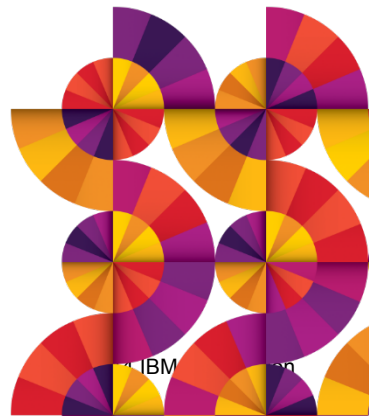


# Spectrum Scale Quality Actions

Spectrum Scale User Group  
SC15, November 2015



# Agenda

- Development transformation
- Quality Improvement Actions
- Future Quality Action Plan

# Agenda

- **Development transformation**
- Quality Improvement Actions
- Future Quality Action Plan

# Spectrum Scale Development/Test Transformation Activities

## 1. Improved Operational Efficiency & Quality

- Introduced causal analysis APARs – APARs go thru causal analysis process (since mid 2013) with dev, test, service involved
- Introduced OPC (Orthogonal Problem Categorization) process to help categorize defect to understand / attack problem areas
- Adopted Coverity source scanning tools to identify security risks and several classes of potential defects.

## 2. Transform Build and Test

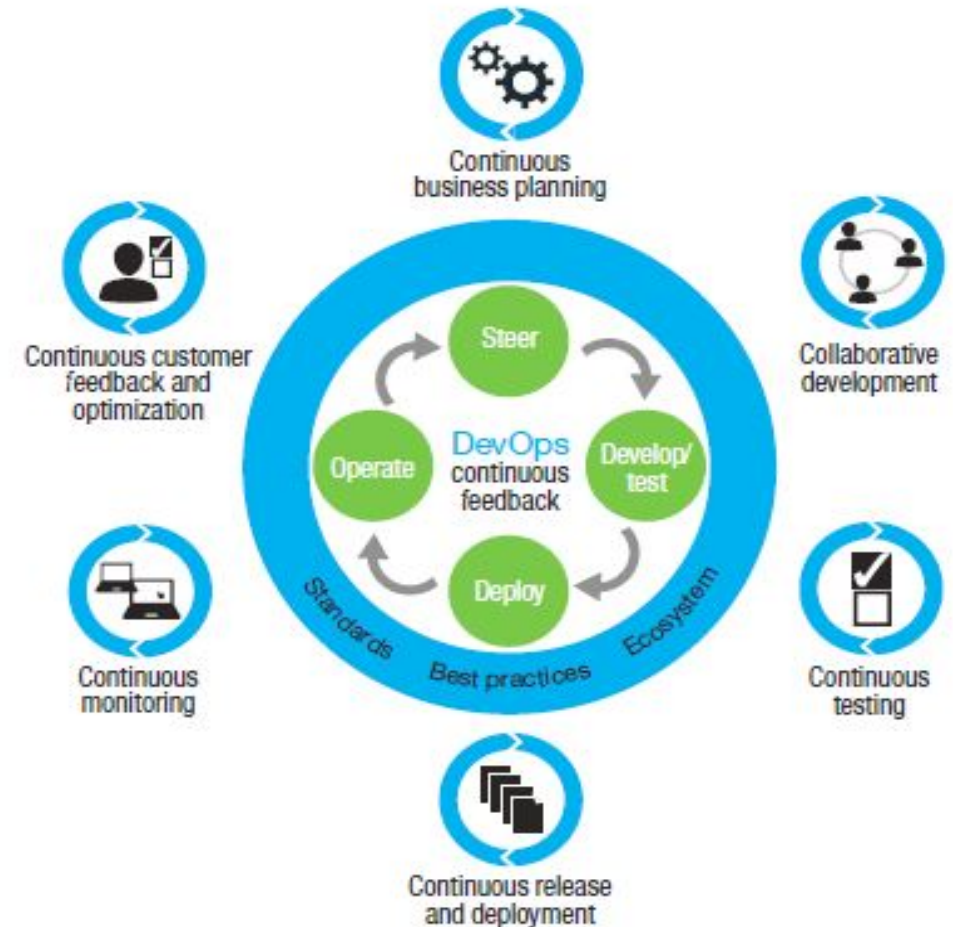
- Began a fundamental test transformation
- Changed the build process to be more automated

## 3. Enhance Customer Experience:

- Introduced Early Feedback and Engagement with Clients
- Increased early touch points with IBM consumers of Spectrum Scale

## 4. Infuse DevOps Culture for Faster Time to Value

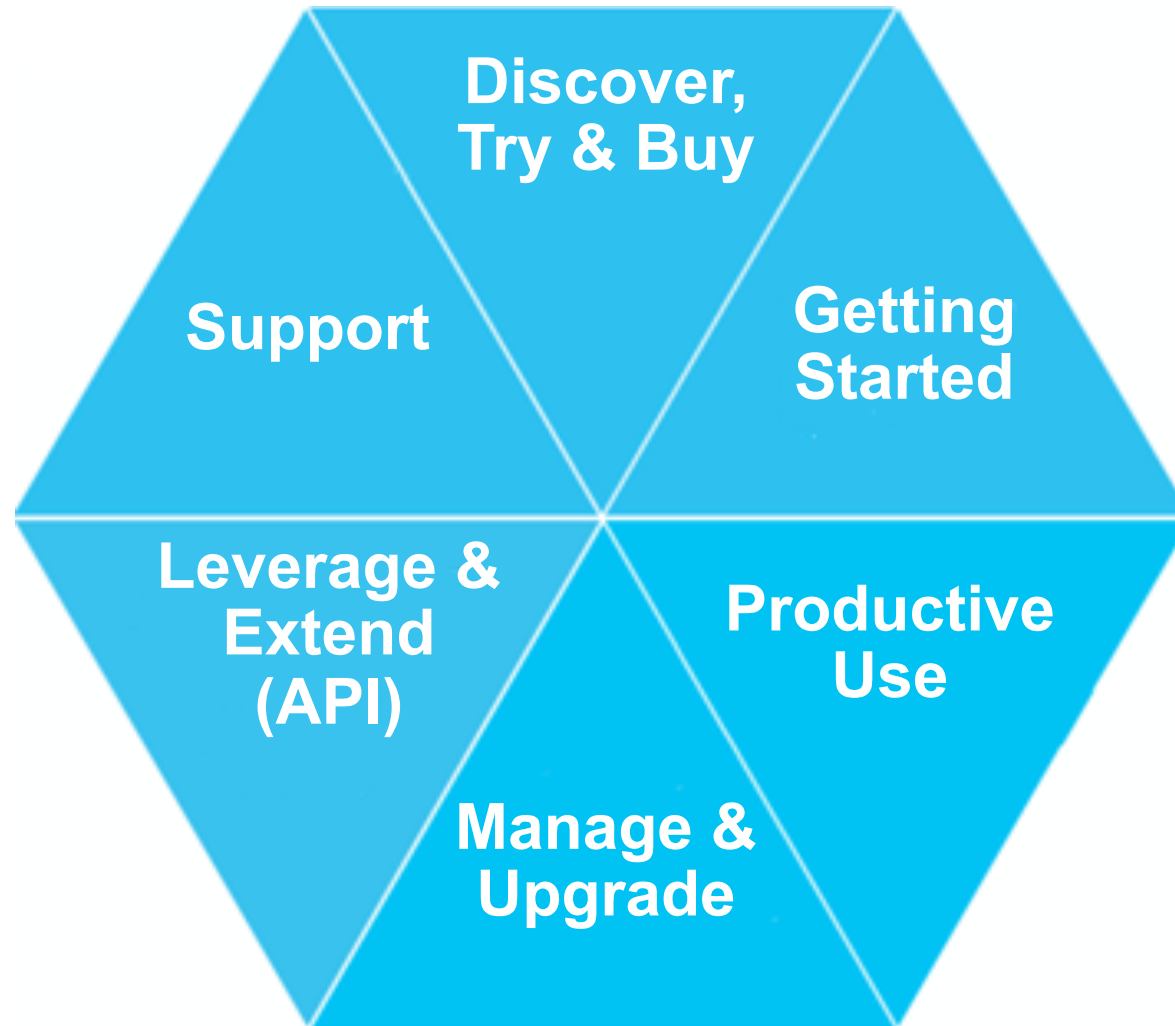
- Introduced Design Thinking, Agile, DevOps



# Agile Implementation

- Started agile with protocol team for 4.1.1 June 2015 Release
  - Created Protocol Delivery Scrum Teams to address cross protocol integration and consumability
  - Base Spectrum Scale team delivered Technical Foundation features
  - Foundation built to expand agile practice for the entire Spectrum Scale development team
- Epics / user stories
  - Minimum viable product content definition
  - Created top epics and prioritized user stories based on release hills / client need.
  - Business commitments done epic level
- Automated delivery pipeline using tools
  - Robot framework for user story test automation
  - RTC as project management tracking tool

# The 6 client experiences Informed Protocol Themes / Epics / Stories

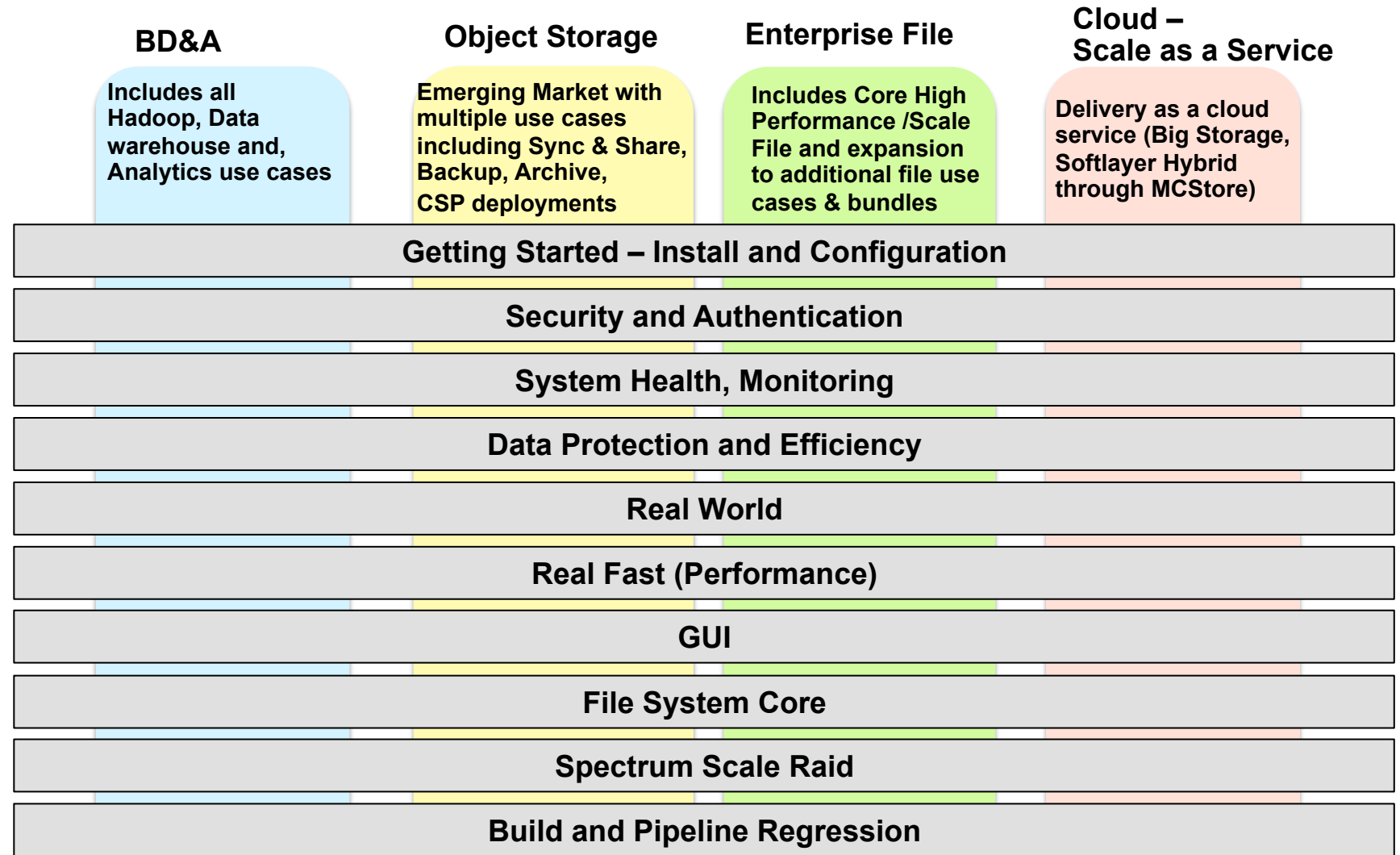


## Spectrum Scale Protocol Scrum Themes:

- Build
- Pipeline Regression
- Getting Started
- Easy Configuration
- Security and Authentication
- System Health
- Data Protection
- Real World
- Real Fast
- File System Core

# 4.2 Release Scrum Structure

- Delivery Scrums (verticals) aligned to market use cases
- Delivery scrums deliver as SW, HW/SW (ESS/GSS), with partners, AND/OR on the cloud to accommodate the needs of the market
- Pervasive Scrums (horizontal) ensure consistency of a single product experience and efficiencies of shared deliverables across use cases



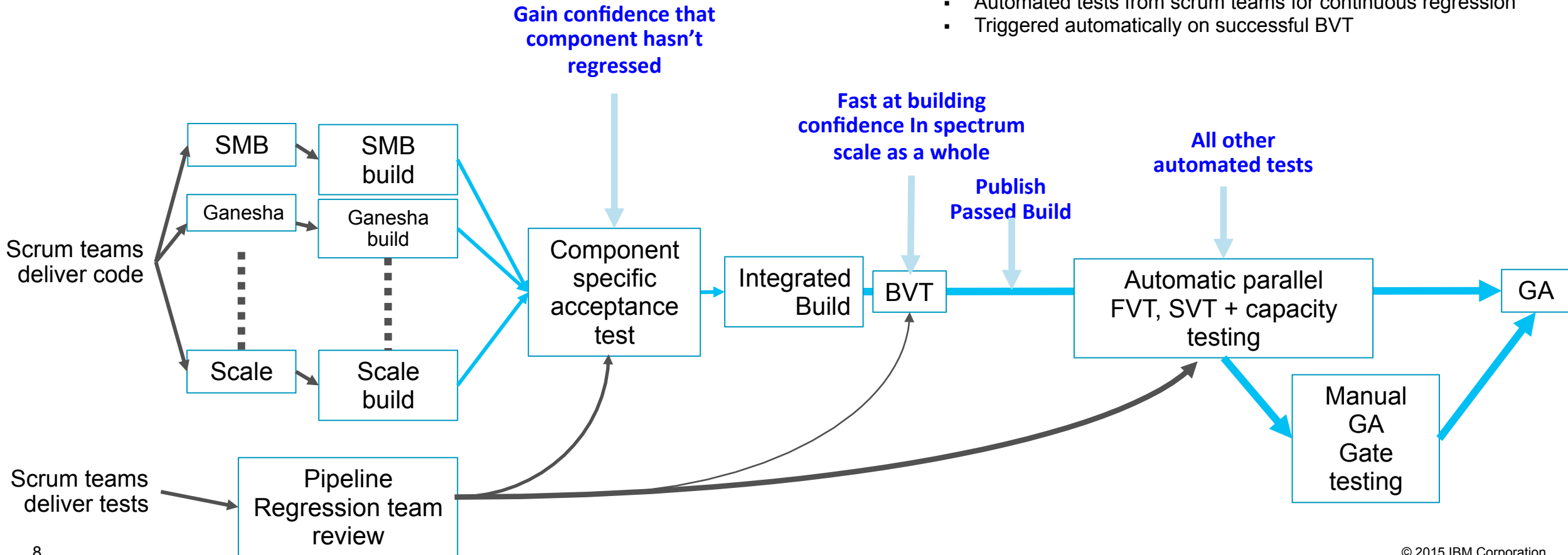
# Delivery Pipeline Improvement for Protocols

## Continuous delivery

- Build & packaging, component acceptance test & BVT, all automated
- BVT covers basic functionality testing for general consumption for the internal dev/test teams
- Infrastructure implemented for continuous delivery on code commits, BVT <2 hours

## Pipeline Regression scrum

- Automated tests from scrum teams for continuous regression
- Triggered automatically on successful BVT





# Agenda

- Development transformation
- **Quality Improvement Actions**
- Future Quality Action Plan

# Top Client Pain Points

Top 5 Customer Pain Points	Source of Information	Solution / Mitigation
1. Complexity of installation / configuration: often requires IBM assistance, many configuration options	Customers	Improved Install-toolkit, GUI Install Wizard, install / configuration checklists
2. Lack of monitoring	Customers	GUI Monitoring. Spectrum Control will also provide monitoring. Gap still remains and will be a top priority for 2016 deliverables.
3. Lack of diagnostics for FTDC and data required to debug tends to be large in size	Customers / L2	We have improved performance impact of traces but tracing is still not on by default. 4.2.1 will continue to focus on problem determination.
4. Complexity of the product – many parameters / tuning options	Customers / Sales	GUI being added to simplify the usage and guide customers to what selections to make. Plan to continue to improve.
5. Pre-sales: lack of ability to accurately model system requirements for given workloads	Studio Designer Interviews	4.1.1 provided information for protocols via sizing guide. Pre-sales material for object and modeling potential customer configurations will be provided on an on-going basis.

# High Risk Components

Top High Risk Components	Source of Information	Solution / Mitigation
1. Performance type of questions and issues is a high driver of our PMRs, with Q&A growing YTY	PMR Analysis	Adding monitoring and zimon data collection which will help to address some of the performance PMRs. Work will continue post 4.2 as well.
2. Non Spectrum Scale issues such as Network, HW, OS, and other SW drive about 40-45% of our PMR calls	PMR Analysis	GUI Monitoring in 4.2 should help improve problem determination. Network Monitoring tool and Problem Determination is 2016 focus.
3. Hangs and deadlocks – although improving, still a heavy hitter for PMRs	PMR Analysis	Each release continues to provide enhancements to reduce the impact of deadlocks / hangs.
4. Closure on secure engineering requirements	Security Assessment	Focus on vulnerability testing, threat modeling, secure coding and source code scanning, using Coverity & AppScan.

# Quality Improvement Actions

Key Lessons Learned from 4.1.1	Process Improvements
Iteration Exit at start of GM <u>Regression</u> did not leave enough time for integrated, high stress, long run type of testing	Added a 4 week hardening test phase between iteration exit and GM regression.
Late <u>system level testing</u> by agile team. Team added a virtual system test scrum team late in cycle and this testing extended into GM regression.	Virtual system test scrum team in place to ensure system test is properly covered. The hardening test phase was also added to address this issue.
Still followed traditional <u>beta</u> where we send code. Need to expand this into doing more mockups, demos, etc to obtain earlier and more frequent customer feedback.	In 4.2, we have scheduled multiple playbacks where the team can provide mockups, demos, etc for early / ongoing sponsor user feedback. GUI provided a alpha, open beta and regular hosted beta.
<u>Documentation</u> came late in the process vs included with each iteration. Resulted in late delivery and rush to get the work completed. Was not available for beta start.	Incorporate documentation into the user stories for each iteration. Improvements still needed in this area for future releases for e2e documentation coverage.
Need to monitor more than open <u>defects</u> to avoid a large inventory of returned / verify defects late in the cycle	Agile process includes addressing the defects in the iteration including defect verify. This is an area the team needs to continue to focus on for we still have a high percentage of defects in verify at the end of the cycle
Although the protocol work made great progress in <u>test automation</u> , the focus was on protocols vs entire product.	Expanding automation test coverage across the product. Pipeline Regression scrum team to cover the full product in the future.

# Agenda

- Development transformation
- Quality Improvement Actions
- **Future Quality Action Plan**

## Better Problem Determination

- **Problem Determination**

- Automatically detect problems in the Spectrum Scale System
- Point customers to the “root cause” of a problem
- Hide system complexity, easy representation of problems

- **Problem Isolation**

- Collect all required information to analyze problems
- Provide tools to help customers to isolate problems
- Provide metrics for the customer to understand system load

- **Problem Recovery**

- Automatically repair problems where possible
- Guide customers through the problem fixing process
- Ensure System availability through failover/recovery

# Problem Determination Areas for Improvement

- **Functional problems of spectrum scale components**
  - File system corruption / NFS down
  - Present well defined States & Events
  - Failover, restart
- **Performance problems in Spectrum Scale**
  - Metadata IO is degraded by 10%
  - Collect and analyze system metrics (Bottleneck analysis)
  - Auto tuning for a given workload pattern
- **Client Access problems in Spectrum Scale**
  - Problem accessing to a file
  - Better error msg
  - Need to guide users to find the root cause (ACLs, Auth, mapping,..)
- **Customer Environment Integration problems (during setup/install)**
  - Wrong authentication servers setup or Network setup complexity
  - Setup fails or is partially successful
  - Tools to query the customer environment or to test network setup

# Problem Determination Design Thinking Hills

- Hill1: Bob, a Spectrum Scale administrator can understand the health of all his cluster components from a single place.
  - Determine state for all system components and show in CLI + GUI
  - Thresholds for performance / Detect miss configurations
  - Notifications
  - Catastrophic performance collapses
  - Understand our customers (Call home + Inventory + Analytics)
- Hill 2: Bob, a Spectrum Scale administrator can get guidance or automated solutions to problems without contacting IBM support.
  - Guide customer through the fix procedure
- Hill 3: Bob, a Spectrum Scale administrator can identify issues with his operating environment that could cause problems from a single tool.
  - Clearly distinguish between customer environment & system problems
  - Guide the customer in setting up network
  - Find performance issues in customer environment