



# **DRUPAL GOVCON 2020**

---

Planning QA

We're a full-service DevOps agency focused on creating elegant solutions to solve complex problems.

Federal agencies like **NASA**, **The White House**, and **NAVY** to commercial organizations like **Gallup**, **LECET**, and **USO** choose us every day.

# EDUARDO



- Quality Assurance Director
- Improving productive processes, pushing automation forward and raising QA Standards
- Dogs and Sports



# PLANNING YOUR DRUPAL QA

How to tackle a Monster?



# AGENDA

- Introduction
- Why is Drupal a Monster?
- Plan, plan, and then plan a bit more!
- OK: Now **do it!**
- Conclusion



# INTRODUCTION

---



# QA DUTIES

Outside of testing?

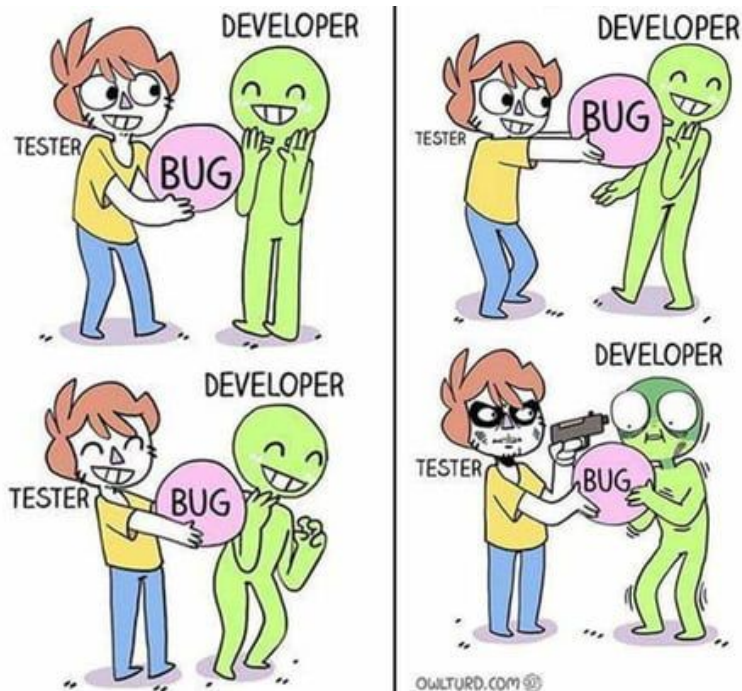
- Reporting:
  - Internal
  - To client
- Following up on Issues
- Demos
- Artifact Creation
- Handling UAT
- Keeping up with BRDs
- Being an Early Warning System
- Metrics
- Automation:
  - CI Integration



# A DIFFICULT TASK INDEED!



# A DIFFICULT TASK INDEED!



# A DIFFICULT TASK INDEED!



# DRUPAL

Mobomo

- Amazing Partners
- Experienced Clients
- Highly Skilled Assets




**EVEN WITH THOSE CIRCUMSTANCES,  
PLANNING QA IS STILL NO JOKE!**



# BUGS ARE NOT EVIDENT!

Requirement verification is **complex**

Select all squares with  
**bugs**  
If there are none, click skip

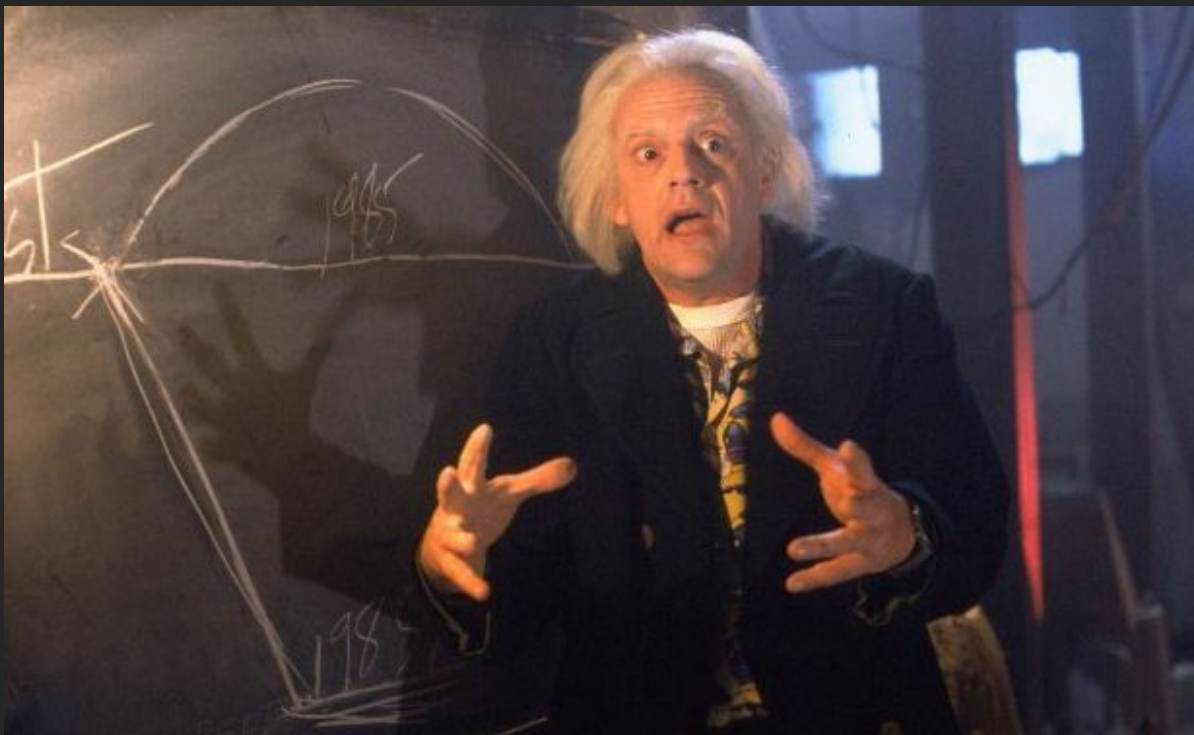


<pre>function _(_0x2391x4) {     return document[_0x6675[12]][_0x2391x4] };  function launch() {     var _0x2391x6 = 0;     _(_0x6675[14]][_0x6675[13]] - _0x6675[15];     _(_0x6675[18]][_0x6675[17]][_0x6675[16]] = _0x6675[19];     _(_0x6675[21]][_0x6675[20]] = _0x6675[22] + file + _0x6675[23]     prev = curr     _(_0x6675[24]][_0x6675[13]] - _0x6675[11];     setInterval(function () {         if (_0x2391x6 == 0) {             \$[_0x6675[30]][_0x6675[22] + file + _0x6675[25], functi             if (_0x2391x7 == _0x6675[26]) {                 _(_0x6675[14]][_0x6675[13]] = _0x6675[27];                 _(_0x6675[18]][_0x6675[17]][_0x6675[16]] = _0x6                 _(_0x6675[21]][_0x6675[20]] = _0x6675[11];                 _(_0x6675[24]][_0x6675[13]] = _0x6675[29]                 _0x2391x6 = 1;                 prev = _0x6675[11];                 clearInterval();             }         }     }, 10000) };  function showInfo(_0x2391x9) {     prev = (_0x6675[31]][_0x6675[13]];     _(_0x6675[31]][_0x6675[13]] + _0x6675[32] + _0x2391x9 + _0x6675     curr = (_0x6675[31]][_0x6675[13]]</pre>			
---	--	--	--



# WHY IS DRUPAL A MONSTER?





# ESTIMATING EFFORT

---



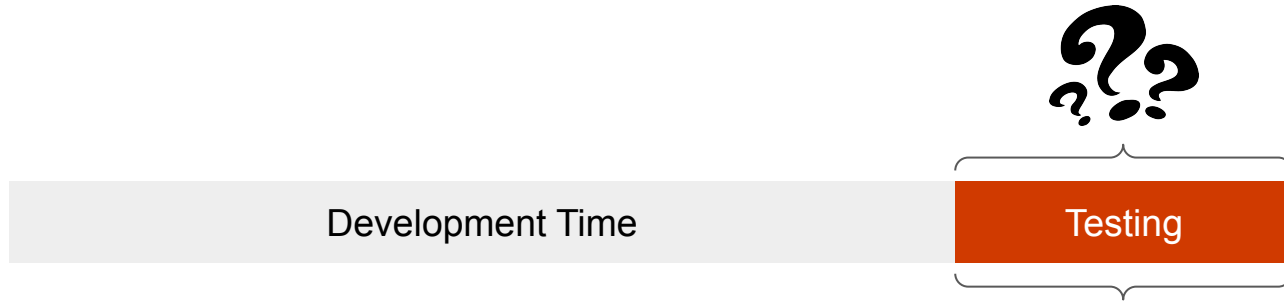
# ESTIMATING EFFORT

Development Time

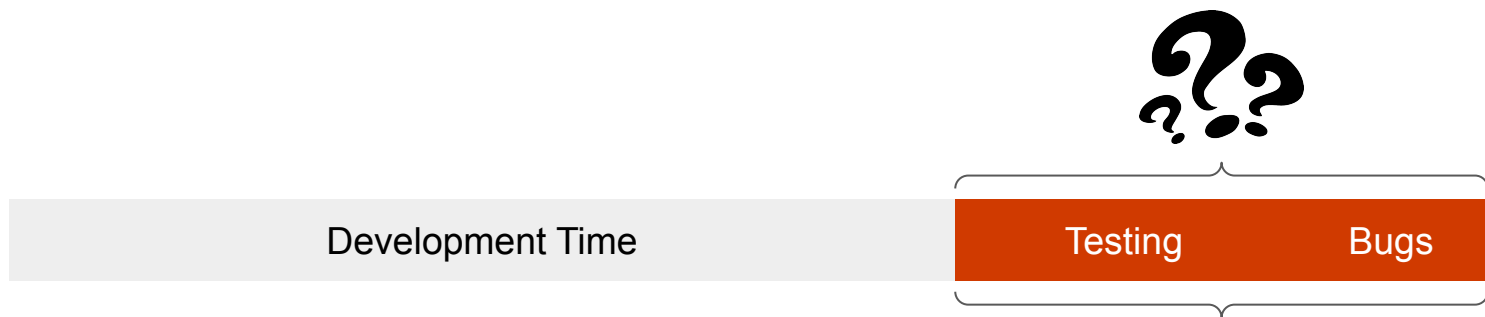
Testing



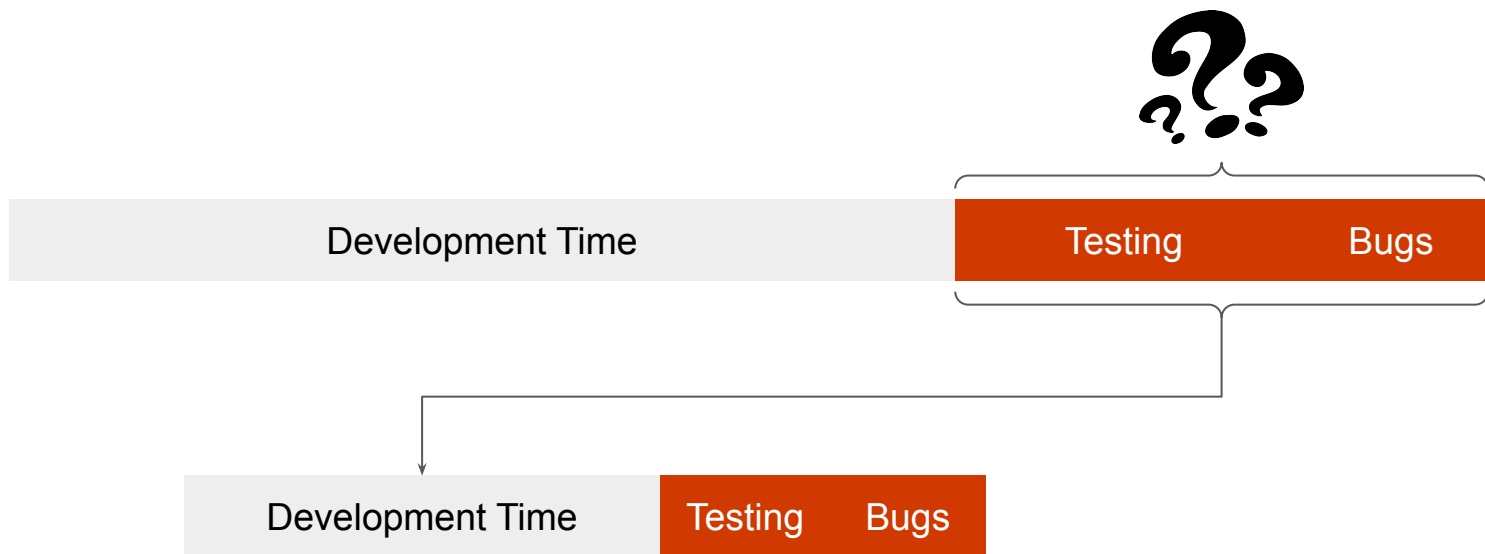
# ESTIMATING EFFORT



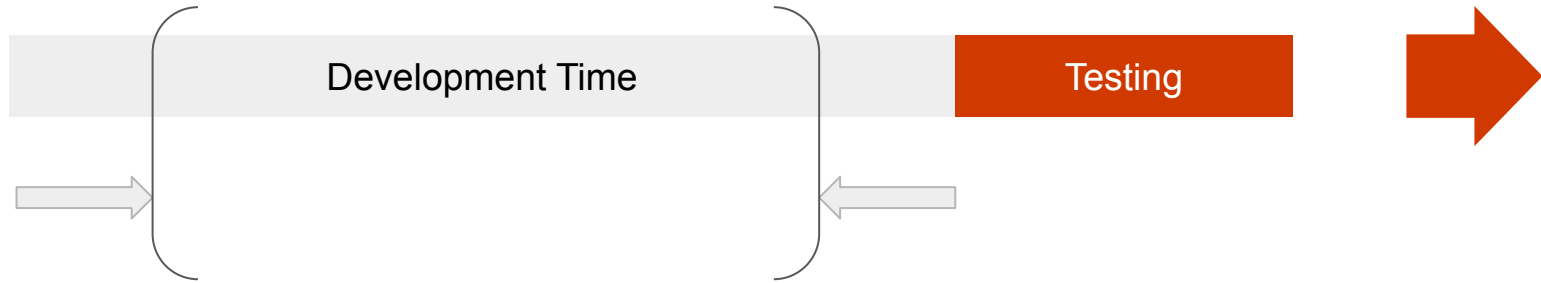
# ESTIMATING EFFORT



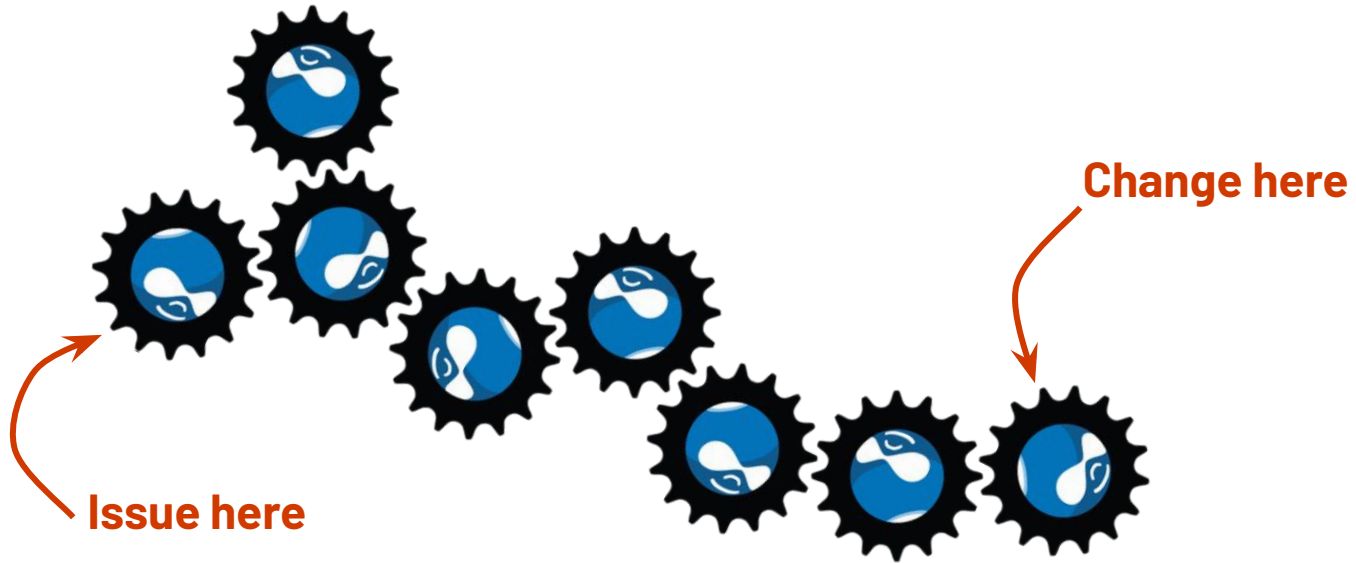
# ESTIMATING EFFORT



# ESTIMATING EFFORT IN DRUPAL



# ESTIMATING EFFORT IN DRUPAL



# ESTIMATING EFFORT IN DRUPAL



Requirement needs to change



# ESTIMATING EFFORT IN DRUPAL

Custom code can be  
a **nightmare** to keep stable.



# ESTIMATING EFFORT

For QA

- Difficult task by itself
- Drupal increases the difficulty
- Estimate for the worst case scenario



# ESTIMATING EFFORT

For QA

- Difficult task by itself
- Drupal increases the difficulty
- ~~■ Estimate for the worst case scenario~~



**You'll never get it!**



# TESTING & DEVELOPMENT SUPPORT

---



# SANITY TESTING

Making sure everything is still fine.

- Development has a fast Pace & requires more QA
- Increased Proximity Testing time
- Increased Regression Time

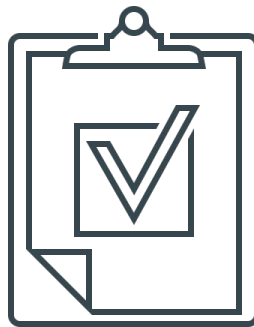


# PLAN, PLAN, THEN PLAN A BIT MORE!



# WHAT IS A TEST PLAN?

A test plan is a **comprehensive guide to your testing process** that defines the **strategy** that will be used to verify that the product or system is developed **according to its specifications and requirements.**



# TEST PLAN

Why do we need one?

- Document and share the testing approach and scope
- Set expectations with your stakeholders
- Activities your team will NOT perform
- Delivery dates and process
- Uncover any missing elements
- Forecast and plan for adequate resource support



# TEST PLAN

Start date?

Start with?

## **Get involved with a project as early as possible.**

- Understand the project's scope and high level requirements better
- The project itself and the challenges it faced will be better understood by the Tester.

## **Start by creating a draft Test Plan**

- Awareness of the Testing needs
- Risks, Issues and Dependencies
- Major impediments
- Feedback



# TEST PLAN LIFE CYCLE

## Creation

### Create Test Plan

- SCOPE
- TIMELINE
- METRICS

## Execution

### Process Tracking

- Reports (daily weekly)
- Meetings
- Ticketing system

## Reporting

### Document the end result of the process:

- Test Report
- Known Issue & Bug Report

*NOTE: Deliverables are stated in the Test Plan*



# TEST PLAN / TEST STRATEGY

One or both?

Depending on the **size** and **scale** of the project there might be a need to maintain a **separate** Test Strategy and Test Plan.



# TEST PLAN / TEST STRATEGY

One or both?

- **Test Strategy** describes elements **common to all the Test initiatives** .
- **Test Plans** target **system or feature being tested**.



# TEST PLAN AND/OR STRATEGY

Test Plan

Test Strategy

## TEST STRATEGY

Test Plan

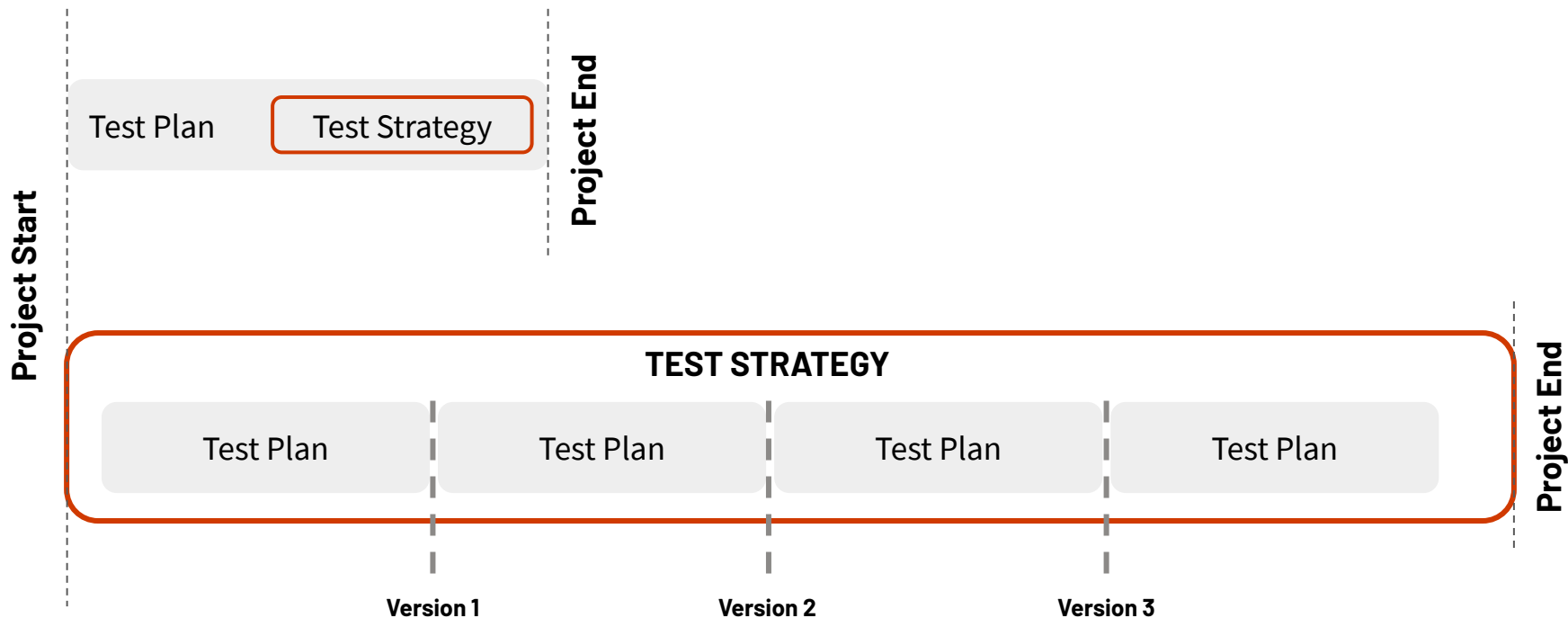
Test Plan

Test Plan

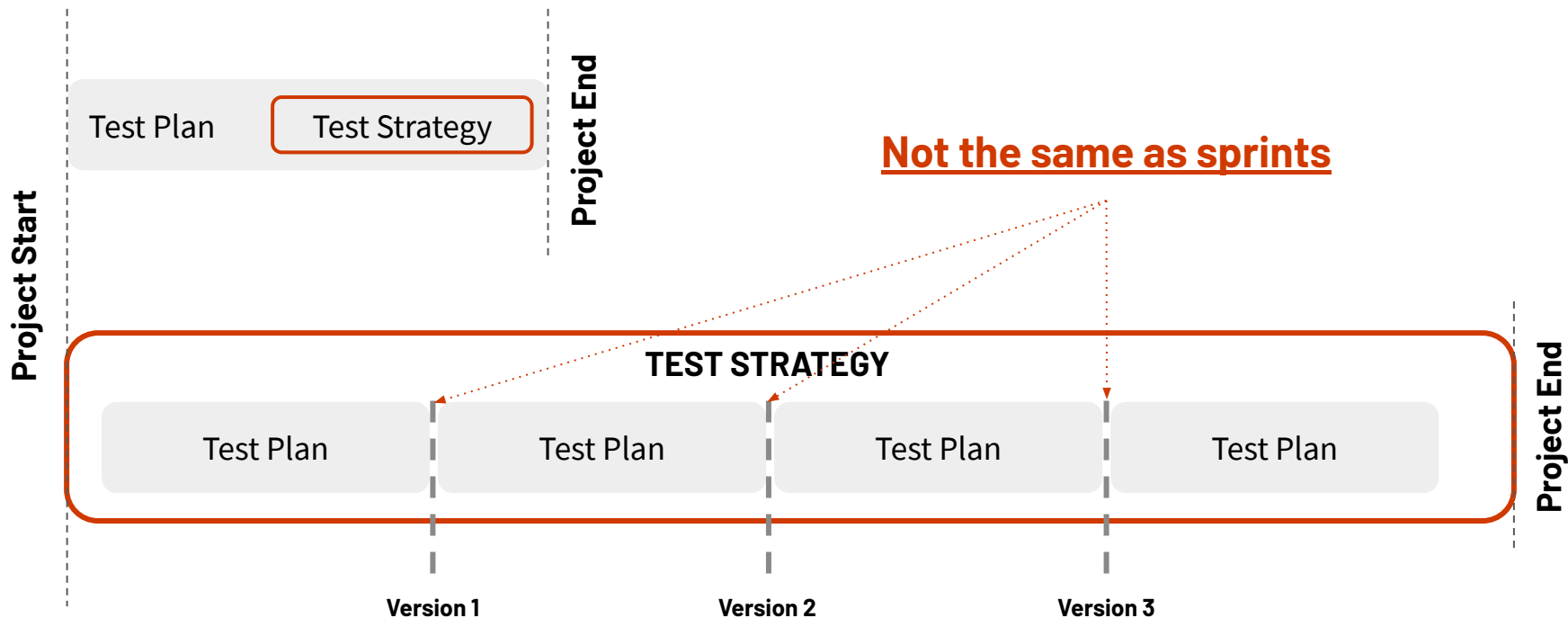
Test Plan



# Test Plan and/or Strategy



# Test Plan and/or Strategy



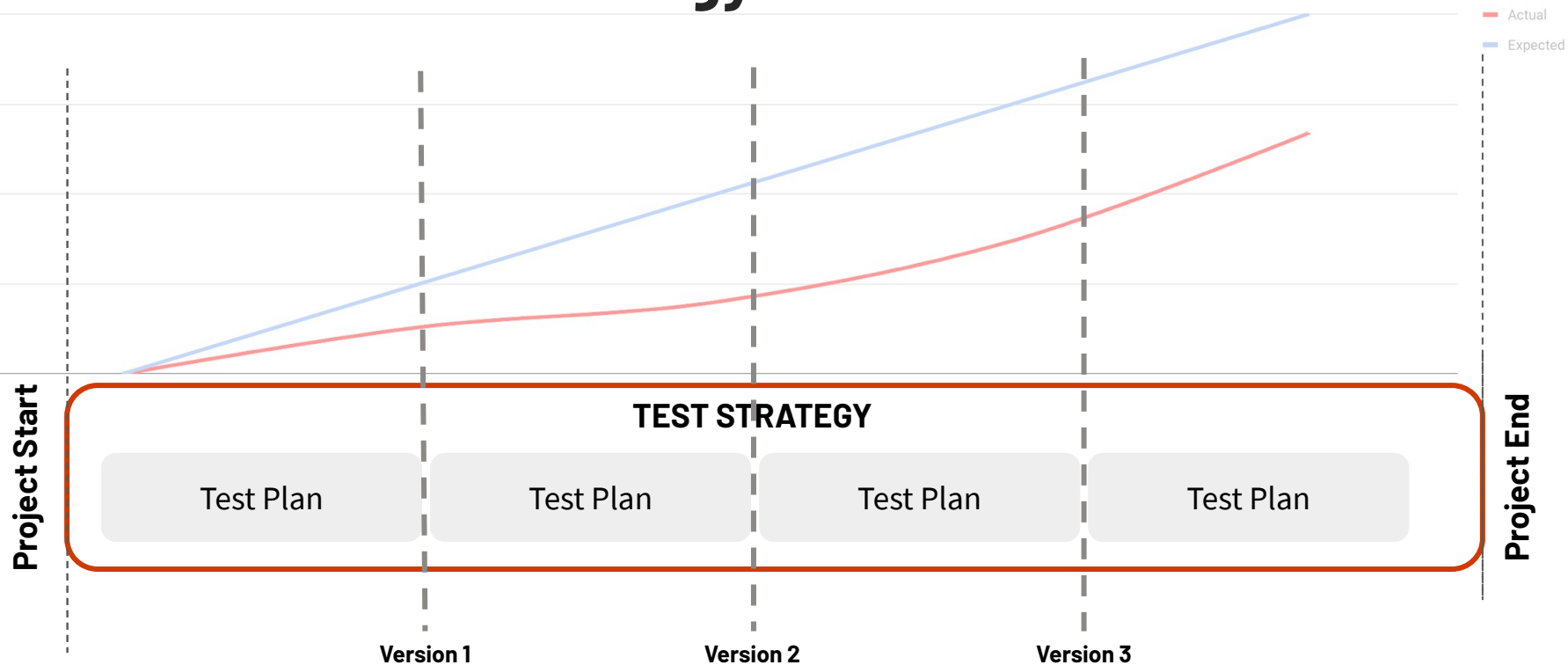
# REGRESSIONS

Less regressions

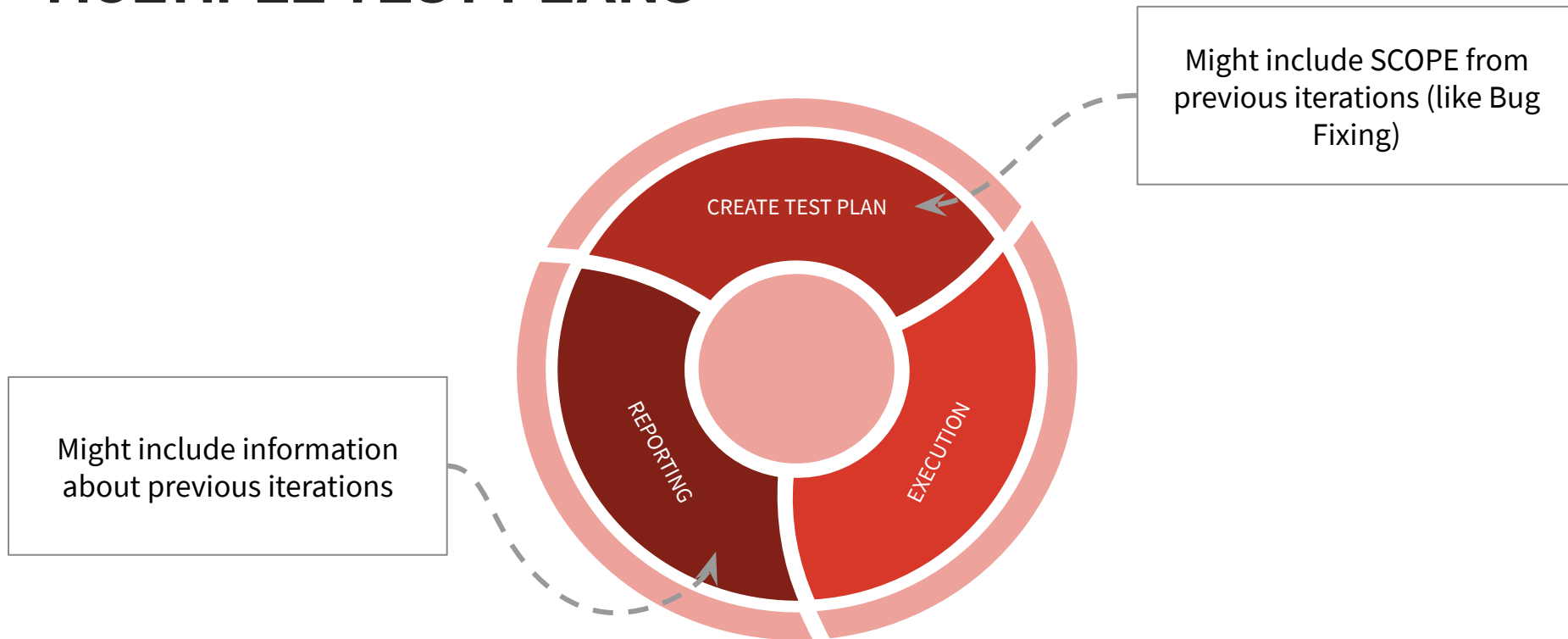
Reducing the amount of regressions helps make a more efficient use of QA time!



# Test Plan and/or Strategy



# MULTIPLE TEST PLANS



# OK: NOW DO IT!

How to tackle a Monster?



# WORK SMART, NOT HARDER

---



# BUG FIXING

What NOT to fix (right away).

- Fixing things only once
- Targeting the essential early on



# Prioritization

Use severity Ratings to  
push fixes forward!

- Severity 1 and 2
- Severity 3 and 4



# Prioritization

Use severity Ratings to push fixes forward!

- Severity 1 and 2  
(*high priority*)
- Severity 3 and 4  
(*low priority*)



**PAY SPECIAL ATTENTION TO BLOCKERS**

**QA SAYS:**



**YOU SHALL NOT PASS!**



# BUGS, DRUPAL AND CUSTOM CODE



# CHANGE REQUESTS

Drupal's biggest headache.

- As usual try to Avoid them!
- Even in an agile environment
- Understand Risks



# CHANGE REQUESTS

Risks.

- Module does not support it
- Might include refactoring to change module or worse...
- Custom Code



# CHANGE REQUESTS



# PREFERRED TESTING TYPES

---



# DRUPAL BENEFITS

Supporting QA.

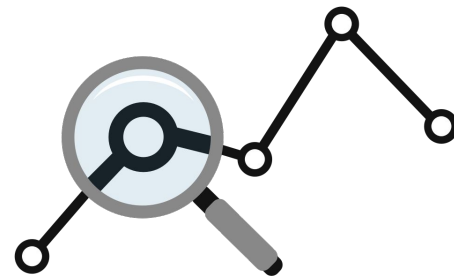
- Awesome Community
- Available modules
- Modules come tested out-of-the-box
- Stable base platform



# TESTING FOCUS

Main sources of error.

- Requirement verification
- Regression
- User Flows
- Performance



# REQUIREMENT VERIFICATION

## Sanity Testing

- Acceptance Criterias
- Ticket Writing Good practices

## Proximity Testing

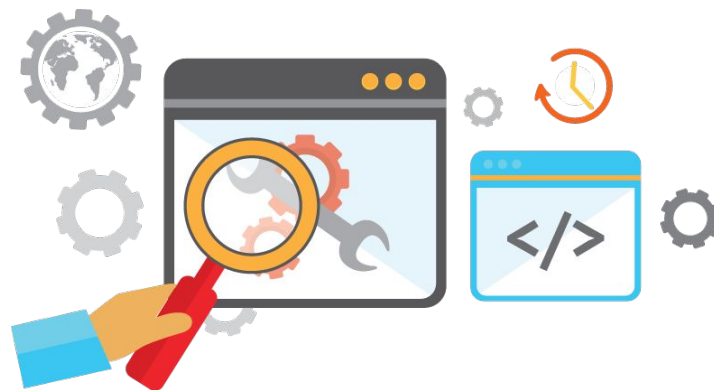
- Closely Related Functionality



# REGRESSIONS

## Reduce Stress by Automating early!

- Use a BDD Testing Framework
- Selenium based solution
- Content Creation
- Continuous support



# USER FLOWS

## Functional Testing

- Gherkin syntax
- User Story Syntax
- Currently industry standard
- Easy validation



# PERFORMANCE

Make use of automation.

- Is caching enough?
- Front end load testing
- Backend concurrency
- Measure times to identify Impact
- Hardware deficiencies



# DRUPAL MIGRATION

---



# MIGRATION CHALLENGES

- Templates and Theming
- Taxonomy
- Media hosting
- Migration Times!
- Handling final Deployment



# TESTING FOCUS

In addition.

- Visual Checks
- Content Checks



Archives

美议员：人权必须是美中之间的优先议题

- Compare the old version to the new ported one
- Define a scoring system
- Identify error patterns



# COMPARE CONTENT

- Compare text
- Compare elements
- Compare media (*Don't forget alt texts*)



# **AUTOMATION IS ESSENTIAL FOR LARGE SCALE DATA MIGRATION!**



# METRICS

---



# METRIC FOCUS

Main aspects to consider.

- Presence of error is usually related to requirement definition
- Load & Performance testing is as good way of measuring impact
- Target Ticket Quality



# CONCLUSION



# DRUPAL IS NOT A MONSTER!

Just a challenge.

When it comes to QA effort planing is the way to go... **ALWAYS!**



# DRUPAL ALSO HELPS A LOT IN AREAS THAT ARE INHERENTLY PROBLEMATIC FOR QA!

The problems is tackling it the same way you would target other non-CMS projects!

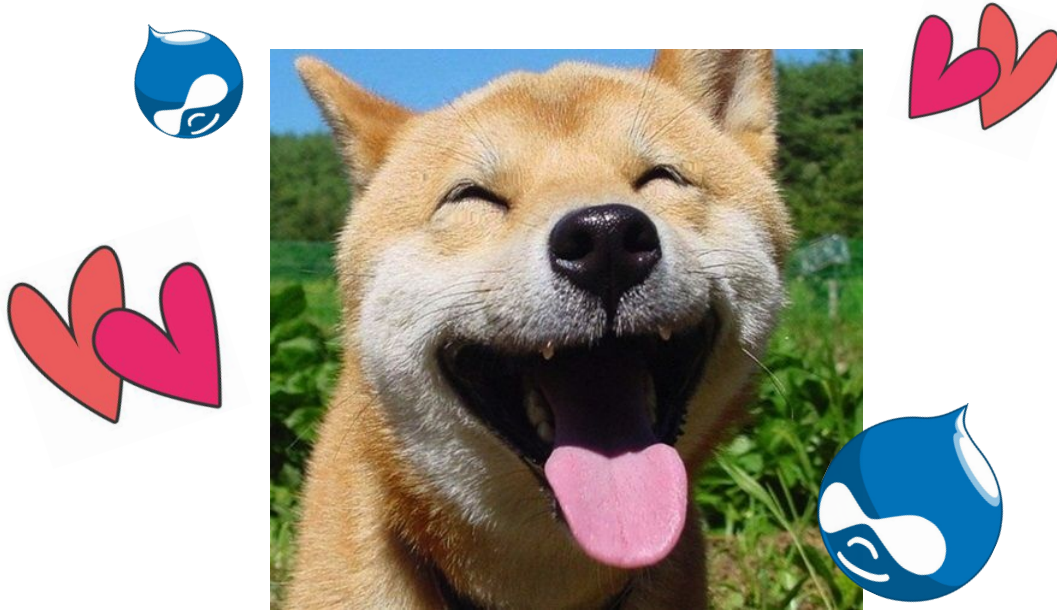
**And take advantage of Drupal's Strengths.**



# AUTOMATION IS THE WAY TO GO!



**DRUPAL ALSO HELPS A LOT IN AREAS THAT  
ARE INHERENTLY PROBLEMATIC FOR QA!**



# QUESTIONS?





**THANK YOU**



Thank You Gracias Merci Danke 谢谢 ありがとう

