A method to establish concurrent Rapid Development Cycle and High Quality in a Storage Array System Environment

Dr. M. K. Jibbe  
Director of Quality Architect Team  
NetApp APG  
mahmoudj@netapp.com

Kuok Hoe Tan,  
MTS QA,  
NetApp APG  
kuokhoe.tan@netapp.com
Agenda

- Early validation from early development to full system validation
  - Advance Dev, POC and Prototype
- Focused targeted outcome for each phase of the release process
  - New content development and integration
  - Whole product integration (new and existing content validation)
  - Solution targeted validation
  - Key customer product use case validation
- Key release metrics to track progress to key outcome for release phases
  - Train metrics, Engineering Quality Index (EQI) and Quality Dashboard
- Internalized industry standard best practices and methodologies to form the framework that serves as the foundation to drive continuous improvements for the release process
  - Agile
    - Improvement Framework (Information flow, Business Process Improvement, RCCA and Retrospectives)
- Outcome metrics – Defects Overlay
- Summary and Conclusions
Early Validation

- Advance Development
  - Research into new technology to identify new promising areas.

- Proof of Concepts
  - Develop proof of concept solutions built on NetApp’s technology that solves critical customer problems.

- Prototype / Spikes
  - Utilize early prototyping to explore viable approaches
  - Identify limitations
  - Data gathering for better work estimation
Targeted Outcomes For Each Development Phase

<table>
<thead>
<tr>
<th>Deliverables</th>
<th>Development</th>
<th>Integration</th>
<th>Stabilization</th>
<th>Regression</th>
<th>Pre/Post GA</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope</strong></td>
<td>New Content</td>
<td>Initial integration of content</td>
<td>Stabilize system solutions</td>
<td>Validate Customer Use Cases</td>
<td>Customer Support</td>
<td>Customer Support</td>
</tr>
<tr>
<td><strong>Schedule</strong></td>
<td>Schedule Integrity</td>
<td>Schedule Integrity</td>
<td>Schedule Integrity</td>
<td>Schedule Integrity</td>
<td>OEM Release</td>
<td>Cadence and On Demand Release</td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td>Stable New Content</td>
<td>Stable Integrated New &amp; Old Content</td>
<td>Stable Branch</td>
<td>Stable Shippable Product</td>
<td>Stable Release</td>
<td>Stable release</td>
</tr>
</tbody>
</table>

- Focus on end deliverable of the entire release
  - Goal is to optimize the release, not each phase (may choose to sub optimize phases)
  - Drives decision making throughout the release

- Each phase provides a building block to support the next phase
  - Drives alignment
  - Move the right work to the earlier phases (as much as possible)
Metrics

- Internal Development Phase Metrics
  - Balanced
  - Covers scope, schedule, budget and quality

- EQI (Engineering Quality Index)

- Quality Dashboard
  - Field issues
  - Critical issues
  - Incident reports
  - On time shipment report
  - Miss-ships report
  - Auto Support
Internal Development Phase Metrics

- **Scope**
  - Estimated work vs actuals
  - Planned content vs delivered

- **Schedule**
  - Execute to plan metrics (start, complete, major milestones)

- **Budget**
  - Per period (quarterly) spend to plan

- **Quality**
  - Defect discovery timeline
Release Metrics

- Release Metrics to track the progress of a Train
  - Engineering Quality Index (EQI)
    - Release EQI
    - Phase EQI
      - Boxcar, EIT and Regression
  - Defect Trending Charts
  - Train Projection Charts
  - Build Quality Checker Board Charts
  - Configurations Utilization
The chart represents the release EQI model

- At EIT Exit EQI should be >=80
- At Regression Exit EQI should be >=95
EQI Components

- **Story Points / Test Cases (TC) Completion**
  - Measures the Feature integration using the Completed and the planned number of Story Points/Test Cases.

- **Defect Quality Impact (QI)**
  - Customer Impact of the unresolved Defects are measured. This includes all the deferral and restrictions.
  - Defect QI varies from 0-10, where 10 is the highest customer impact.

- **Incoming Defect Rate**
  - This component measures the Defect Incoming rate normalized by the Test Run execution. This implies the stability of the release.

- **Code Coverage**
  - Code Coverage measures the percentage of the source code validated during the module testing. Currently Code Coverage threshold is set at 75%.

- **Code Change**
  - Number of Lines of Code (LOC) Changes since last check point.
Incoming defect counts are overlaid on the phase EQI charts
- Defect Counts are shown by the secondary axis
Projection Metrics

- Weather chart represents the defect projection of a train.
- Test cases execution of a train can be projected using the second chart illustrated above.
Sample Schedule Metric

Independent Scrum Teams
The Journey to Agile

The Problem
Changing requirements are difficult to digest (impacts plan)
Late & Complex defects cause delays in product releases
Makes the Release cadence unpredictable for customers

The Vision
- Regular, repeatable and sustainable release cadence
- High-Level requirements are well-known and communicated
- But details will be learned along the way
- Adapt to new environments and evolving requirements
Traditional Waterfall Characteristics

- Requirements are known and fixed at beginning of project
- The assumption is that the more time spent up front making sure requirements and design are correct will save you much time and effort later
- Emphasis on comprehensive documentation early in process

“Big Design Up Front (BDUF)”
New product functionality is identified by PRs and ERs which are assigned to Scrum Teams by the PDT.

Scrum Teams are responsible for implementing working functionality.

Scrum Teams deliver working functionality in Sprints time-bound to 4-weeks or less. Sprints should be the same duration through the scrum development cycle.

Tested Functionality grows in complexity over time.

The Scrum Iteration Plan is defined by a sequential series of Sprints, leading to fully implemented and tested functionality ready for a Release Train.
Improvement Framework

<table>
<thead>
<tr>
<th></th>
<th>Intra Phase Feedback Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Intra Train Feed Forward Loop</td>
</tr>
<tr>
<td>3</td>
<td>Common Cause Inter Train Feed Forward Loop</td>
</tr>
<tr>
<td>4</td>
<td>Special Cause</td>
</tr>
<tr>
<td>5</td>
<td>Special Cause Inter Train Feed Forward Loop</td>
</tr>
</tbody>
</table>

Boxcar EIT  Stability  Acceptance Regression  OEM Pre & Post GA  Field  

Future Train Feedback Loop
Parallel Improvement Frameworks

- Boxcar EIT
- Stability
- Acceptance Regression
- OEM Pre & Post GA
- Field

Special Cause
Intra Train Feed forward Loop
Full Train Feedback Loop
Process Improvement

Planning
Boxcar EIT
Stability
Acceptance Regression
OEM Pre & Post GA
Field

Inter Train Feed forward Loop

Planning
Boxcar EIT
Stability
The Results!

- **Shifting Left**
  - Defects found earlier in parallel development releases
  - Real time information feed forward to parallel development teams
  - Total defects found relative to the size of each release
Summary and Conclusions

- Alignment of development and QA during the development and certification process
- Work smarter and more efficient
  - Allow us to reduce redundancies
  - Accelerate information flow
  - Allow us to reduce redundancies
- EQI provides more confidence during phase transitions
- Achieved regular, repeatable and sustainable release cadence
- Shift Left result confirms the benefit of this approach for concurrent development of a high quality product
Backup
Defects Shifted Left

Before Final Regression Test (Boxcar, EIT)

During Final Regression Test (Regression)

After Regression during first 18 months of Field Experience (Pre-GA, Post GA)