

# Are Your Tests Floating Away?

Build Stability into Your Test Automation



# Introduction

In the world of test automation, there are many challenges and hurdles to overcome. Your focus should be on shipping quality software and services to your customers, both internal and external, while meeting the team and department's needs. Thus, creating concrete test cases that run with confidence within your CI/CD pipeline.

But this isn't always a smooth launch...nothing strikes more fear than the **"RED" build pipeline** staring you in the face and having to sift through build logs, test logs, and application logs, stepping through test and dev code only to find out that there was a problem with the test data right from the start.

At AIM Consulting, we have perfected a foolproof system to recognize potential pitfalls and bring best practices to our clients to create **OUT OF THIS WORLD** test automation stability every time.

## What you can expect?

**What is Test Automation?**

---

**Benefits of Test Automation**

---

**Test Case Design & Data Management**

---

**Effects of Bad Test Design**

---

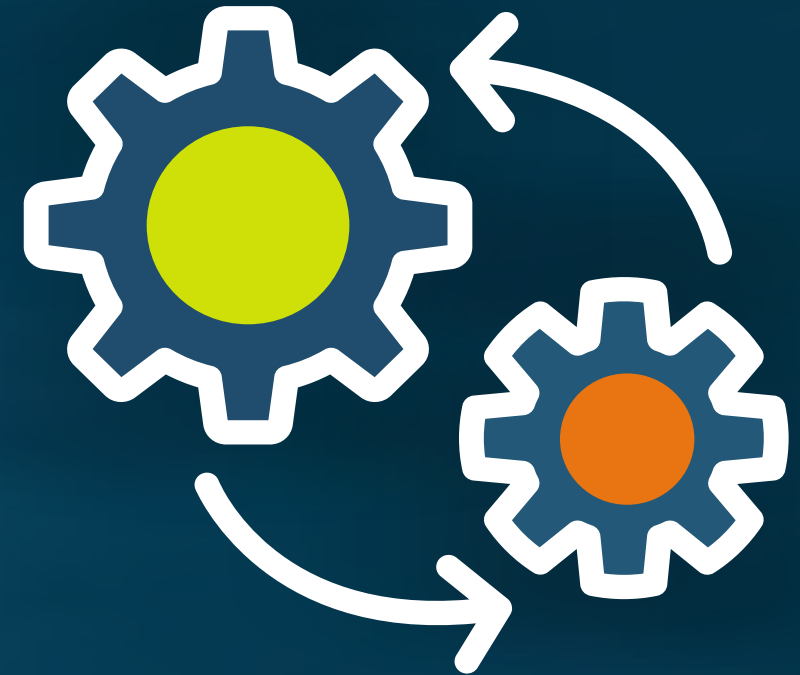
**Avoiding Pitfalls**

---

**Bringing it Back to Earth**

# Let's Start From the Beginning...

What is Test Automation?



Test automation is the practice of running tests automatically, managing test data, and utilizing results to improve software quality.

## **Test automation is more than a quality assurance measure.**

It is a commitment to the entire software production team, ensuring that everyone from business stakeholders to DevOps engineers is getting the most out of the test.

## **Automated test should be...**

**...automatic**

**...exercise the application as a user would**

**...run against software freshly built by developers**

**...report results in a structured, easy to consume manner**

**...depend on connected data and systems**

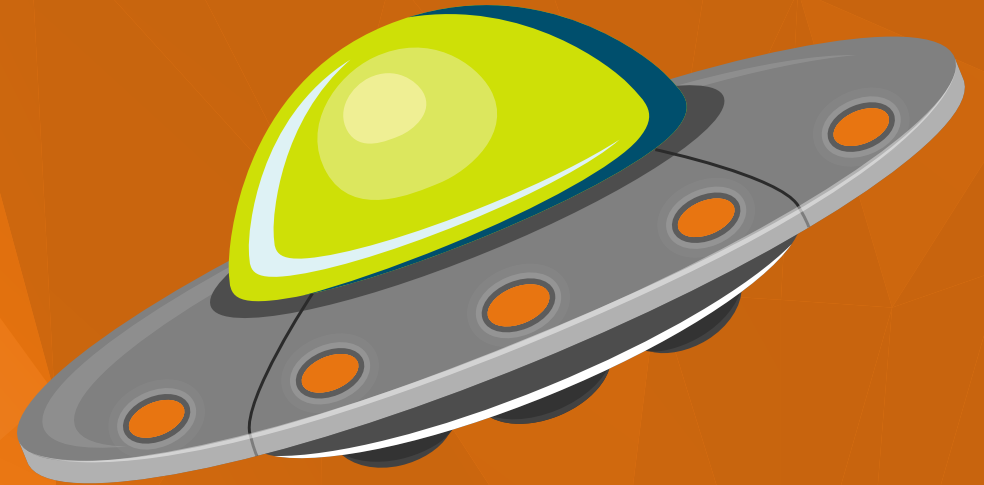


# How Does Test Automation Increase Stability?

- ▶ Tests are most importantly about the truth of the software
- ▶ Stable, predictable tests become a reliable indicator of quality
- ▶ Your team can move with greater confidence when they know that low effort test automation tells how they are doing
- ▶ Tests designed for stability require less maintenance

# How Bad Design Can Steer You Off Course...

Test Case Design & Data Management



# Test Data, Bad Test Data, and it's effects on Your Product Delivery

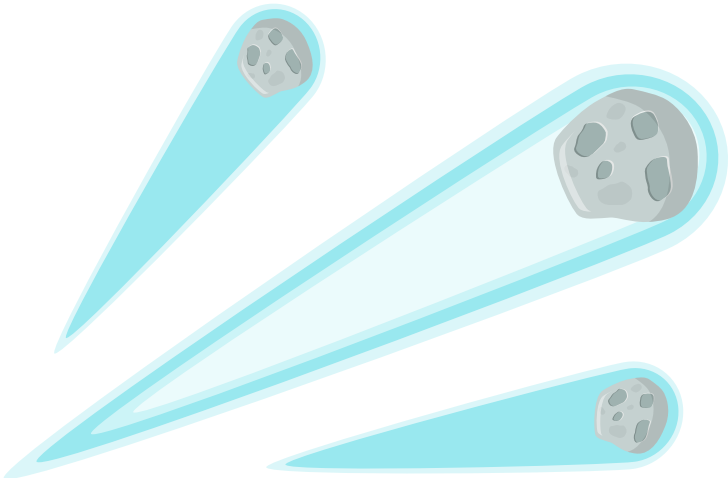
Test case design and acceptable test data management practices **can save you many headaches** when deciphering whether failed tests are indeed an issue with your system or just a false failure due to insufficient test data/test design.

With poor test case design and data mismanagement, your team will end up with wasted time and resources pursuing an issue that shouldn't have occurred from the start.



# Effects of Bad Test Design

Avoid a Meteor Shower of Design Missteps



“

Results of poor test design and poorly managed test case data can impact every stage of the cycle, including the build, sprints, and deliverables.

”

Patrick Taylor

Principal Consultant, Application Development  
AIM Consulting



# The Build: CI/CD & Downstream Customers

One of the first areas that you'll see affected in the world of lousy test case design and test data is your Continuous Integration / Continuous Deployment (CI/CD) pipeline. It's at the forefront of a company's software delivery system, and nothing speaks more to the stability of your builds in CI/CD than a history of build failures.

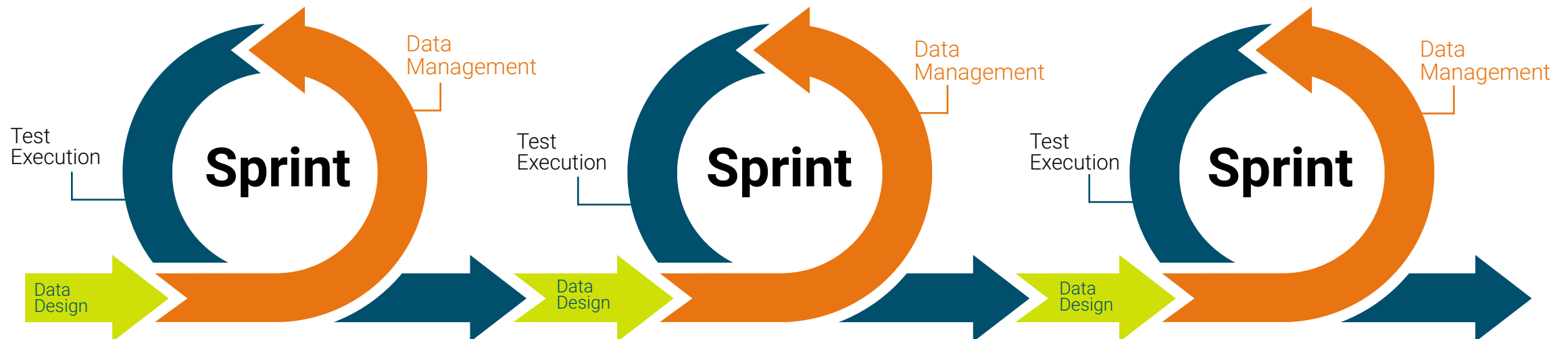
If tests are unreliable, the pipeline failures are always seen as a failure in the test code and not in the system being tested. A failed build should light the developer's hair on fire, not the SDET's. However, if the process is reactive, and only when the pipeline turns red, we correct it by fixing the tests, then developers are never going to look at it.

The usual practice in this scenario is to contact the test owner and start to analyze the problem. If we have issues with bad test data or test design, the test engineer will have to find the test's problem and correct the test to get it to pass.

Even more impactful, test design can affect the downstream users. These may be the end-users of your software and services, or they may very well be internal company customers that rely on your piece of a grander software vision. The delay to those customers can be costly both for the project and those dependent on your resources.

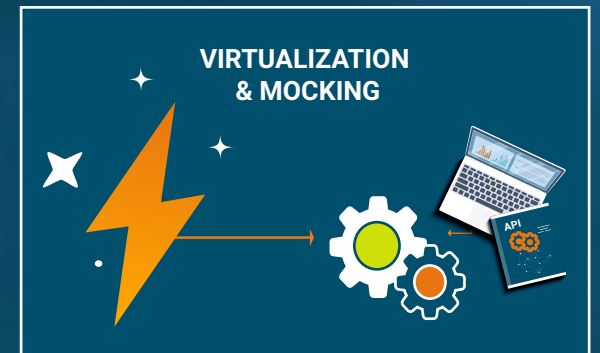
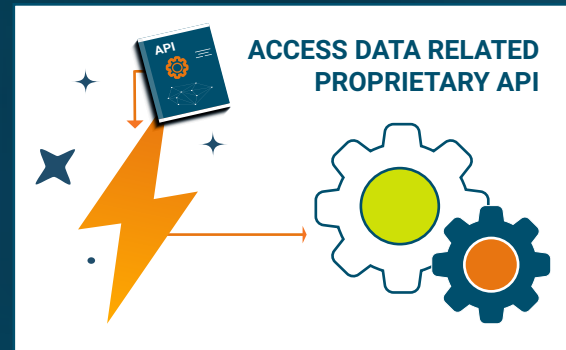
# Sprints & Overall Delivery

Depending on the severity of the debugging required to track down an issue and the number of times you must perform this debugging process, it can ultimately force you to slip on sprint deliverables. In turn, this can have a cascading effect on the overall product delivery timeline. With the wasted resource time tracking down build issues, you also delay delivery of your product, which can harm your profit margins, etc.



# Test Data Management Techniques

---



# Do It Yourself & Leave No Trace

Building test data and garbage collection is sometimes required to ensure that a scenario has everything it needs to perform the use case being tested. Whether you're creating new data or changing existing, you should always try to leave your test data in the same form as it was when you entered your test case.

In many cases, test data may be changed or massaged to ensure it meets the use case requirements. It's good practice to get the test data values before execution and store those values to reset the data back to its original state on the test case exit. Ensuring that the next test relies on that same data can expect it to be in an appropriate state. This holds when dealing with any data/SQL-related resources being allocated for the test case. Not every data-related API handles the proper cleanup of its resources, so it's always good practice to perform your own object cleanup.



# Access Data Related Proprietary APIs



Another option is to utilize the company's proprietary APIs with the test data in question. For example, Retailers have test data in the form of inventory and store information (including locations); your company may already have APIs developed that accesses and modifies this type of data.

Having access to APIs can be useful when setting up your test data for a given scenario and can, in some cases, save time not creating from scratch.

# Designate Automation Specific Test Data



It's a prevalent scenario that both manual and automated tests share the same test data pool. Another way to mitigate unstable build environments is to set up a list of test data resources that are off-limits to all except for automated test cases executed as part of the CI/CD build process.

By doing so, you significantly reduce the risk of test data becoming tainted by other testers attempting to use the same resource.

# Virtualization & Mocking




Mocking and/or virtualization software can be used to avoid test data issues within your organization. The idea of mocking is nothing new and has been around for quite some time. But the idea of using virtualization software to deal with potential automation pitfalls is growing in momentum.

Say, for example, you are testing a restful service. And that service relies on another service outside of your teams' control. You could use virtualization software such as Parasoft Virtualize to mimic the restful call from that service with a set of given responses based on your test scenario or the passed in the payload.

Other types of tools and software do similar things, including Wiremock, a lightweight NodeJS server, or even plain files on Nginx. This gives you the ability to take the guessing out of your automated testing and build a bit more stability into your pipeline.

# Bringing it Back to Earth...

## Retail Senario

Let's take a hypothetical look at a company struggling with its automation pipeline and ways to combat its issues. 

ACME Company has implemented a typical CI/CD pipeline. They have a couple of lower environments; dev and test, where the first stage of development and testing is performed. ACME also has a staging environment that is the final gated staging area until code gets pushed to production. Their testing department consists of both manual and automated test engineers, and both sets of testers are using the same pool of data for their test cases. This is a pretty common practice and nothing new in the software world, but it's also a scenario that poses significant challenges.

When the manual testers start their UI testing of the company's product, they directly affect the data being used to validate the backend API and methods used by the UI. They run into the issue where the automation fails because the test data has been altered by the manual testers or other automated test cases.

**A robust test data management approach avoids these problems before we even blast off and lets us achieve lighting-speed CI/CD runs.**



# Bringing it Home!

The ability to deliver a quality product, on time, and within resource constraints, means trusting the CI/CD pipeline to report the true nature of work and the entire software delivery life-cycle (SDLC).

False failures due to poor test case design and test data management can lead to time delays and misuse of development resources, all of which can result in a less than desirable bottom line. But with smart test case design techniques, your team can maintain stability and assurance within the software delivery systems.

AIM Consulting's established methodology for creating a successful CI/CD pipeline and test automation framework has been utilized across a multitude of industries, including; retail, consumer goods, healthcare, manufacturing, entertainment/media, and food services.

We are ready to support your team, business objectives and drive actionable results. Want to learn more? [Request a consultation.](#)



# Meet the Mind Behind the Magic...



# Patrick Taylor

Principal Consultant, Application Development

Patrick Taylor is a principal consultant with over 17 years of professional experience in testing, automation, quality management, and leadership. Patrick has used his passion for technology and quality at all levels of development and testing organizations from individual contributor to Director of QA.

Prior to AIM Consulting, Patrick owned a software consultancy where he offered services in test automation, technical product management, mobile application development, and more. He was also an SDET at Custom Control Concepts where he built automation and test utilities for cabin control and entertainment systems for private and VIP aircraft. When not squashing bugs in code, Patrick enjoys spending time with his family at their home in West Seattle, listening to experimental music, cooking, spending time in the outdoors, and playing in film and video arts.



# About AIM Consulting



AN ADDISON GROUP COMPANY

AIM Consulting, an Addison Group company, is an award-winning industry leader in technology consulting and solutions delivery. AIM's differentiation is our collaborative engagement model that provides cross-functional results. We work with clients, shoulder to shoulder, for one goal – their success. Founded in 2006, with offices in Seattle, Minneapolis, Denver, Houston, and Chicago, we are ranked among the fastest growing private companies and best companies to work for due to a long track record of success with our partners and consultants. Our long-term relationships with the best technology consulting talent allows us to deliver on expectations, execute on roadmaps, and drive modern technology initiatives.

