# Practical Metrics for Managing and Improving Software Testing

## Presented By: Shaun Bradshaw

shaun.bradshaw@zenergytechnologies.com

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

- ## Part 1 – Test Metrics
  - Ten key metrics testers should track
  - One bonus metric
- ## Part 2 - GQM
  - Aligning test metrics with business goals
  - Defining meaningful metrics
- ## Part 3 – S-Curve
  - Plotting the test execution effort
  - Managing the effort with metrics

*Software Peace of Mind in Action*

# Test Metrics

Software Peace of Mind in Action

- All testers track some type of metrics
- Common metrics include:
  - Number of requirements to test
    - by function      - by priority
  - Number of test cases
    - created          - executed
    - passed           - failed
  - Number of defects
    - by severity      - by status

- These measurements are adequate, but require more context to be useful

- Test metric context occurs when multiple measures are used together (formulas)

- Test metrics must provide actionable information to be useful

*Software Peace of Mind in Action*

- Why are test metrics important?
  - Improvement comes from understanding
  - Answers questions important to our development process*
    - Which areas have the most defects?
    - How long does it take to repair defects?
    - Which areas have the highest re-work rates?

*Sample questions from https://wiki.mozilla.org/QA:Metrics%26Coverage

## Test metrics provide two types of information:

| Progress | Quality |
|---|---|

## Measures

| Metric | Value |
|---|---|
| #  TCs to be Executed | 1353 |
| # Executed | 1286 |
| # Passed | 1221 |
| # Failed | 64 |
| # Re-executed | 82 |
| Total Executions | 1414 |
| Total Passes | 1222 |
| Total Failures | 157 |
| 1st Run Failures | 133 |

## Test Metrics

| Metric | Value |
|---|---|
| % Complete | 90.2% |
| % Test Coverage | 95.0% |
| % TCs Passed | 94.9% |
| % 1st Run Failures | 10.3% |
| % Failures | 11.1% |
| % Defects Corrected | 59.2% |
| % Rework | 10.0% |
| % Bad Fixes | 18.0% |
| Defect Discovery Rate | 0.52 |
| % Test Effectiveness | 98.7% |

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

## % Complete

- # Passed / # To Be Executed

- Indicates readiness for deployment to production

By: Component, Tester, Priority

## % Test Coverage

- # Executed / # To Be Executed

- Provides summary of known test execution results

By: Component, Priority

zenergy TECHNOLOGIES

## % TCs Passed

- # Passed / # Executed

- Provides trending information to ensure no major issues hinder testing

By: Component, Team

## % 1$^{st}$ Run Failures

- 1$^{st}$ Run Fails / # To Be Executed

- Provides indication of how clean code is on delivery to test
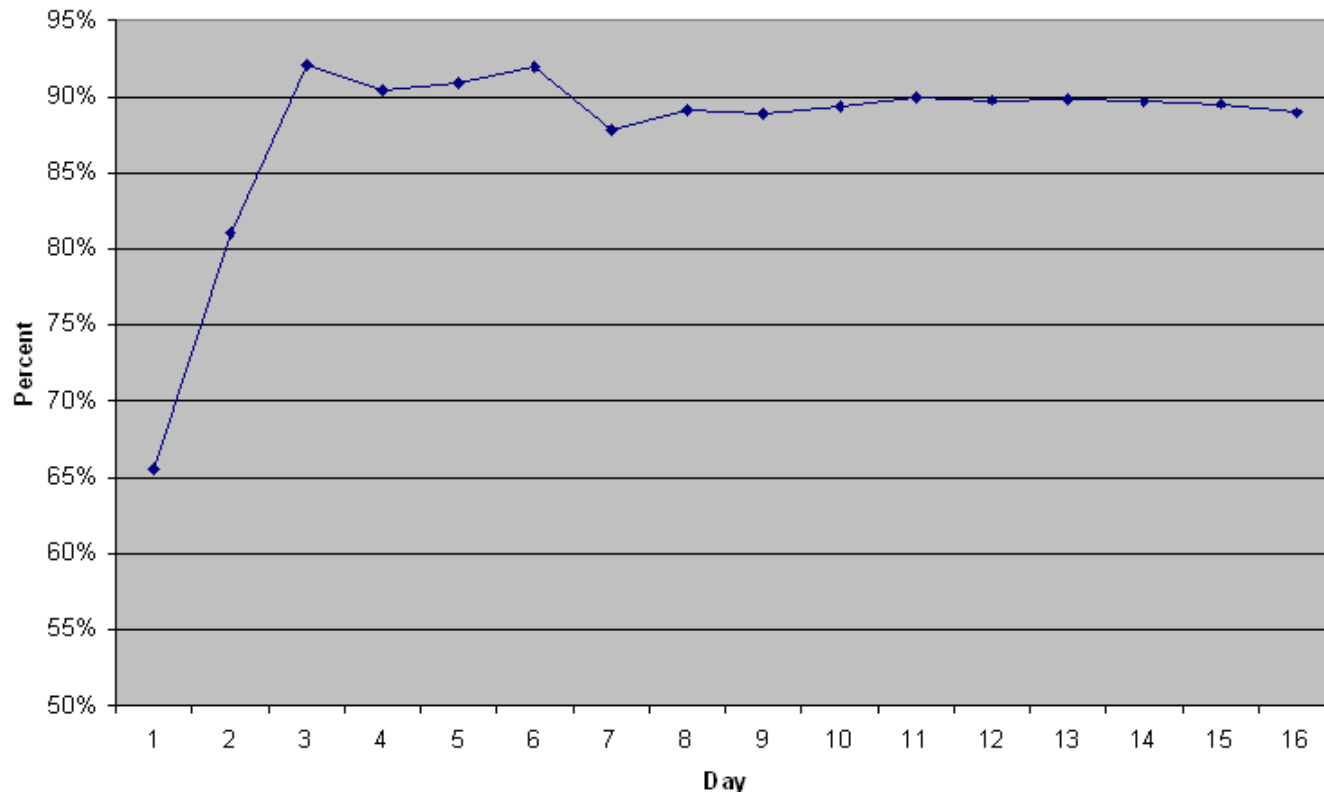
By: Component, Team

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

## % TCs Passed

## % 1st Run Failures

- # Passed /
  # Executed

- Provides tre
  information
  ensure no n
  issues hinde

By: Component, Tea

**Test Cases Passed Trend**

## % Failures

- Total Failures / Total Executions

- Indicates quality of the application and used for historical purposes

By: Component, Priority, Severity

## % Defects Corrected

- $1 - (Defects_{Open} / Defects\ Found)$

- Indicates production deployment readiness

By: Component, Team, Severity

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

## % Rework

- (Total Executions / # TCs Executed) / # TCs Executed

- Indicates quality of the application development process

By: Component, Team

## % Bad Fixes

- (Total Failures – $1^{st}$ Run Failures) / $1^{st}$ Run Failures

- Indicates quality of defect corrections

By: Component, Team, Severity

zenergy TECHNOLOGIES

## Defect Discovery Rate

- Defects Found / Man Days in Effort

- Indicates value of the development processes

By: Component, Team, Tester, Severity, Phase/Iteration

## % Test Effectiveness

- $\text{Defects Found}_{Test} / (\text{Defects Found}_{Test} + \text{Defects Found}_{Production})$

- Indicates quality of the test process

By: Component, Team, Tester

**zen**ergy TECHNOLOGIES

# Quality Index

- $(1^{st}$ Run Fail Rate$_{Index}$ * $1^{st}$ Run Fail Rate$_{Weight})$ + (Sev 1 Defect Rate$_{Index}$ * Sev 1 Defect Rate$_{Weight})$ + (Bad Fix Rate$_{Index}$ * Bad Fix Rate$_{Weight})$

- Provides standardized method of comparing projects

| Quality Index | | | |
|---|---|---|---|
| **Metric** | **Index** | **Weight** | **Max** |
| 1st Run Failures | 1 | 50% | 100% |
| 1st Run Failures | 2 | 50% | 75% |
| 1st Run Failures | 3 | 50% | 50% |
| 1st Run Failures | 4 | 50% | 25% |
| 1st Run Failures | 5 | 50% | 10% |
| Sev 1 Defects | 1 | 35% | 100% |
| Sev 1 Defects | 2 | 35% | 50% |
| Sev 1 Defects | 3 | 35% | 30% |
| Sev 1 Defects | 4 | 35% | 20% |
| Sev 1 Defects | 5 | 35% | 10% |
| Bad Fixes | 1 | 15% | 100% |
| Bad Fixes | 2 | 15% | 75% |
| Bad Fixes | 3 | 15% | 50% |
| Bad Fixes | 4 | 15% | 25% |
| Bad Fixes | 5 | 15% | 15% |

**zenergy TECHNOLOGIES**

- The metrics shown above provide a strong starting point for any organization eager to start a test measurement program.

- The next step must ensure tracked metrics fully align with business goals...

*Software Peace of Mind in Action*

# Goal / Question / Metric

*Software Peace of Mind in Action*

- Goal (Conceptual)

  – Define what the organization wishes to accomplish relative to products, processes, and resources

- Question (Operational)

  – Used to understand how to meet the goal and determine quality

- Metric (Quantitative)

  – Set of data associated with questions to answer them quantitatively

- Benefit

  – Ensures test metrics align with organizational goals and provide important, meaningful, and actionable information

*Software Peace of Mind in Action*
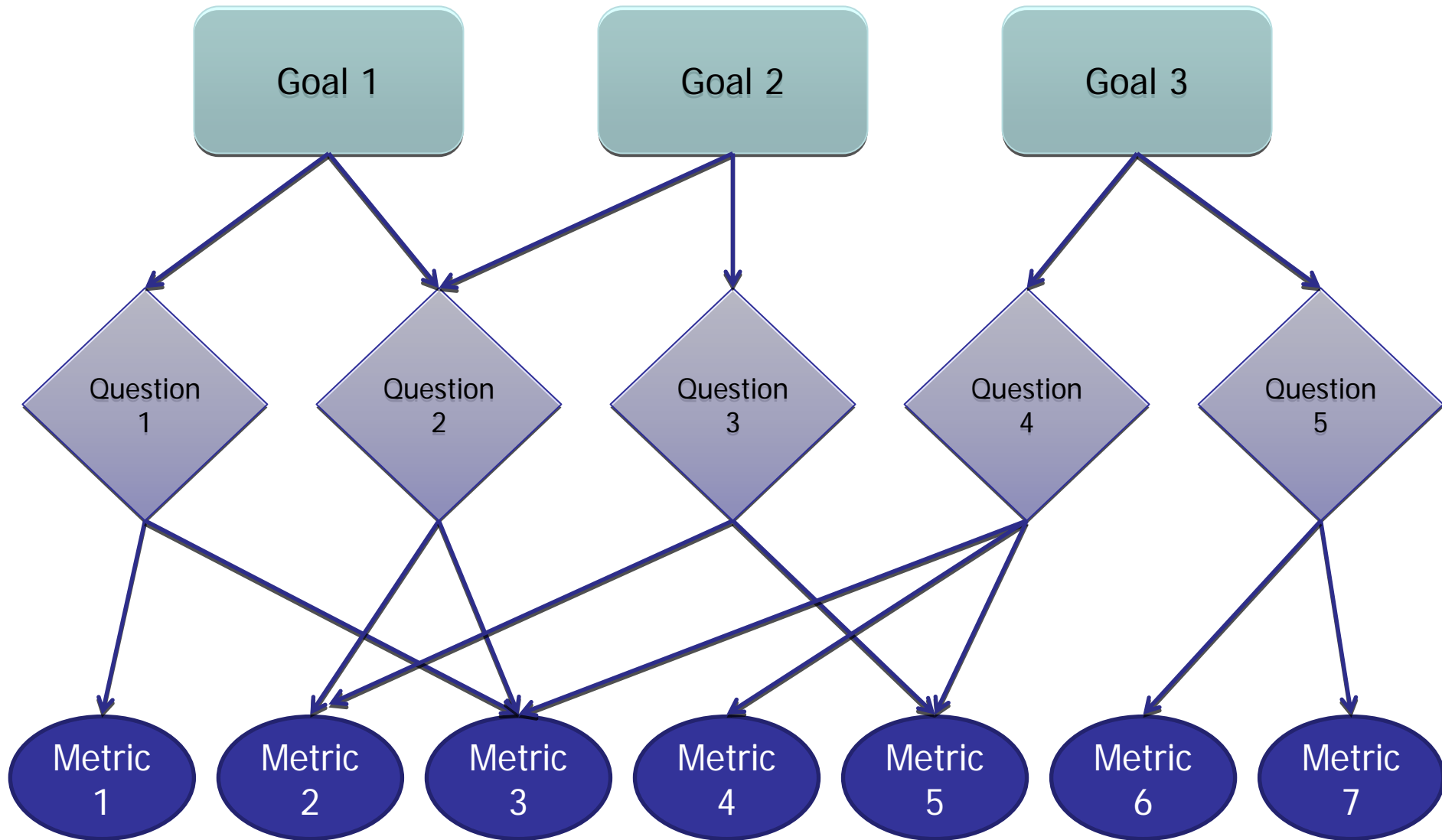
- Goals
  - Identify components and files that require more community support
  - Reduce the number of feature bugs
- Questions
  - Which areas have the most defects?
  - How long does it take to repair defects?
  - Which areas have the highest re-work rates?

- Metrics
  - Trend graphs of coverage (all bugs filed)
  - Trend graphs of coverage (all bugs fixed)
  - All bugs : Filed Vs Confirmed by SEV or PRI
    - Filed vs Confirmed is when a volunteer finds and files a bug, but it stays unconfirmed with respect to developers until it has gone through triage and been confirmed.
  - All bugs : NEW Vs FIXED by SEV or PRI
    - The NEW vs FIXED provides the mean time to fix a bug and helps to develop heuristics on the bug fix capacity of a given group.

*Example from https://wiki.mozilla.org/QA:Metrics%26Coverage*

## Establish Goals

- Identify business goals and the measurement goals derived from them
- Determine the Object, Purpose, Focus, Viewpoint, and Context of each goal

## Generate Questions

- Clarify and refine the goals (transition from conceptual to operational level)
- Identify the nuances and perceptions related to the goal
- Provide common understanding and interpretation of the goal

*Software Peace of Mind in Action*

**zen**ergy TECHNOLOGIES

## Specify the Measures

- Determine how the questions can be answered (move from operational to quantitative level)
- Involve stakeholders to minimize ambiguities and false assumptions
- Identify base measures, derived metrics, and indicators

## Prepare for Data Collection

- Develop procedures for data collection (*Measurement Plan*)
- Identify roles responsible for data collection
- Determine what tools can/will be used for collection

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

## Collect, Validate & Analyze Data

- Automate where/when possible
- Ensure data is correct, complete, and consistent
- Analyze and interpret results; communicate results to stakeholders

## Implement & Improve

- Utilize data results to identify areas of improvement
- Recommend changes to affect positive change
- Measure results of changes

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

Get the right people involved from the beginning

Those involved interpret the metrics

Integrate and automate metrics activities

Set & state explicit measurement goals

Consider context and stay focused on the goal during analysis

Do not use metrics for other purposes

Don't create false measurement goals based on data available

Identify implied quality models

View measurement as a tool not the goal

*Software Peace of Mind in Action*

**zen**ergy
**TECHNOLOGIES**

- **GQM Plan**

  - Contains each measurement goal and its corresponding questions and metrics

- **Measurement Plan**

  - Defines the base measures, derived metrics, and indicators identified in the GQM Plan

  - Establishes procedures for collecting and calculating the metrics

- **Analysis Plan**

  - Documents the methods to analyze, aggregate and present metrics information in a meaningful way to the stakeholders

  - May be documented as a dashboard

## Goal
Decrease the walking effort necessary to travel from the parking lot to the airport terminal.

| Object | Purpose | Focus | Viewpoint | Context |
|---|---|---|---|---|
| Walking | Reduce | Effort | Personal | Parking lot to terminal |

| Questions | Metrics |
|---|---|
| • How do we measure *effort* <br> • What is the least effort necessary <br> • Should luggage be a factor?  If so, how many pieces and of what size | • Travel Distance <br> • # of bags <br> • Caloric Output <br> • Weight / bag <br> • Travel Time (total & average) <br> • Total baggage weight |

| Baseline Hypothesis | Variation Factors |
|---|---|
| • Average walking time is 8 minutes <br> • Effort and time increases by 1 minute for every 20lbs of luggage | • Flight delays <br> • Traffic delays <br> • Parking lot fullness |

| # of Defects | | | | | | | *Progress / Quality* |
|---|---|---|---|---|---|---|---|
| This metric provides base data for calculating additional metrics related to the quality of the application and progress of the development effort. | | | | | | |
| **What** | A defect is counted when the expected results do not match the actual results. | | | | | | |
| **Who** | Role | Collect | Consume | **When** | • Daily<br>• Weekly<br>• Monthly<br>• Quarterly<br>• Semi-Annual<br>• Yearly | **Where** | • QC<br>• TFS<br>• ClearQuest<br>• Excel |
| | QC Analyst | X | X | | | | |
| | QC Manager | X | X | | | | |
| | Developer | X | X | | | | |
| | Lead Dev | X | X | | | | |
| | Project Mgr | X | X | | | | |
| **How** | Defects must be captured in a defect log or tool in order to be counted. The tools used should allow filtering by context. | | | | | | |
| **Context** | • Test Type (Regression, Performance, Shakedown, Final Acceptance, etc.)<br>• Status (New, Open, In Progress, Retest, Reject, Fixed, Closed, Deferred)<br>• Severity (Critical, High, Medium, Low)<br>• Root Cause (Missing Code, Incorrect Code, Environment, Documentation)<br>• Project/Release (varies)<br>• Phase (Analyze, Design, Build, Test, Rollout, Production) | | | | | | |

Use one of the following business goals to determine questions and metrics from a test perspective:

Increase customer satisfaction with the application

Improve market reputation and industry recognition

*Software Peace of Mind in Action*

- The GQM method of metrics development ensures that test metrics align with what's important to the organization

- By accounting for business goals when framing our metrics we make our group more valuable and better able to exact changes deemed useful to the organization as whole

- Once you have your metrics in place, its time to use them...

*Software Peace of Mind in Action*

# S-Curve

*Software Peace of Mind in Action*

# What is an S-Curve?

- An S-Curve is a graphical representation of cumulative work effort

- Used to plan and manage test efforts by:

  - Measuring progress against a theoretical effort

  - Determining stability of the application prior to release to production

*Software Peace of Mind in Action*

zenergy
TECHNOLOGIES

## Why is a test effort an "S" shape?

- Test execution starts slowly as the team works through configuration issues and major blocking defects

- Once initial issues are resolved a larger variety of tests can be executed, increasing test execution velocity

- As testing nears release, there are fewer tests to be executed and only a few defects remain outstanding, leveling out the speed of execution

*Software Peace of Mind in Action*

**zenergy** TECHNOLOGIES

$$\% \text{ Tests Executed} = \frac{DE}{DE + e^{(3-8\,*\,DE)}}$$

- The theoretical curve is derived from the formula above, which determines the cumulative percentage of tests that have been executed on any given day of the test effort

- Plotting the points for each day's percentage results in an uniformly distributed curve indicating "optimum" test progress

- $DE$ = Current Day Number of Effort / Total Days in Effort
- $e$ is the natural logarithm (2.71828182845904)
- $3$ and $8$ are constants which set the curve points

*Software Peace of Mind in Action*

| S-Curve Calculations - Passed | | |
|---|---|---|
| # Days | # TCs | |
| 15 | 100 | |
| | Theoretical Curve | |
| 1 | 0.56% | 1 |
| 2 | 1.89% | 2 |
| 3 | 4.70% | 5 |
| 4 | 10.08% | 10 |
| 5 | 19.28% | 19 |
| 6 | 32.82% | 33 |
| 7 | 49.28% | 49 |
| 8 | 65.43% | 65 |
| 9 | 78.40% | 78 |
| 10 | 87.30% | 87 |
| 11 | 92.80% | 93 |
| 12 | 96.00% | 96 |
| 13 | 97.79% | 98 |
| 14 | 98.78% | 99 |
| 15 | 99.33% | 99 |

Theoretical curve for a 15 day test effort with 100 tests to be executed.

**Test Metrics Graph - Passed**
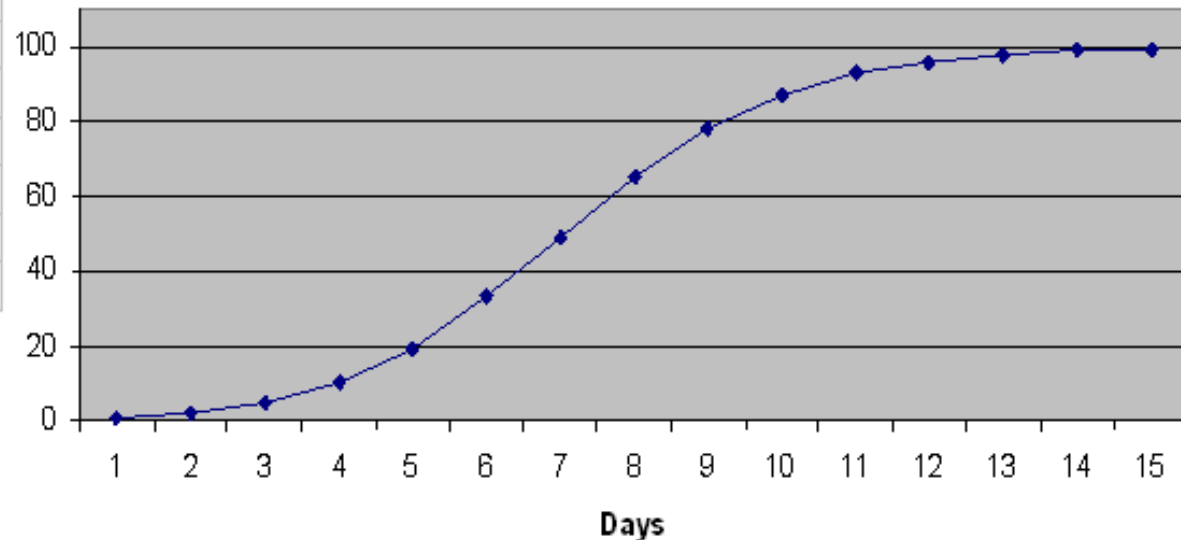
By plotting the actual cumulative number of tests passed and comparing it to the theoretical curve we are able to identify potential issues and make adjustments to the effort to ensure testing is as successful as possible.



**Test Metrics Graph - Passed**

*Soft*
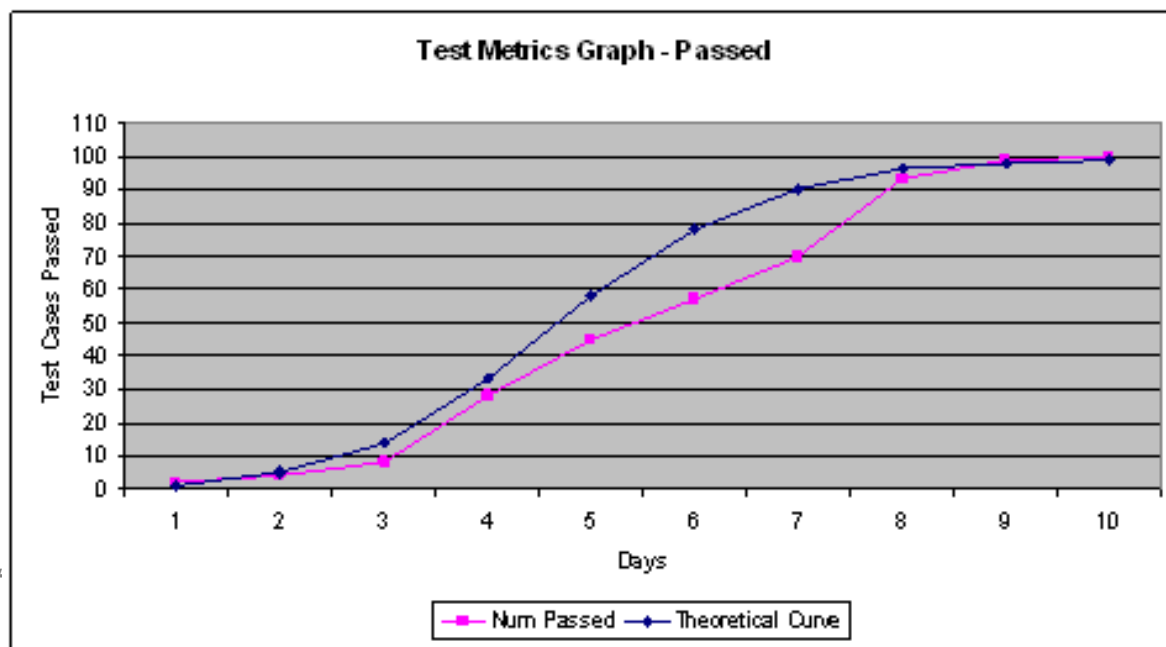
- **Background**
  - Best-in-class manufacturer is nearing the end of a 3-year implementation of a highly-configurable ERP package
  - Prior to release of the "Global Template" to the first manufacturing market (Asia), the team chose to execute a full regression of the following types of tests:
    - Functional – executed and validated by single tester
    - End-to-End – executed and validated by multiple testers
    - Accounting Process Flows – executed and validated by multiple testers

- Team
  - No professional testers
  - 30-40 SMEs in the areas of:
    - Supply Chain Planning, Supply Chain Management, Manufacturing, Operations, Sales (divided by product category), Accounting, etc.

- Test effort
  - Complete manual test execution of 370 E2E, 390 functional, and 600 process flow test cases in 15 days
  - Daily stand-up held to review metrics and adjust test execution strategy

The following are snapshots of the S-Curve throughout the test effort with an interpretation and directions provided based on analysis of the curve and other information available.

*Software Peace of Mind in Action*

## Day 1

Good start, as expected. Since this is a regression test effort most environmental issues should be taken care of, as well as the few, if any major blocking issues
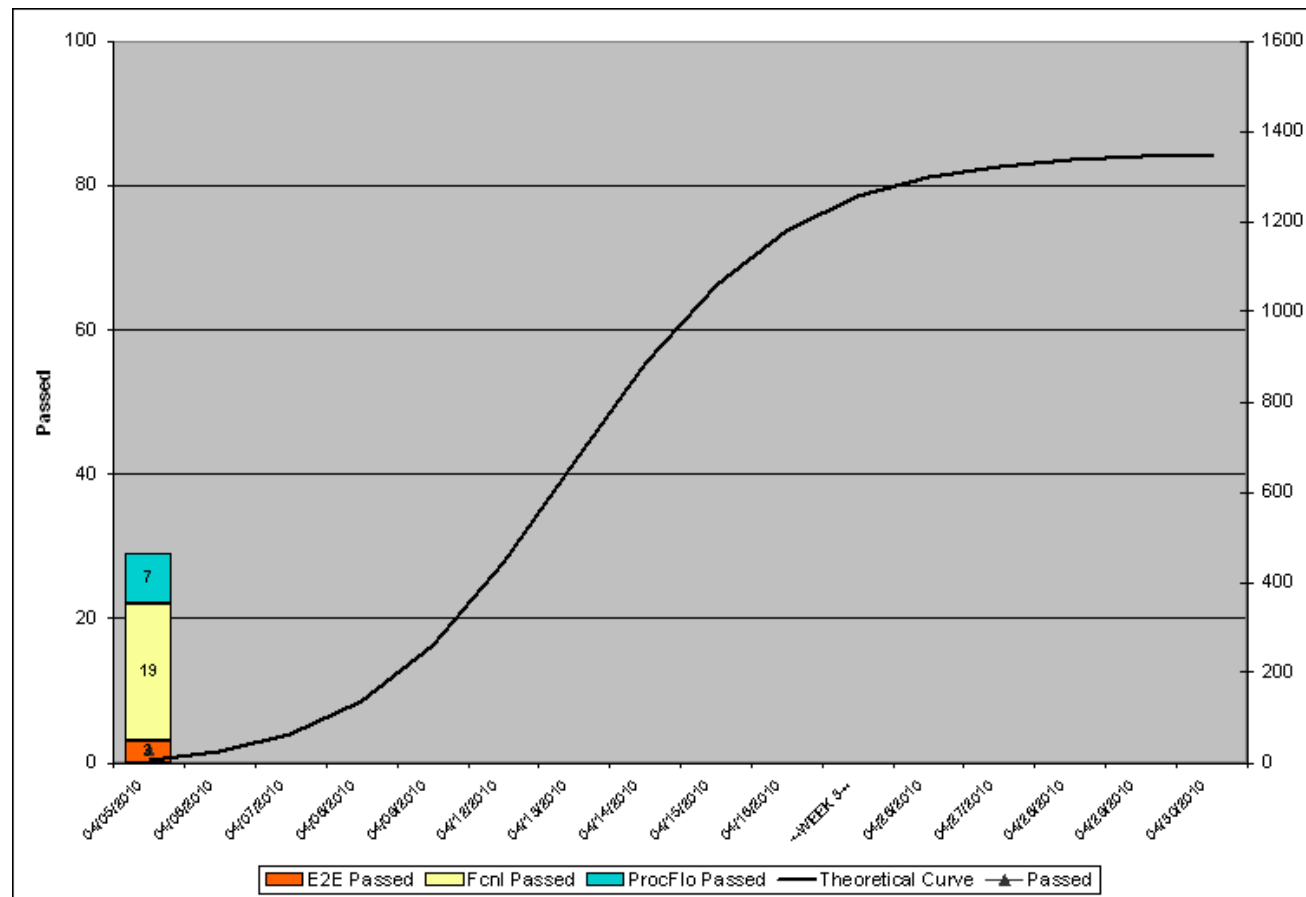
## Day 3

Pass rate proves higher than anticipated by the S-curve, but is explained by it being a regression test. Few process flows have been completed, but this was anticipated because the accounting group is finalizing end-of-quarter results.
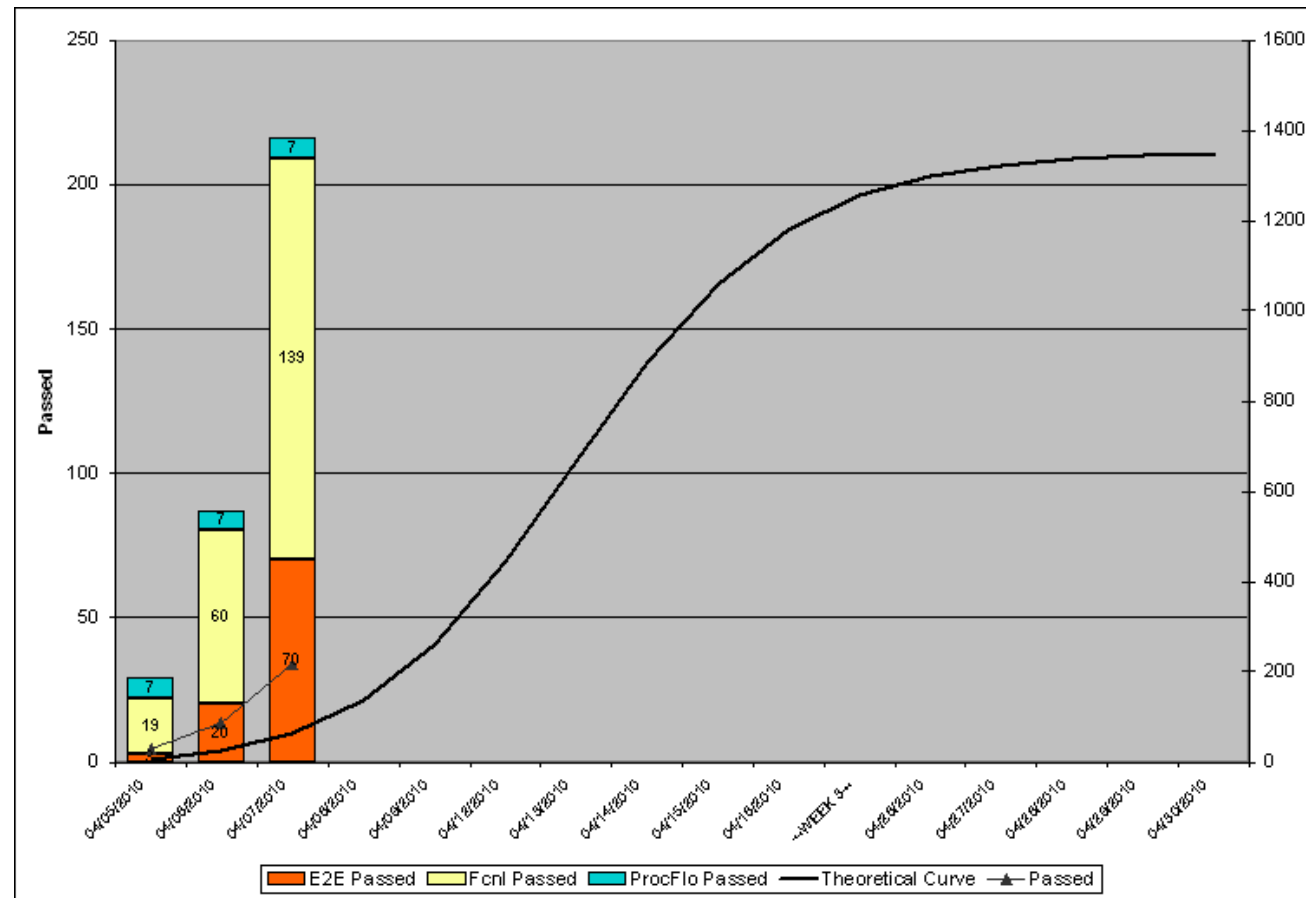
## Day 5

Pass rate at this point indicates a possible early completion. Process flow execution and passes have been high since the accounting group shifted resources to test execution.  The best news is that the team is about 2 days ahead of schedule. Manager verifies with team that they are execut-ing high priority tests first.
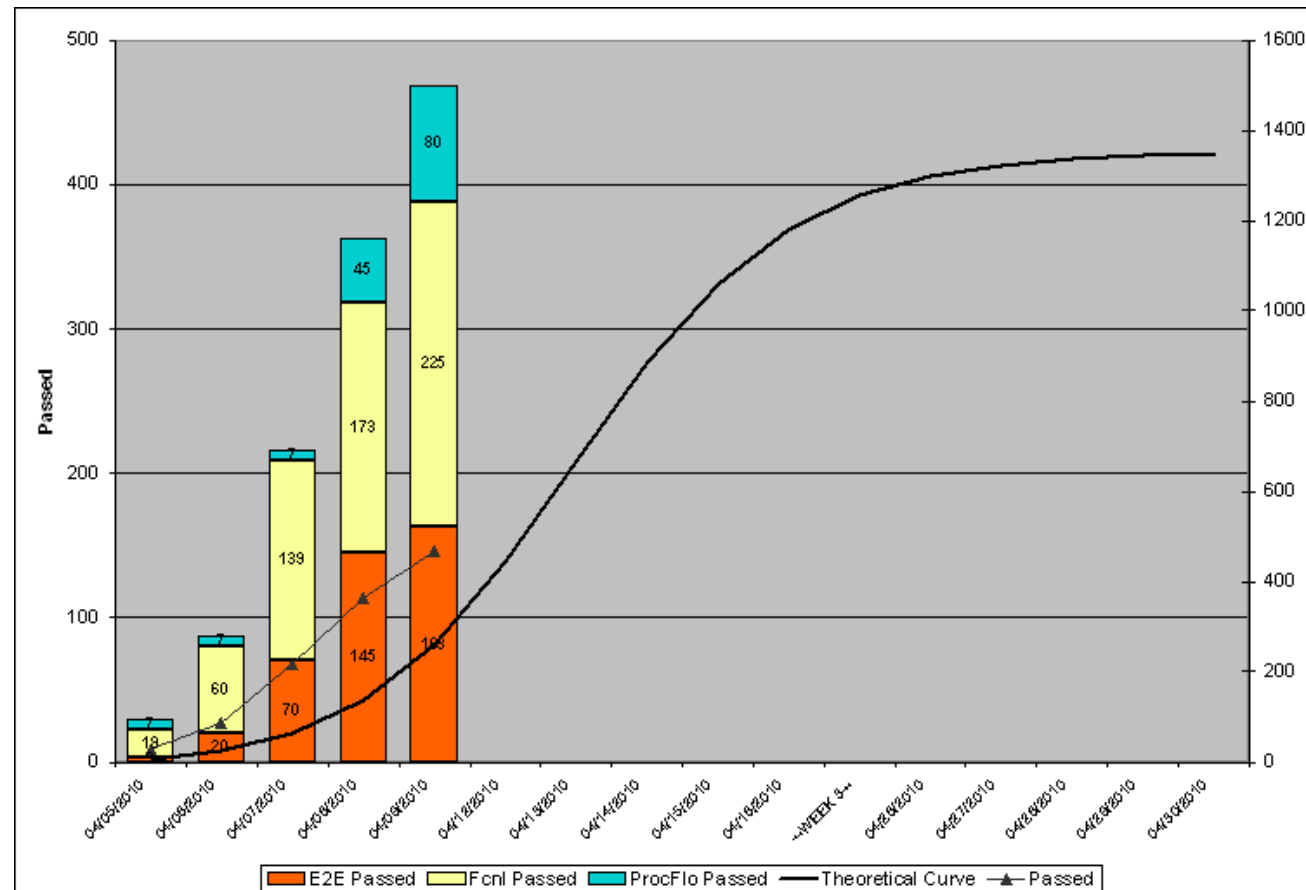
## Day 7

The team is unable to keep up with the steep rise in test case passes indicated by the theoretical curve.  An issue with custom manufacturing has been discovered and is being investigated.  This issue impacts the team's ability to execute about 30 functional and 50 E2E tests. Redirected team to focus on process flows.
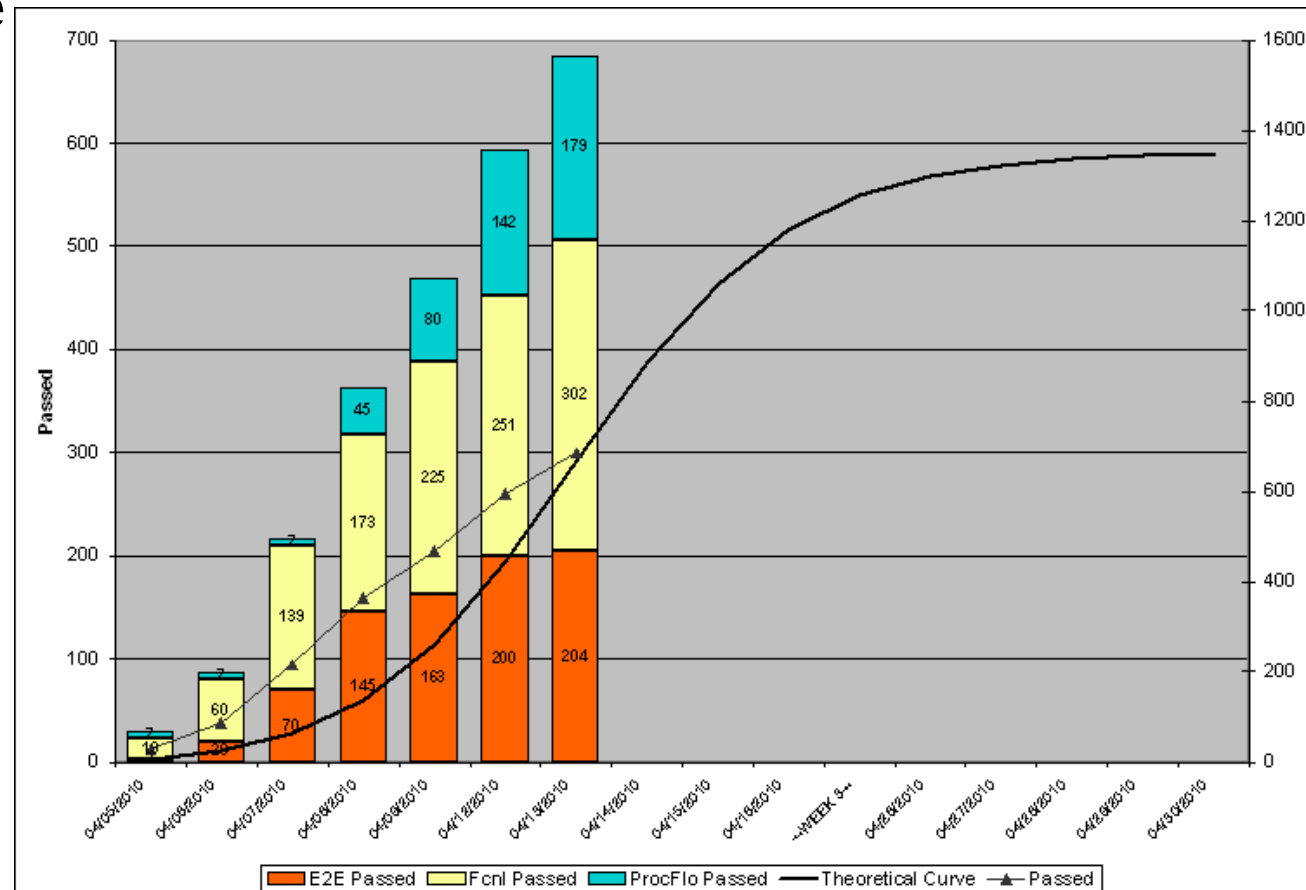
## Day 10

Execution has become anemic. Accounting is not completing process flow validations because of issues with taxes and a known problem with VAT. Next week is focused on defect correction and re-testing. Development indicates the custom manufacturing issue will be corrected by the end of the week and VAT should be corrected as well.
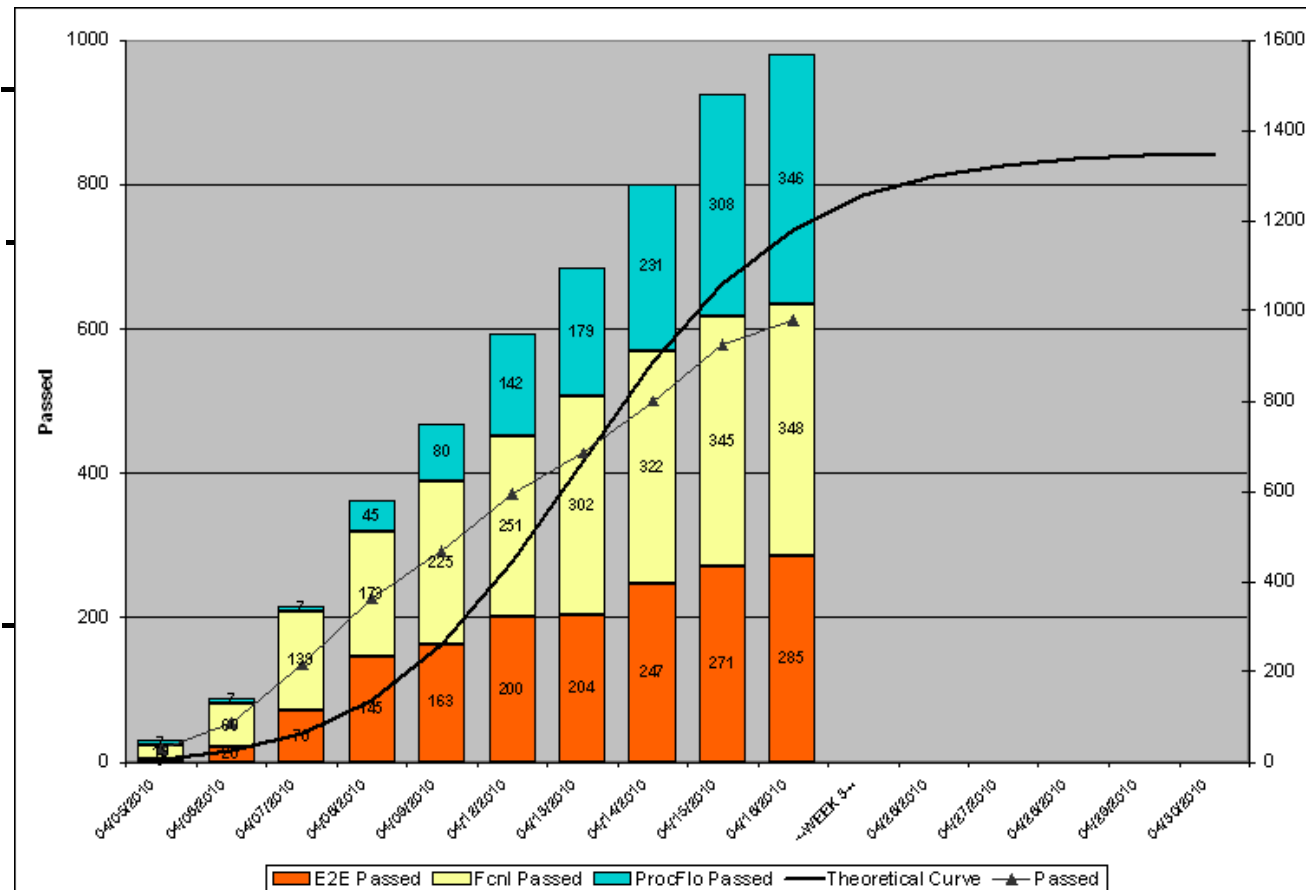
## Day 11 (after 1 week dedicated to defect corrections)

A solution was determined for the custom manufacturing issue, but implementation is not complete.  VAT remains a problem also.  The team has executed nearly all possible tests not related to known issues.  Asked team to "queue up" data for tests that must be executed once solutions are implemented.
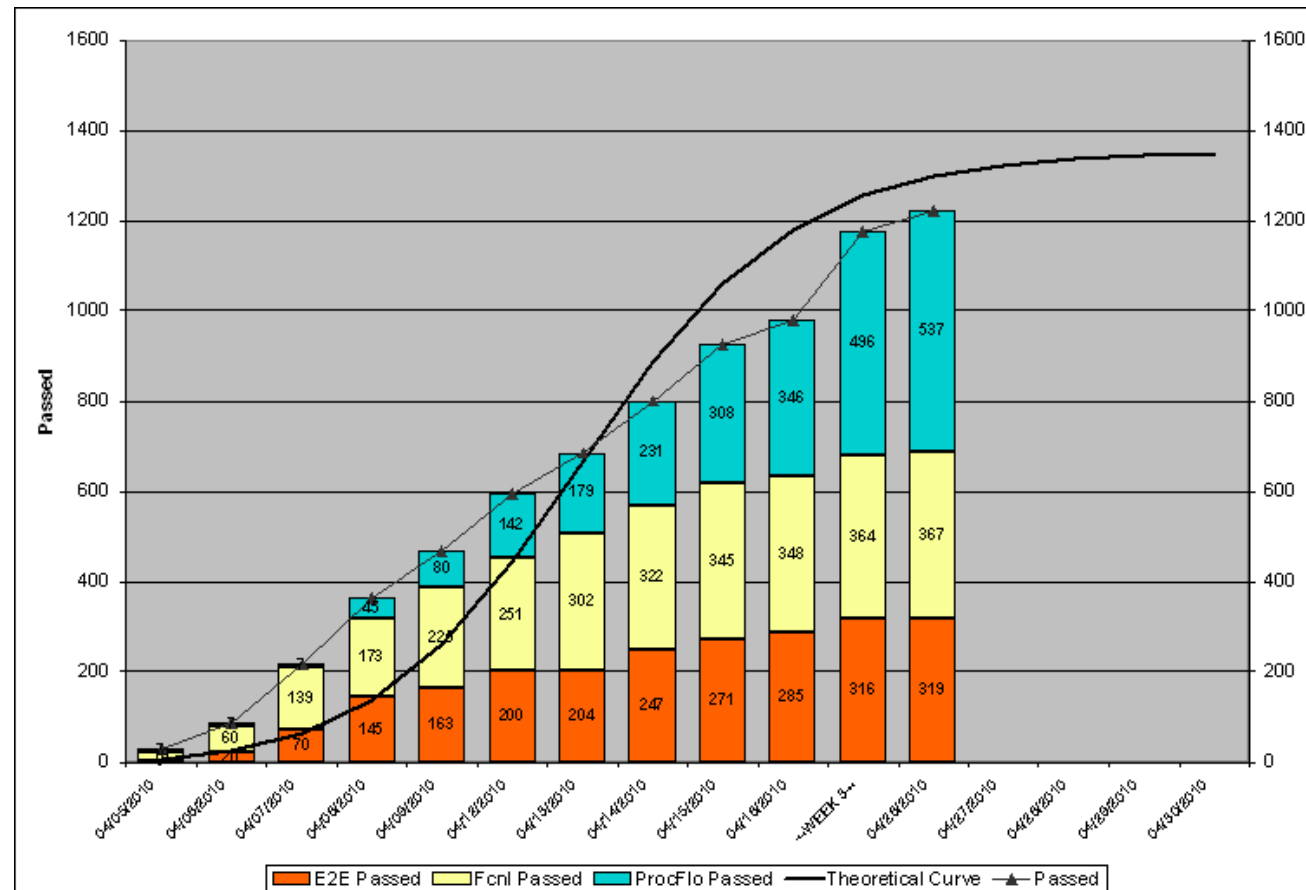
## Day 15

Although the team did not achieve 100% completion, fewer than 40 tests were outstanding.  Furthermore, specific acceptance criteria were established prior to test execution and all criteria were met or exceeded.
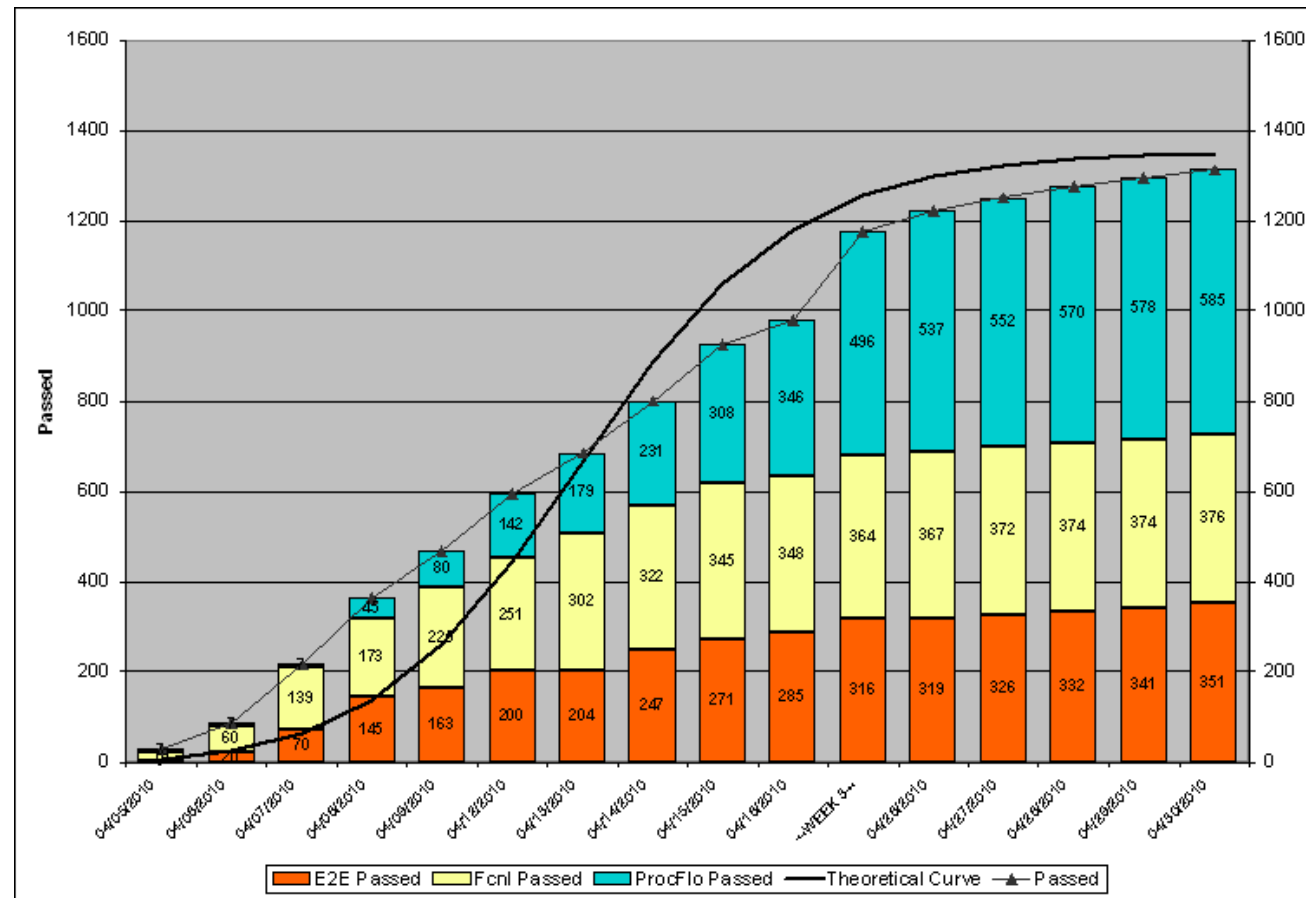
- Using the S-Curve allows the test manager to make necessary adjustments to the test effort based on a realistic projection of how the test activities should go

- I have used this graph successfully manage test efforts over the last 10 years and it is one that I highly recommend implementing

*Software Peace of Mind in Action*

- Basili, Victor, Gianluigi Caldiera, and H. Dieter Rombach. The Goal Question Metric Approach. 1994. 27 July 2009. <ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf>

- "Goal-Question-Metric (GQM) Approach" *Software Tech News*. Vol. 12 No. 1. April 2009. 27 July 2009. <http://goldpractices.com/practices/gqm>.