1. Introductory Story (Team not having a way to run tests after someone left)
2. Introduction: Test Automation Delivery Channel
3. Parts: Test Automation Delivery Channel
4. Your Benefits Of A Test Automation Delivery Channel
5. Pitfalls To Avoid
6. How Do We Setup A Test Automation Pipeline? (DEMO)
7. Finishing Story & Call To Action

-Story: Don't destroy your automation effort, set up the pit of success!  
-A delivery channel outlines the way a test suite is kicked off, where it runs, and how the test results are compiled  
-Pipe within the PipeLine!  
1. Avoid BottleNecks  
2. Avoid Useless Re-factoring  
3. Decrease the feedback loop  
4. Consistent Environment  
  
Introductory Story:  
  
1. Formation of the idea of test automation (Beginning)  
2. Initial successes (tests!) (Beginning)  
3. Execution failures (Middle)  
4. Introduction of Solution (Automation Channel)  
  
Let me tell you about a team which really wanted to "shift left" or "automate all the things", and gain the ability to release with quality in a reasonable amount of time. The team was newly formed and did not have a lot of expertise but had a lot of passion, and decided to go all in on automated testing. The team was a very typical cross functional team, full of excitement about automating anything and everything! Builds, tests, and most importantly features!  
  
The team was working diligently on picking the right tools and the right tests. They settled on Selenium and the page object modelling pattern. The tests were being developed and ran by one person. They were running, and working.   
  
Shortly after the first series of tests were completed, the engineer who wrote them went on maternity leave. Even though the tests were written in the same language as the team was developing, the same IDE and in the same project area, they were not executed nor were they maintained.  
  
What could have been done to make this situation better? How could that engineer have made sure her legacy continued while she was not there? We all know the importance of writing tests that are maintainable, and easy to run, but how do we ensure this happens?   
  
Enter the automation channel.  
  
An automation channel creates a stable, repeatable way to run automated tests. It's a way to ensure your team's automation efforts are setup for the pit of success, not the pit of failure.   
  
In this talk we are going to discuss the benefits of setting this channel up early, and examples of how to do it for different popular framework and test combinations.   
  
Theory of A Test Automation Delivery Channel  
-Parts  
\*Controller  
\*Agents  
  
-Key points  
\*Early!  
\*Independent   
\*Easy  
  
-Benefits  
\*Re-use for DevOps pipeline  
-Pitfalls   
  
Setup Steps (Environments to use, frameworks to adopt)  
  
Difficulties To Get Over   
  
-Demo of Setup Using Free Tools (VSTS, C#)  
  
Finishing Story  
  
After it became apparent to the team that the tests which were created could no longer be run, but the team wanted to maintain a high level of quality via automation, something had to be done. Scripts existed, but they were not running, what were they to do? A Test Automation Delivery Channel had to be created. The team decided to use TFS, SauceLabs and Selenium. In a few days a delivery channel which allowed for anyone on the team to execute the tests was created. A trigger was set, and the team gained a stable, reproducible way to run tests.   
  
Call To Action  
-Every project where test automation exists requires a way to run tests from the beginning of the project  
-Build an automation test delivery channel (automation pipeline) as a first step in your automation effort  
-Invest early, and remove ambiguity. Allow your tests to fall into the pit of success!

Presentation Outline

1. Introduction Story
   1. Team Without Test Automation Delivery Channel
2. Test Automation Delivery Channel: What Is It?
   1. Concept
   2. Parts
3. Test Automation Delivery Channel: What are the parts?
   1. Controller
   2. Feedback Area
   3. Environment To Test On
4. Test Automation Delivery Channel: Decrease Your Quality Feedback Loop
   1. Eliminate Bottle Necks
   2. Avoid Useless Refactoring
   3. Decrease Test Run Time
   4. Decrease Environment Issues
5. Test Automation Delivery Channel: Walk Through Of Setup (of one)
6. Conclusion Story & Call To Action
7. Intro Story:

Press Button Get Test Execution

Press button run one test

Press button run many tests

Press button run one test on many computers

Press button run many tests on many computers

Press button run many tests on many computers in a short period of time?

1. Story: Most people can do a lot of these steps, but this common story happens, we want to get to holy grail

-We had the same problem and we took it upon ourselves to fix it.

-How does a new software engineer run tests

-How does a new QA run these new things?

1. Introduce Solution: Test Automation Delivery Pipeline
2. Solve Problem: Push Button Get Test,
3. Go back to problem, and give solutions for each step. “Let’s go back and start from the solution”

-How many of you are business analysts, how many business analysts can run tests? So you can tell the business you are ready to deploy.

\*BUILD SOLUTION TO PROBLEM through videos of problem step solutions

6. Conclusion: What have we caught, what we’ve run, what are our results.

“If you’re not completely sold, check this out”

RESEARCH THE TERM “Automation delivery channels”.

REMEMBER WHAT I TOLD YOU? Here are some of the benefits.