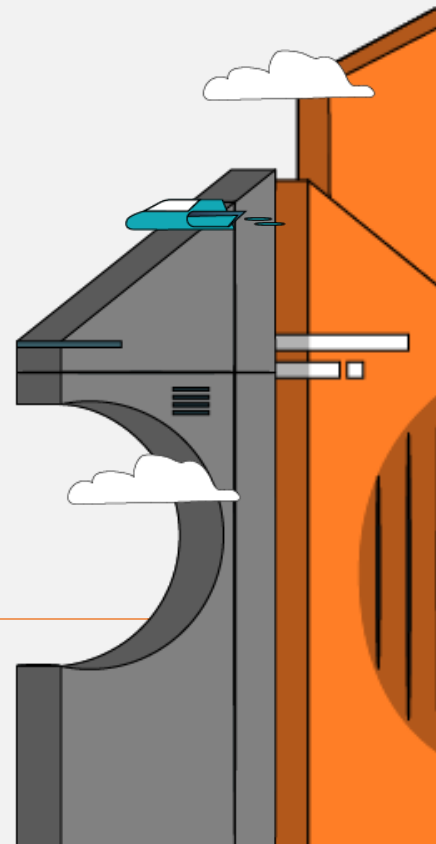
# Order Breaking Mutations : Permutations

- Dependencies between steps
- Logical order(s)

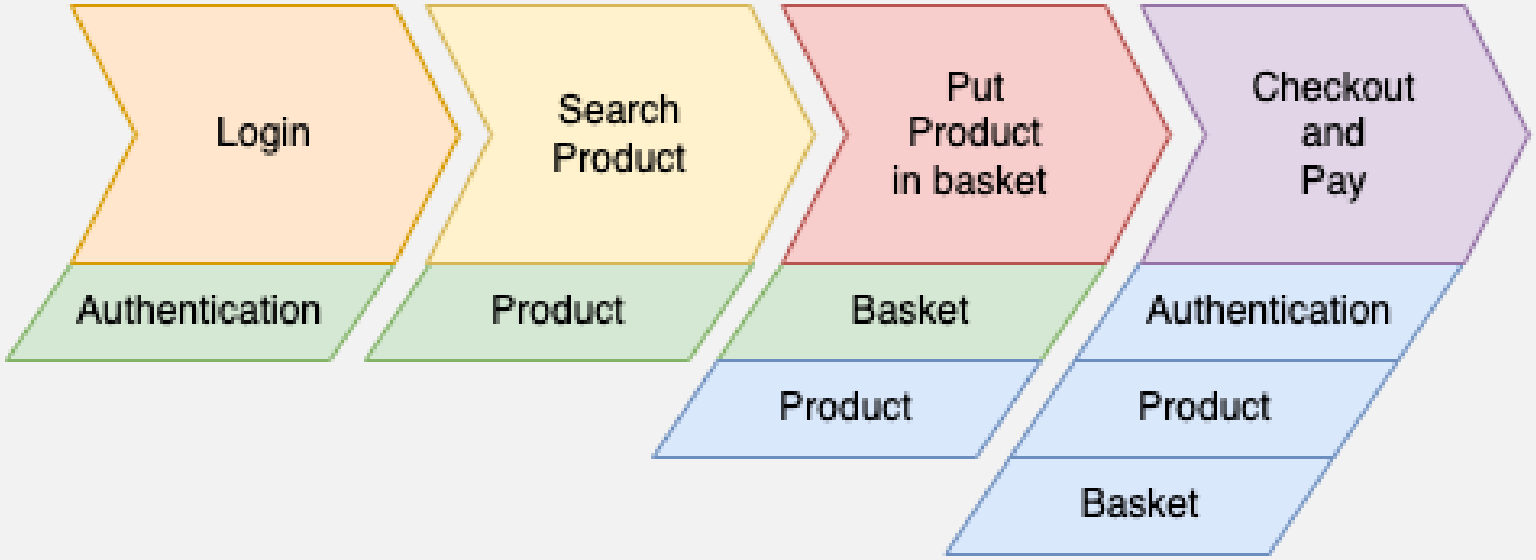


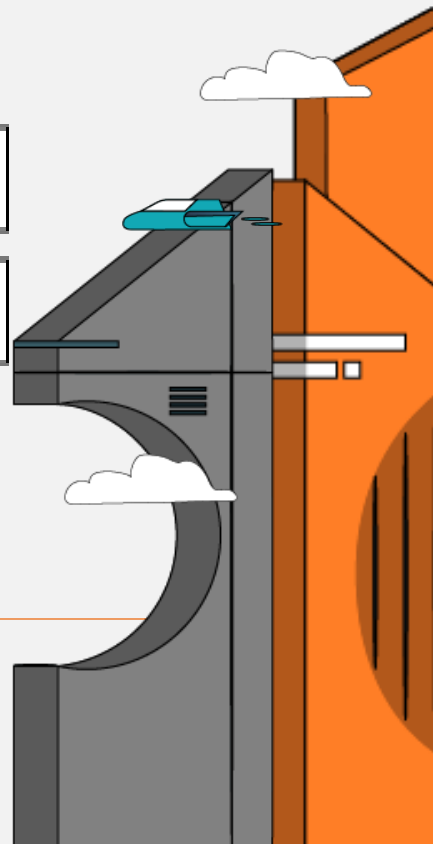
# Order Breaking Mutations : Permutations





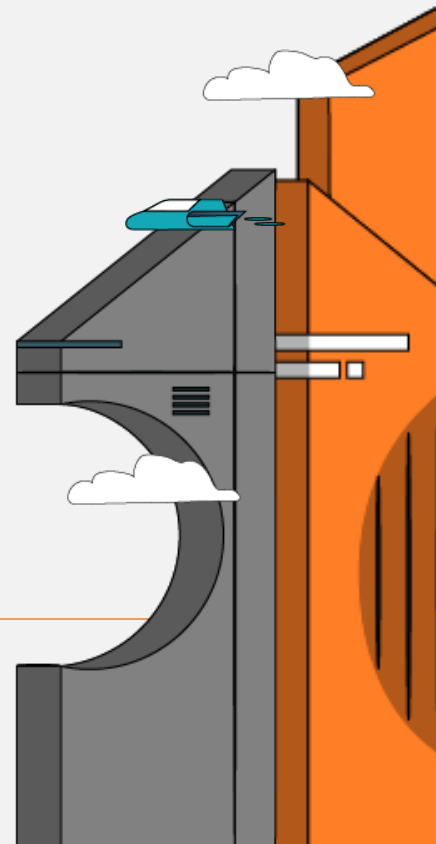
# Order Breaking Mutations : Dependency Map





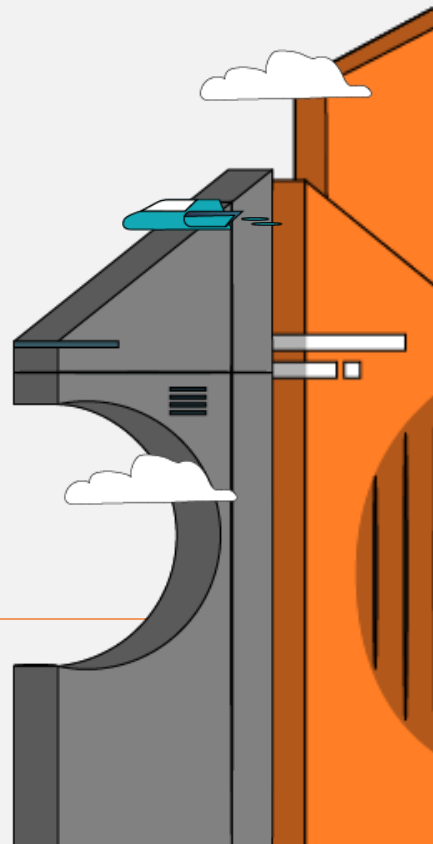
# Order Breaking Mutations : Permutations

| | |
|---|---|
| **ShuffleGroup_permutational_1_3** | Login → Search Product → Put Product in basket → Checkout and Pay |
| **ShuffleGroup_permutational_2_3** | Search Product → Login → Put Product in basket → Checkout and Pay |
| **ShuffleGroup_permutational_3_3** | Search Product → Put Product in basket → Login → Checkout and Pay |



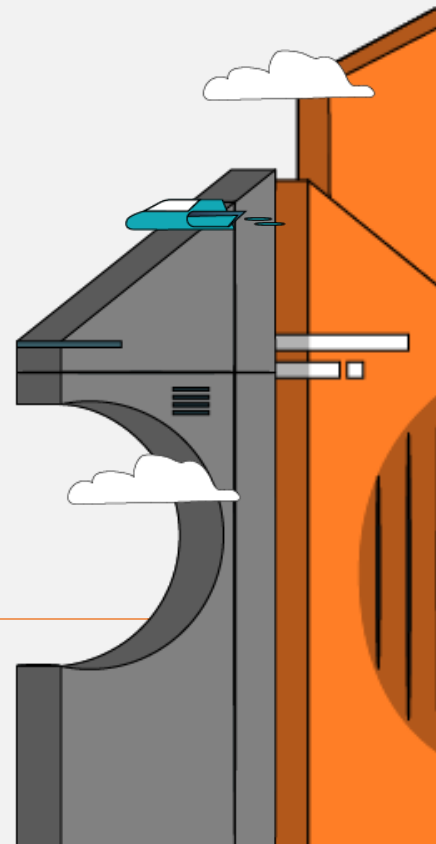
# Conclusions - Takeaways

- A scenario can and should be used in many contexts
- Start asking your product questions



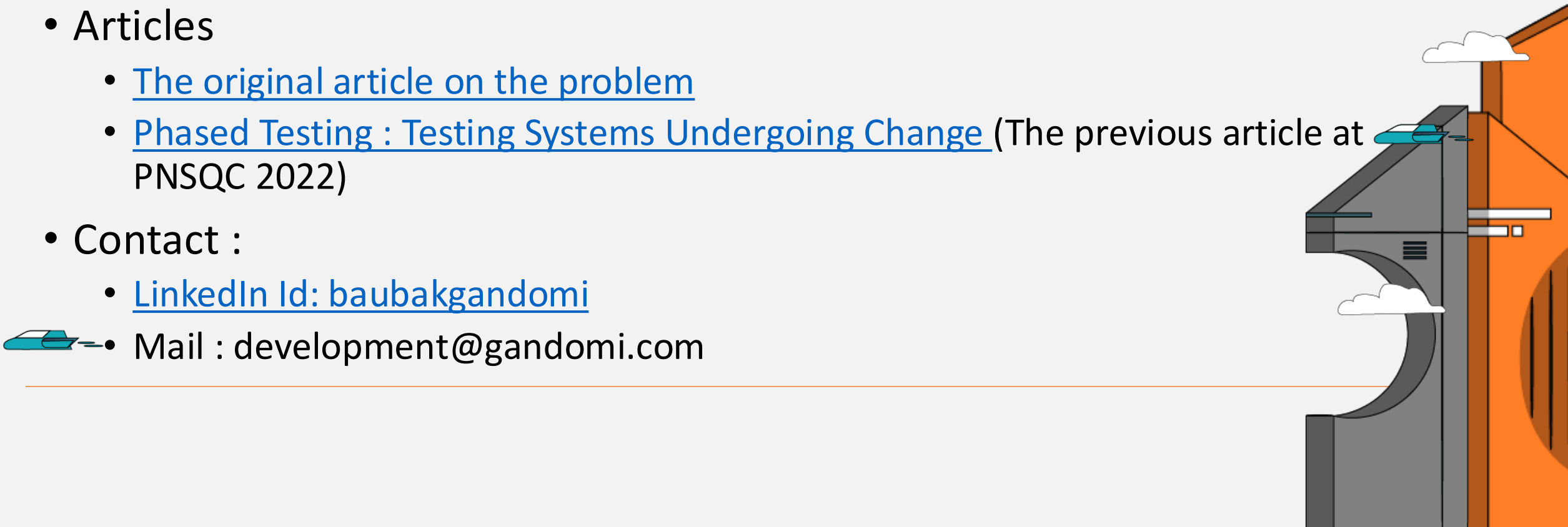
# Conclusions - Evolution

- Bye bye Phased Tests – Hello Mutational
- Disappearance of a static notion of the scenario
- Moving towards intelligent steps
- Work to be done on mapping and wrapping events
- Gaining in maturity
- A Cucumber implementation



# References

- The Mutational Test Branch – at https://github.com/adobe/phased-testing
- The Demo
- Articles
  - The original article on the problem
  - Phased Testing : Testing Systems Undergoing Change (The previous article at PNSQC 2022)
- Contact :
  - LinkedIn Id: baubakgandomi
  - Mail : development@gandomi.com



# THANK YOU

PACIFIC NW SOFTWARE QUALITY CONFERENCE

## THE FUTURE IS NOW

PNSQC.ORG

OCTOBER 14-16 2024