Quality Metrics Through Continuous Integration/Continuous Delivery

June, 2019

A story of transition…and measurement…
Executive Summary

• Continuously Integrating source code and Continuously Testing is a **culture** and a **journey**

• A major part of this journey is **measurable** success!

• The journey begins with **common philosophies** to deliver **business value** across the organization that **everyone believes in**…
  • Product/quality engineers, business management, program/project/product management…
  • Philosophic examples…
    • Enable engineering and confidence through continuous build, deliver and test
    • Build Business confidence by increasing traceability through transparency
    • Work towards having reliable and stable baseline of source code

• Those philosophies **shape** operational vision, strategic goals and processes…
  • Identify that software development is a manufacturing process
  • Every commit must have associated testing
  • Measurement of success includes **code coverage**, mean-time-of-delivery, commits to defects, etc
Continuous Integration and Continues Testing with an Embedded System

• 25+ year old, niche embedded product in all aspects of industry, focused on enterprise customers in…
  • Finance (financial institutions, banks, etc)
  • Medical (healthcare management)
  • Industrial (i.e. airlines, processing systems)
  • Education (colleges)

• This organization was one of the first movers in this space and time to market deliverables were years

• Product culture was engineering focused and primarily manual due to complexity of the product…
  • Manual testing
  • Manual integration
  • Manual build/release

• Quality Assurance was responsible for quality, NOT ENGINEERING
A Business Case: Continued

• Fast forward to today…
  • no longer a niche market – commoditized market
  • high productivity expectations
  • faster deliverable expectations
  • globalization pressures
  • less available people resources
  • larger market competition
  • quality and time to market are key elements for success

• We have begun to transition this organization to continuous integration and continuous testing...how do you measure success?
Building a **Measurable “shift-left” culture**

- Start with building a **Shift Left** or **Testing Early and Often** as a cultural philosophy.
- Measurement and evaluation starts at the beginning of the software development life-cycle!
- Focus on the measurements of software automation, integration and testing!
  - Create quantitative, measurable success criteria!
  - Identify objective metrics of success (i.e. time to delivery, commits to defect ratios, **code coverage**
    etc)
  - Remember, software development is a manufacturing practice! The goal is to maximize profitability!
  - Identify your points of variance (incoming commits, tests and configurations)
  - Create a quality and integration strategy based on testing types (unit, functional, feature, integration and customer acceptance, etc)
  - Create a configuration management automation strategy (how and who will handle config management)
  - Tools support your processes, strategies, metrics and philosophies!
Changing the Integration and Testing Dynamic

- As organizations shift left, their critical testing occurs at all different levels of the software development life-cycle and **EVERYONE** is responsible for quality!

- Quality begins during requirements collection!

- Integration is automated, the majority of testing is automated and objective measurements are set to promote code forward automatically.

**DevOps Platform**

## COPPER
- Doc Review & Approvals
- Functional, Design, Test Plan, Test Reqs

## BRONZE
- Automated Pre-commit Unit Testing
  - Off Array
  - Private branches
- Automated Post-commit Unit Testing
  - Functional Testing
  - Off/On Array
  - Release branches

## SILVER
- Automated Feature Testing
  - On Array
  - Single Build/Full Package
- Integration & Interop Testing
  - On Array
  - Full Package

## GOLD
- Manual Customer Acceptance Testing
  - On Array
  - Full Package

## PLATINUM
- Beta/Partner Testing
- On Array
- Full Package
Manual Testing in a Continuous Integration environment

• Automated testing and automated integration is key to any continuous integration strategy!
• Automated testing can focus on the core functional, feature and integration tests
• Manual testing can be focused on configuration management, and “customer acceptance” tests
• There is a place for manual testing, especially in the embedded product space!
• Product experts who are manually testing experts can test at the “customer acceptance” phases testing…
  • Normal workload usage for the product – going through normal customer acceptance tests
  • Abnormal or eccentric workload usage for the product – going through abnormal customer acceptance tests
  • Large configuration testing (large workloads simulating large customer environments)
What to measure?

• Continuous Measurement is a journey through the software development life cycle
• It is critical to create tangible, objective measurements which can be quickly assessed
  • Create quantitative, measurable success criteria!
  • Identify objective metrics of success and measure at the right time!
  • Metrics that are helped measure success…
    • design proofs – proving our design is working through specific design testing
    • time to delivery – How long does it take to deliver a fully qualified package?
    • commits to defect ratios – How many defects are we opening per set of commits?
    • code coverage – When testing, what code paths are we hitting?
    • cycle time of a commit – how long does it take to fully test a single commit?
    • pass/fail percentages per feature
Putting it all together...

Define WHAT we will code & test by documenting reqts, functional design, test plans/cases.

COPPER

Write small piece of code

Write automated tests for that code

Run all against a small config

More code!

Automated tests for aggregate code

More configs!

Customer-focused configs

MEASURE!
- Design reviews
- Documentation proofs
- Test case review
- Hardware usage
-...

MEASURE!
- Automated Test Pass %
- Code Coverage
- Cyclomatic complexities
- Static code analysis
- Hardware usage
-...

BRONZE

SILVER

GOLD

• Automated Test Pass %
• Code Coverage
• Static code analysis
• Hardware usage