

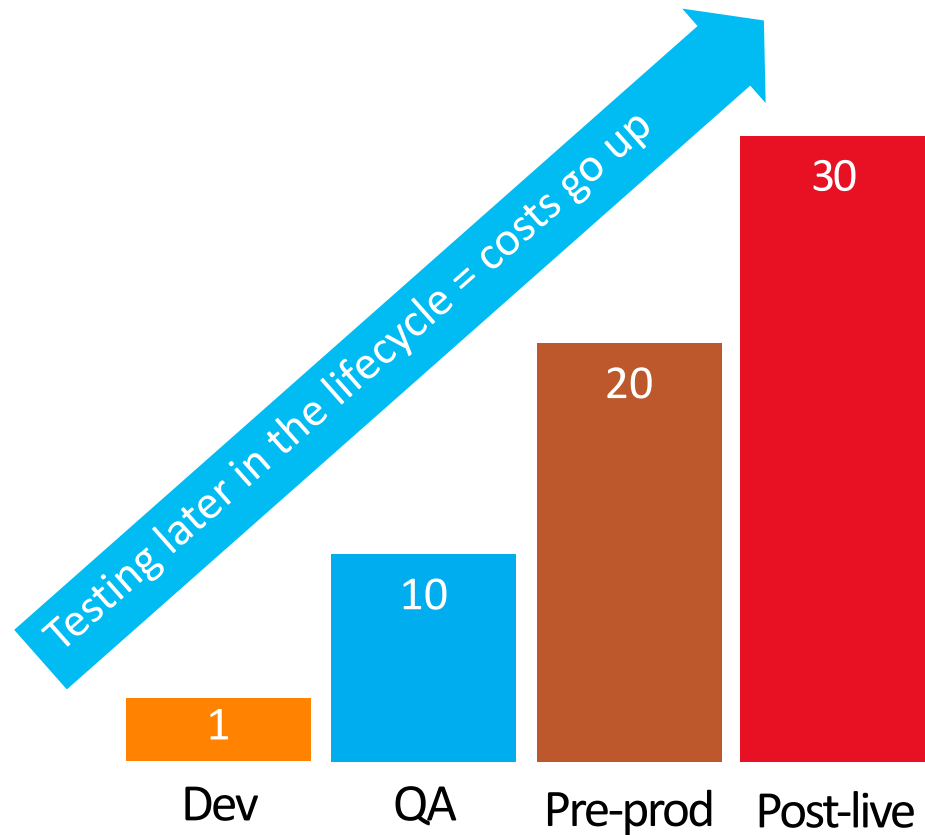
Test and Attack

THE ART, SCIENCE AND MYTHS OF PERFORMANCE TESTING

Presented by Geoff Gray
Senior Perf Test Consultant
geoffgr@Microsoft.com

Did you know? Cost of NOT testing

These numbers are from 2012



\$59 B

Annual cost for software failure to economy



\$1 B

Lost in the past 20 years in aerospace industry



1 in 10

2012 Olympics seats left empty due to ticket website crashes



Yep.. This really happened

Healthcare.gov plagued with issues

*This happened 5 months after the initial failure.
I guess they didn't learn from their mistakes.*

Politico reports that HealthCare.gov was down for six hours on Monday morning. Health and Human Services officials told ThinkProgress that the multiple website issues have now been resolved. “The tech team monitoring HealthCare.gov in real time identified an issue with users creating new accounts.”

<http://thinkprogress.org/health/2014/03/31/3421103/obamacare-site-glitches-deadline/>

Yep.. This really happened

A \$440 Million Test “Glitch”


It took only 45 minutes for the software to cause an entire day of bad trading across 130 stocks

Knight Capital, a firm that specializes in executing trades for retail brokers, took \$440m in cash losses Wednesday **due to a faulty test of new trading software**. This morning reports were calling it a trading “glitch”, which isn’t nearly as accurate as the term I’d use: “****ing disaster”.


http://www.theregister.co.uk/2012/08/03/bad_algorithm_lost_440_million_dollars/

Eschew Obfuscation

What The

es·chew  **verb** \e-'shū, i-; es-'chū, is-; *also*
e-'skyū\

: to avoid (something) especially because you do not think it is right, proper, etc.

ob·fus·cate  **verb** \'äb-fə-,skāt; äb-'fəs-,kāt, əb-\

: to make (something) more difficult to understand

Eschew Obfuscation = “avoid making things more difficult to understand”.

About Load and Performance Testing:

- The Art
- The Science
- The Myths

Test and Attack

**“Testing With
the Customer”**
instead of
**“Testing For
the Customer”**

- The Art
- The Science
- The Myths



The Thrill of Victory

- NORAD Santa Tracker
- HALO Reach launch
- The 7 Terabyte Switch
- The Suspect Simulator
- The “Whiteboard” test plan

And the Agony of Defeat

- You say Tomato, I say Tomahto.
- Man, that's a heavy test harness.
- All we care about is throughput.
- Let the System Tell You
- Too Fast
- The 3.5 minute delay ([TcpTimedWaitDelay](#) to the rescue)

What is really true Grasshopper?

- Visual Studio must be broken (*The 90-Percentile conundrum*)
- Garbage Collection run Amuck
- It's a bird....
 - No, it's a plane....
 - No, it's a bad requirement!
- A requirement without a consequence is just a desire.
- An SLA is NOT a goal.

Test and Attack – The Process

Process and Methodology

Plan

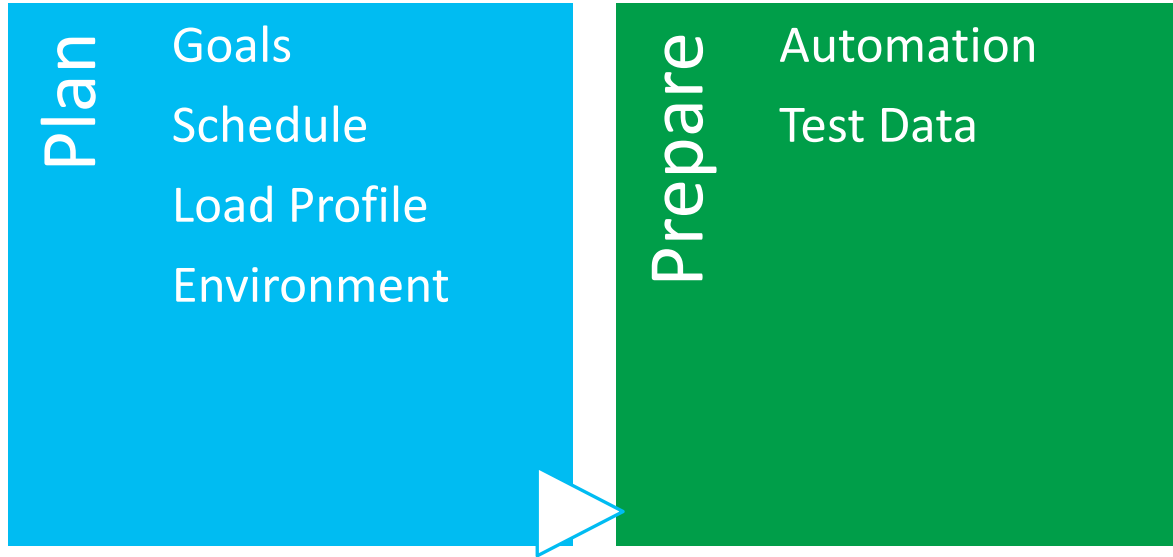
Goals

Schedule

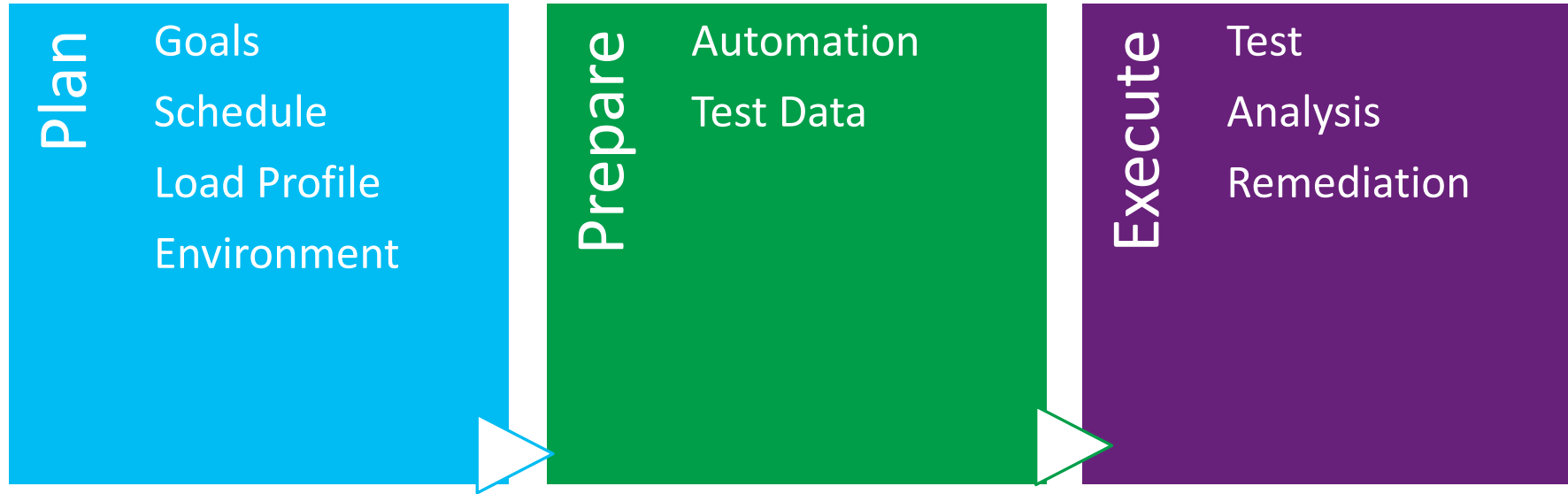
Load Profile

Environment

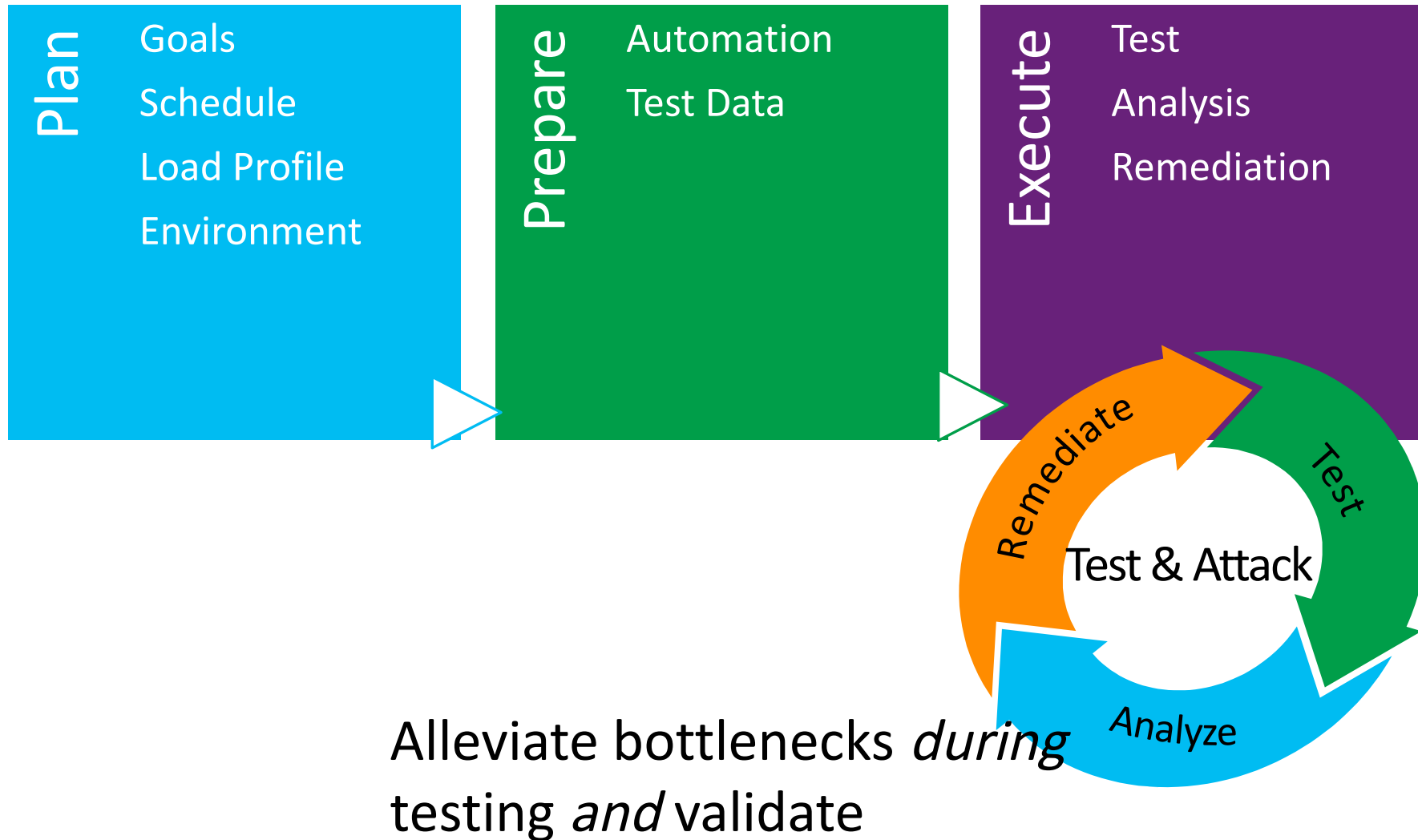
Process and Methodology



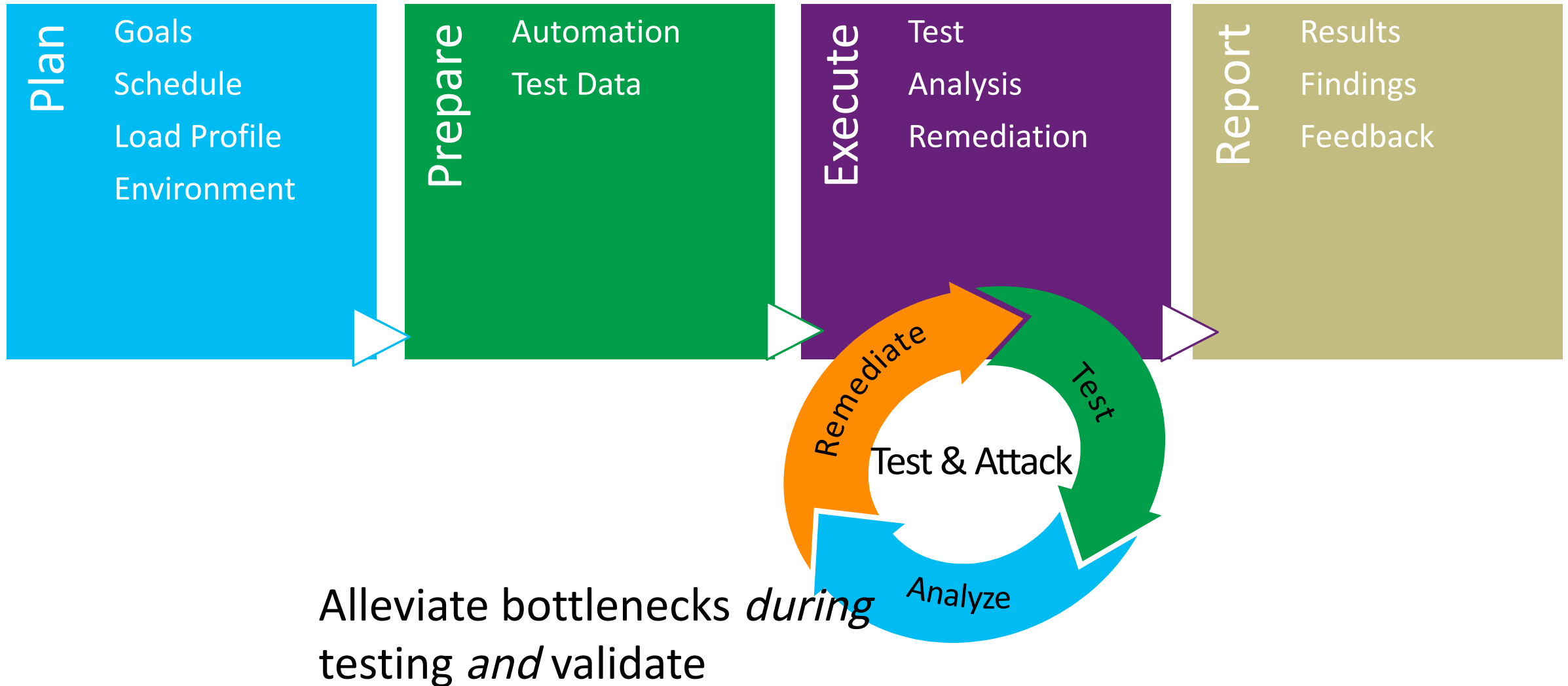
Process and Methodology



Process and Methodology



Process and Methodology



The Science – The Process Details

Plan

Define goals and requirements

Task: Develop detailed written test plan:

- Goals and success criteria

- Test scenarios, use cases, load profiles

- Test environment(s)

- Key metrics

- Initial daily plan for execute phase

- Schedule and task assignments for subsequent phases

Plan

Goals

Schedule

Load Profile

Environment

Key Outcome:

Test Plan

Key Participants:

Business Partners, Key Stakeholders,
Development and Test, Infrastructure Support

Typical Duration:

One to two weeks

Test Plan Document

1. Executive Summary
2. Engagement Criteria
3. Testing Criteria
4. Data and Load Considerations
5. Application Information
6. Real World Use Cases
7. Test Harness Use Cases
8. Action Items List
9. Appendix

Prepare

Create artifacts and prepare for Execute phase

Tasks:

Test automation

Test data

Tools and processes

Application setup process

Refine test plan

Prepare

Automation
Test Data

Key Outcome:

Test Plan

Test Automation

Key Participants:

Development and Test

Infrastructure Support

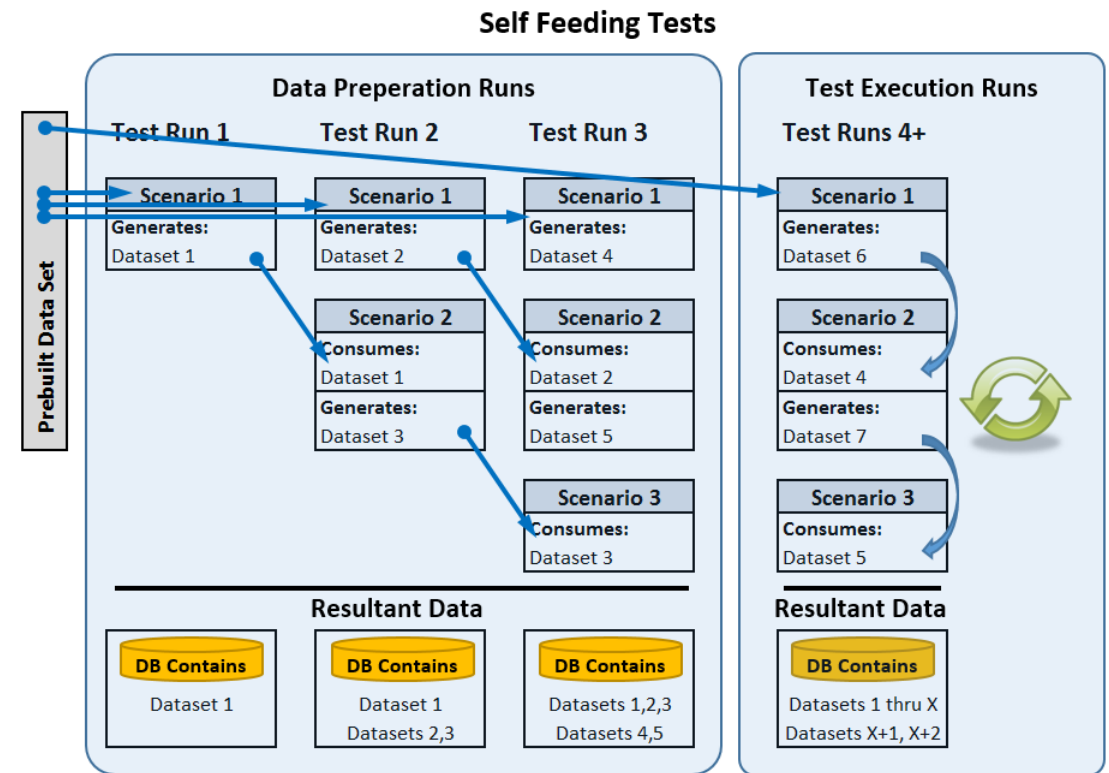
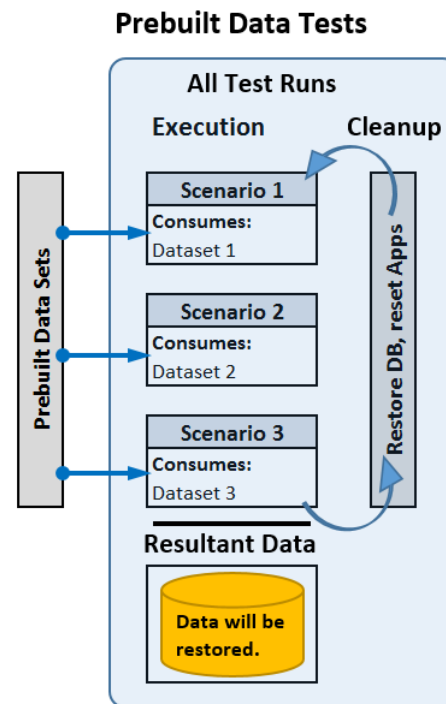
Typical Duration:

Two to eight weeks

Data

Is the harness:

- Pre-Populated?
- Self Feeding?



Execute

Test sprint to implement the test plan

Tasks:

Execute test

Analyze data and results

Make changes

Track progression

Update bug database

Key Outcome:

Test Results

Bugs and Fixes

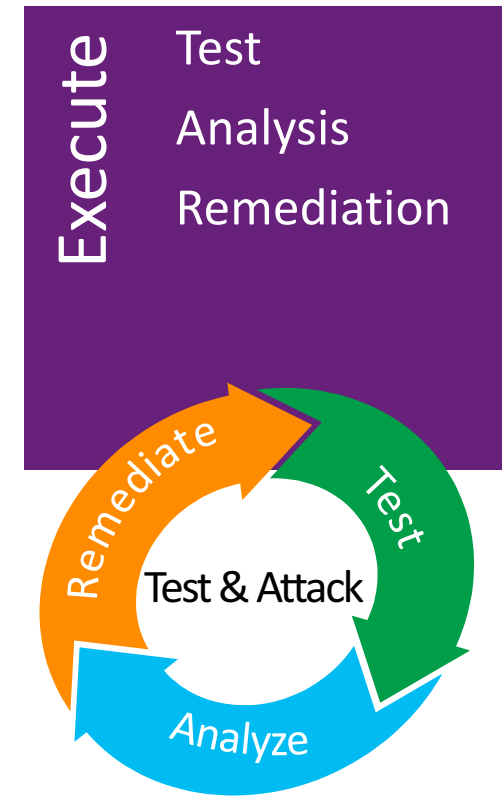
Key Participants:

Development and Test

Infrastructure Support (for initial setup)

Typical Duration:

Two to three weeks



TORs and Break/Fix Runs

- Test of Record
- Break/Fix
- *“Give me a ping Vasili. **One Ping Only.**”*

“Fire in the Hole”

- Pre-Flight Checklist
 - Servers Restarted?
 - SQL DMVs Cleared?
 - Data Restored?
 - Test Rig reset?
 - Data Collection enabled?
- ***“Fire in the Hole”***

“Today’s In-Flight Movie”

- Main Key Metrics Graphs
- Summary Data
- Time Remaining
- Etc.

“Pack ‘em, Stack ‘em and Rack ‘em”

- After the run completes:
- Data collection
- System Cleanup
- Data Processing
- Results Analysis

“Graphs, Tables and Data Points, Oh My!”

- Graphs – For Trending
- Tables – For Criteria Comparison
- Data Detail Points – For Diagnosing

Daily Standup

- 10 minutes max
- On track?
- Need to shift focus?
- Plans for the day?

Report

What was learned and accomplished

Task: Develop final report:

- Key finding and changes

- Detailed test results

- IP developed during engagement

- Additional application recommendations

- Testing process recommendations

- Feedback to improve future efforts

Report

Results

Findings

Feedback

Key Outcome:

Engagement Summary
Report

Key Participants:

Development and Test
Business Stakeholder(s)

Typical Duration:

One week

The Art – Managing the Process

Caveats

- Discipline
- Scope Creep
- Time

Resources

➤ Documentation on Test Planning

➤ <https://blogs.msdn.microsoft.com/geoffgr/category/planning/>

➤ Documentation on VS Load Test Results

➤ <https://blogs.msdn.microsoft.com/geoffgr/category/understanding-results/>

Questions?
