Software Testing and Quality Assurance in Company with Distributed and Multidimensional Teams: How to Improve?

Emma Danielyan

CQG

2013 CEE-SEC(R)
Software Engineering
Conference in Russia

Agenda

CQG's

- Software Testing and Quality Assurance Process
- Unified Testing Process
- Unified Test Plan
- Metrics
- Benefits

CQG

Celebrating 30 Years

Innovating for thirty years... and counting.

CQG Geography

SQA Department Structure

Distributed and Multidimensional

- Distributed by location & time zone
- Distributed by development systems
- Experts in the team
- Each functional area has at least 2 experts
- Traders team as a testing team
- Automated and Manual testing
- Flexible working schedule

Projects

Development projects:

- new functionality
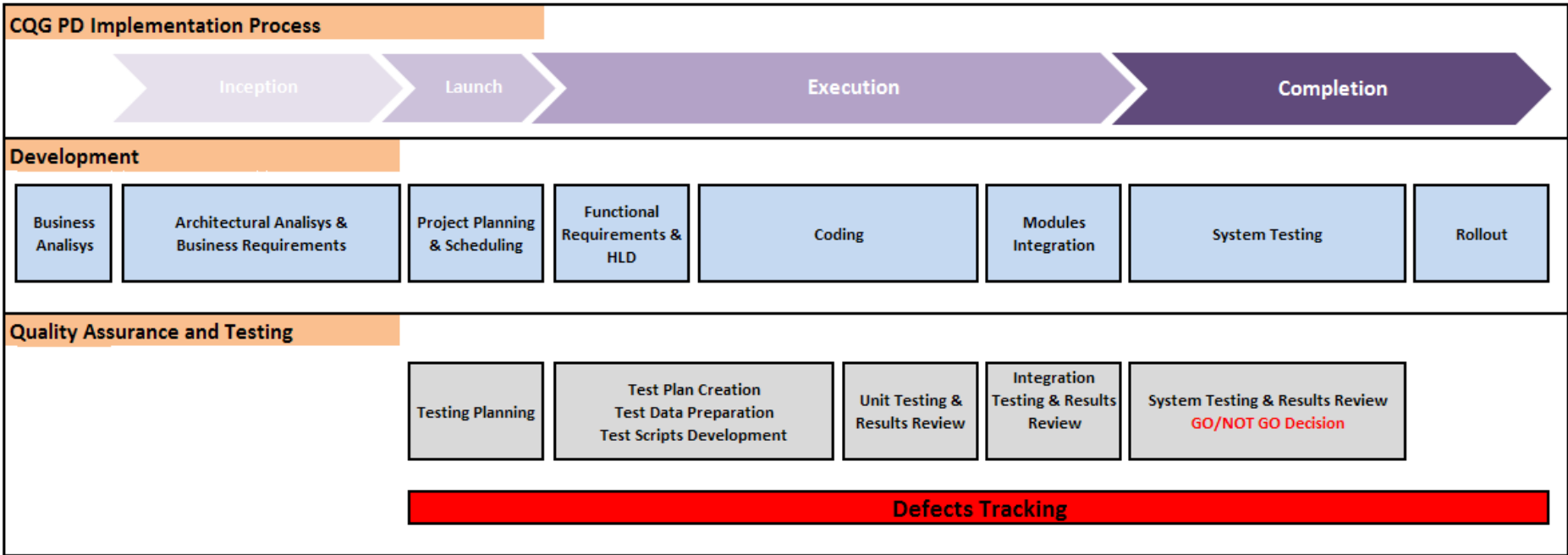- reengineering of problem
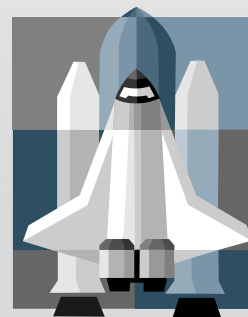
Maintenance projects:

- fix, investigate and repair
- small improvements

CQG Product Development Process

CQG PD Implementation Process

Inception → Launch → Execution → Completion

Development

Business Analisys	Architectural Analisys & Business Requirements	Project Planning & Scheduling	Functional Requirements & HLD	Coding	Modules Integration	System Testing	Rollout

Quality Assurance and Testing

Testing Planning	Test Plan Creation Test Data Preparation Test Scripts Development	Unit Testing & Results Review	Integration Testing & Results Review	System Testing & Results Review GO/NOT GO Decision

Defects Tracking
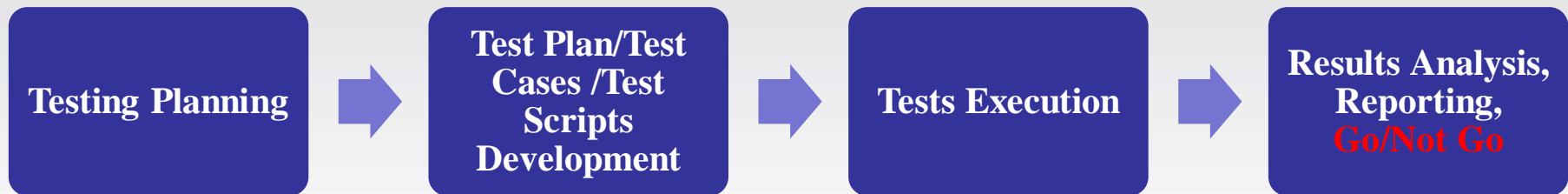
SQA Early Involvement

Results & Benefits

- SQA Team is fully educated on the project

- Better understanding the aim of the project

- Familiarity with the project Reqs and ITP

- Opportunity to prevent defects before code is written

- Gain an understanding of the complexity

- Better SQA activities planning/prioritizing

- Adequate time to prepare for the testing

- Good working relationship with the development team

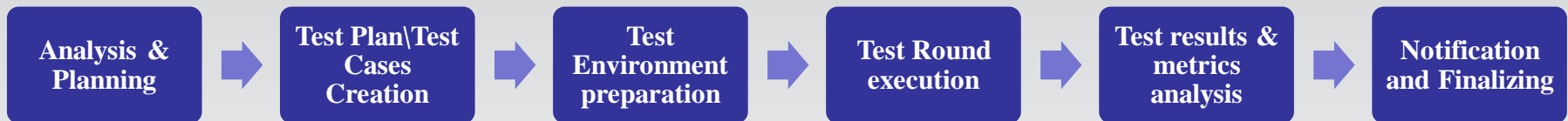- Better communication between all participants

Testing Levels

Unit Testing
- Automatic verification of individual units of source code

Integration Testing
- Manual/automatic verification of integrated units

System Testing
- Manual/automatic verification of complete, integrated system

Regression testing
- Manual/automatic verification of bugs fixes
- Manual/automatic verification that old functionality is not broken

Testing Levels

| Testing Planning | → | Test Plan/Test Cases /Test Scripts Development | → | Tests Execution | → | Results Analysis, Reporting, Go/Not Go |

Major Testing Steps in Each Level

Unified Testing Process

Goal

- To ensure and guarantee high quality of software Product, Feature, Component, Unit.

| Analysis & Planning | → | Test Plan\Test Cases Creation | → | Test Environment preparation | → | Test Round execution | → | Test results & metrics analysis | → | Notification and Finalizing |

UTP – Unified Testing Process

Results & Benefits

- We have a process that could be "instantiated" in any particular testing process in any dev system.
- UTP defines major roles and artifacts used in testing.
- UTP summarizes the most common parts and steps in testing.
- UTP is used as a template for processes in specific system or product;
- UTP provides process flexibility that is essential for multisystem projects.

Testing & Documentation

Test Documentation

- Test Strategy
- Test Plan / Test Cases
- Test Check Lists
- Test Reports
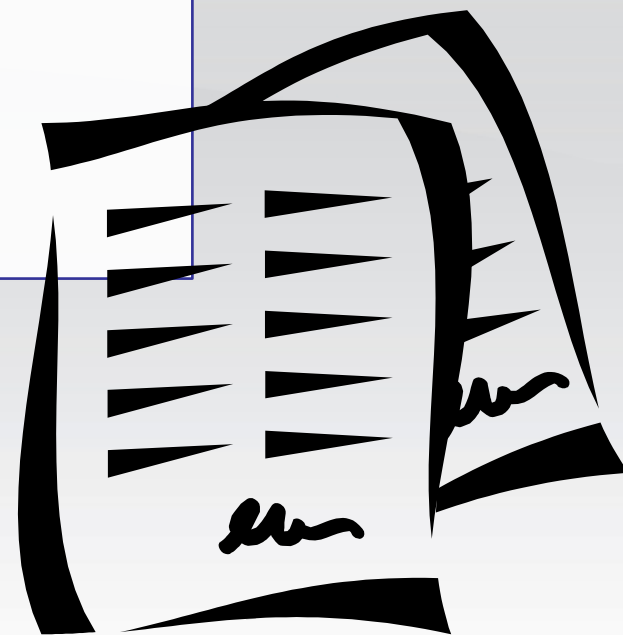
Test Plan

Test Plans by Phases

- Integration Test Plan
- System Test Plan
- Regression Test Plan

Unified Test Plan

- Everyone is aware
- Reduction of work duplication
- All tests are in one place

Updated Process

Changes

- SQA early involvement into the project
- Developers involvement into the testing process
- More automation, less manual testing
- Unified Test Plan
- Risky areas are being tested first

Benefits

- Reduction of the project completion phase
- Increased testing speed
- Increased test coverage
- Reduction of time spent on different level Test Pans creation
- Critical issues are found first

Process support via set of tools
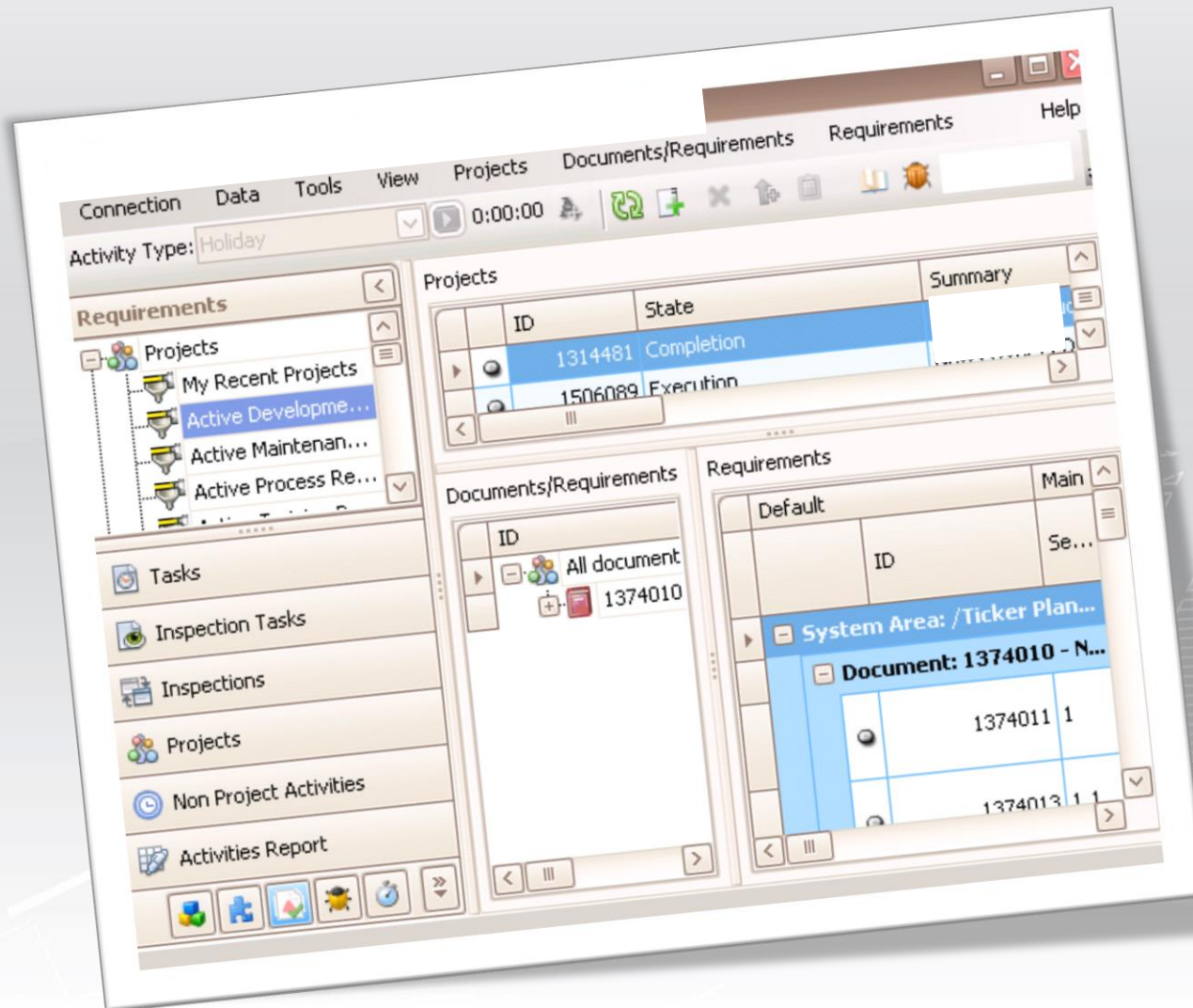
Data Visibility

Project Quality Evaluation

Methods

- Project Metrics Review
- Regular Project Checkpoints
- Project Postmortem with lessons learned session

Software Measurements

Data to measure

- **Time** (spent for different activities and phases)
- **Size** (product size produced)
- **Defects** (or issues) data
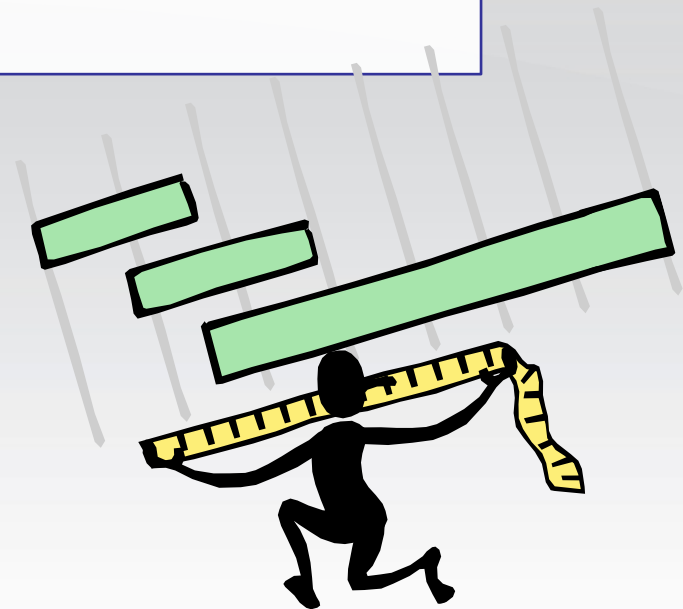
Metrics

Categories

- Consistency & compliance
- Progress
- Planning accuracy
- Productivity
- Software quality
- Process quality (e.g. inspection metrics)

Metrics (CQG specific)

Levels

- Whole PD
- System
- Development Project
- Maintenance Project
- Office
- Team
- Team - Project
- Individual

Quality Metrics

Project Level

- Integration defects density
- Actual Integration defects vs. Estimated Integration defects
- System defects density
- Actual System defects vs. Estimated System defects
- Integration defects vs. System defects
- Amount of code rework
- Total Numbers of found defects of different types should comply to formula:
- **Inspection DE # > Integration DE #> System DE #> Late DE #**

Project Activities and Defects Metrics

Activity Distribution

- Requirements, 1%
- Management, 2%
- Environment Setup, 1%
- System Testing (+ Defect Fix), 2%
- Integration (+ Defect Fix), 20%
- Inspection, 13%
- Architecture, 3%
- Test Cases Development, 1%
- Design, 5%
- Code & Unit Test, 52%

Legend:
- Requirements
- Architecture
- Test Cases Development
- Design
- Code & Unit Test
- Refactoring
- Inspection
- Integration (+ Defect Fix)
- Maintenance Defects
- My Team Defects
- SCM Non-Project Work
- SCM Project Work
- System Testing (+ Defect Fix)
- Environment Setup
- Investigation
- Management
- Documentation

Defect Metrics:	Total (#)	Density (#/KLOC)
Inspection (F) Defects:	65	2,880
Integration Defects:	50	2,216
System Defects:	5	0,222
Late Defects:	1	0,044

Metrics on Project Dashboard
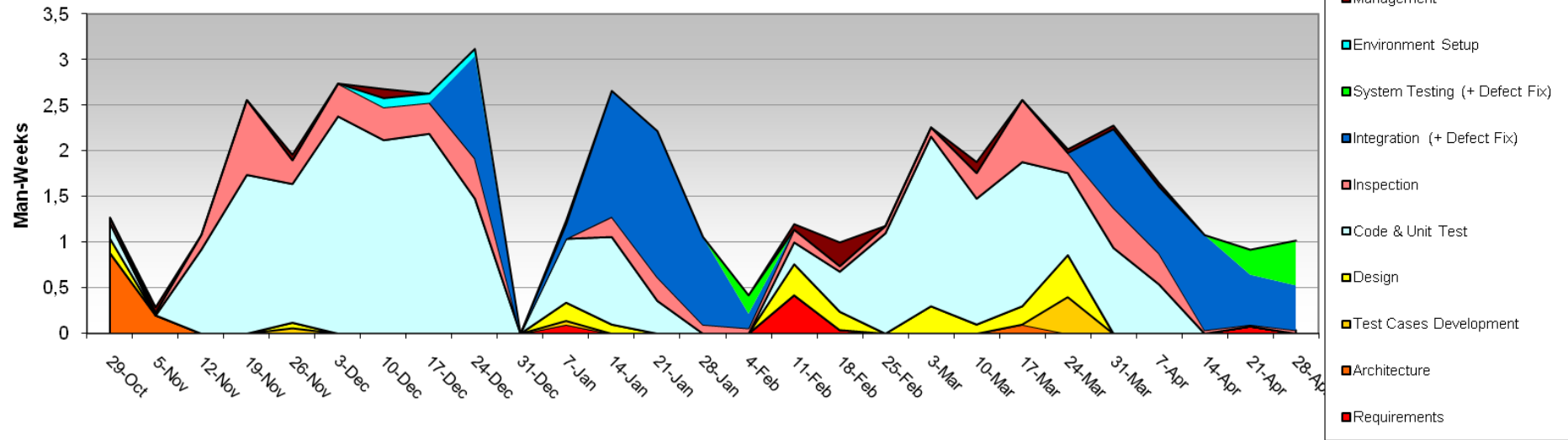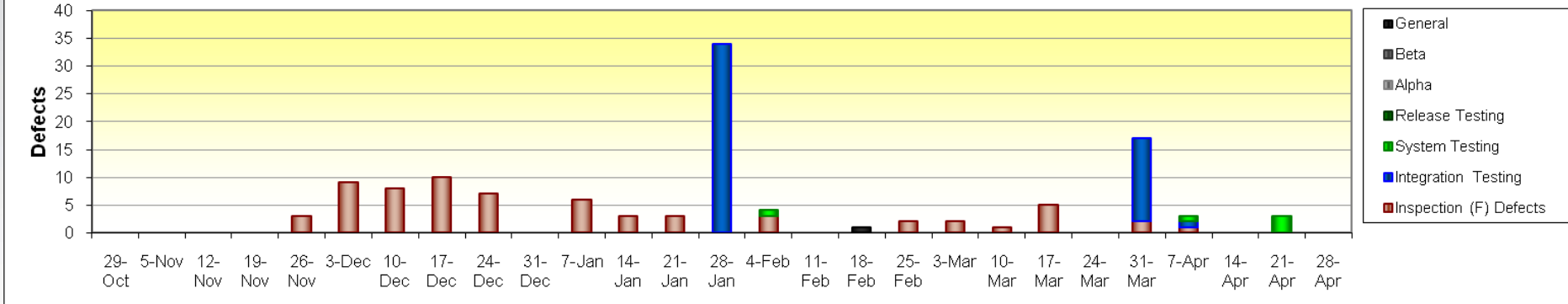
SQA Team Level Metrics

Quality Metrics

- Reported D/S/I distributed by priorities
- Total Suggestions reported and prioritized
- Initial Inquires reported
- % of relevant (P0-P3) suggestions among total number of reported for a month
- % of defects closed as "Created in Error" or "Cancelled"
- % of inquiries converted to defects/suggestions
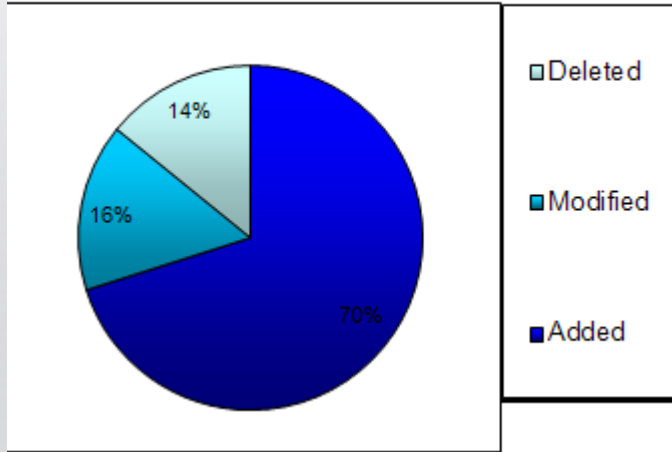- Number of Failed Testing defects and suggestion

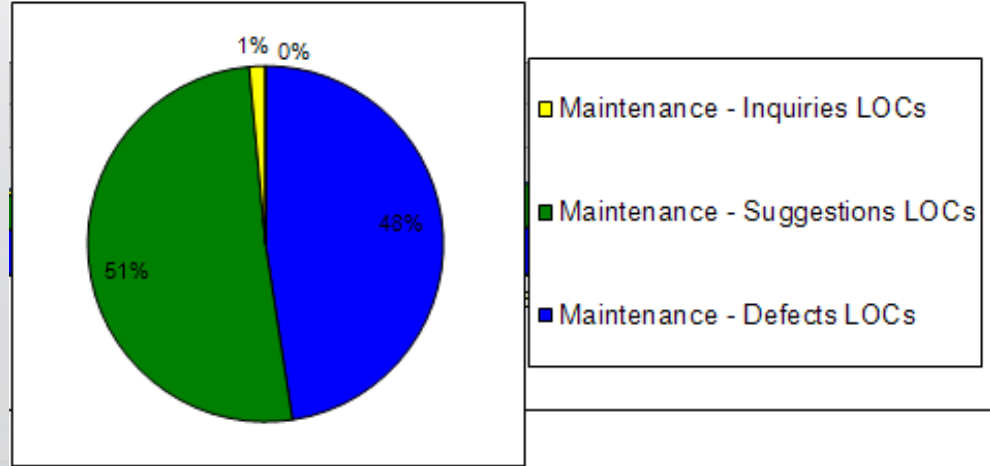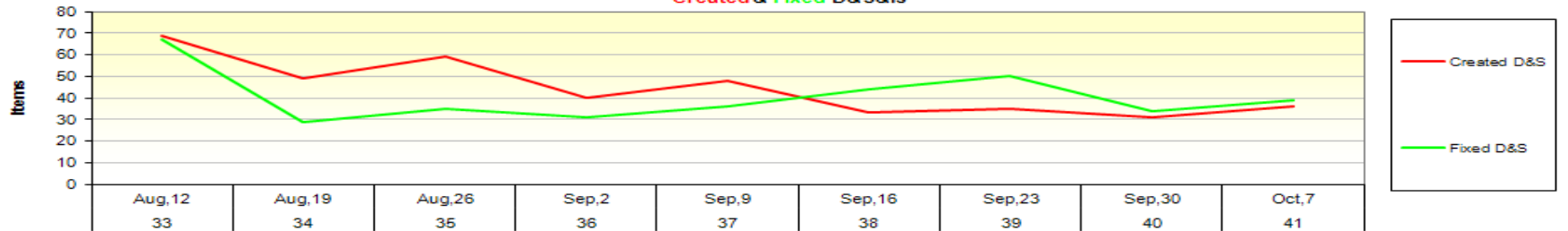# SQA Team Dashboard
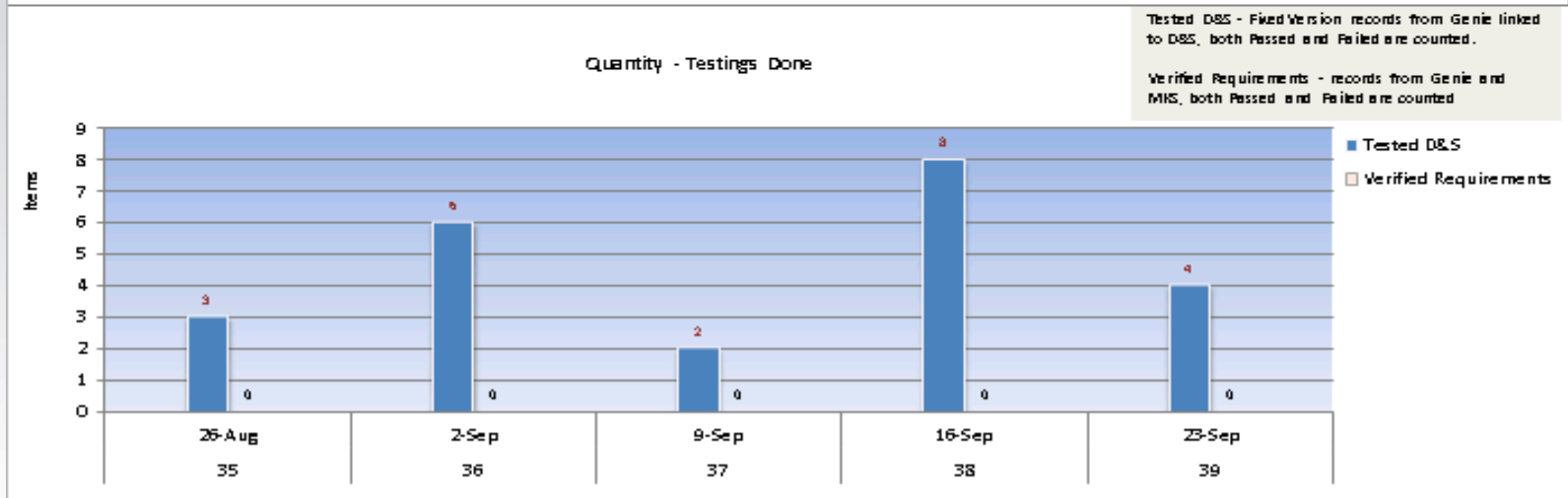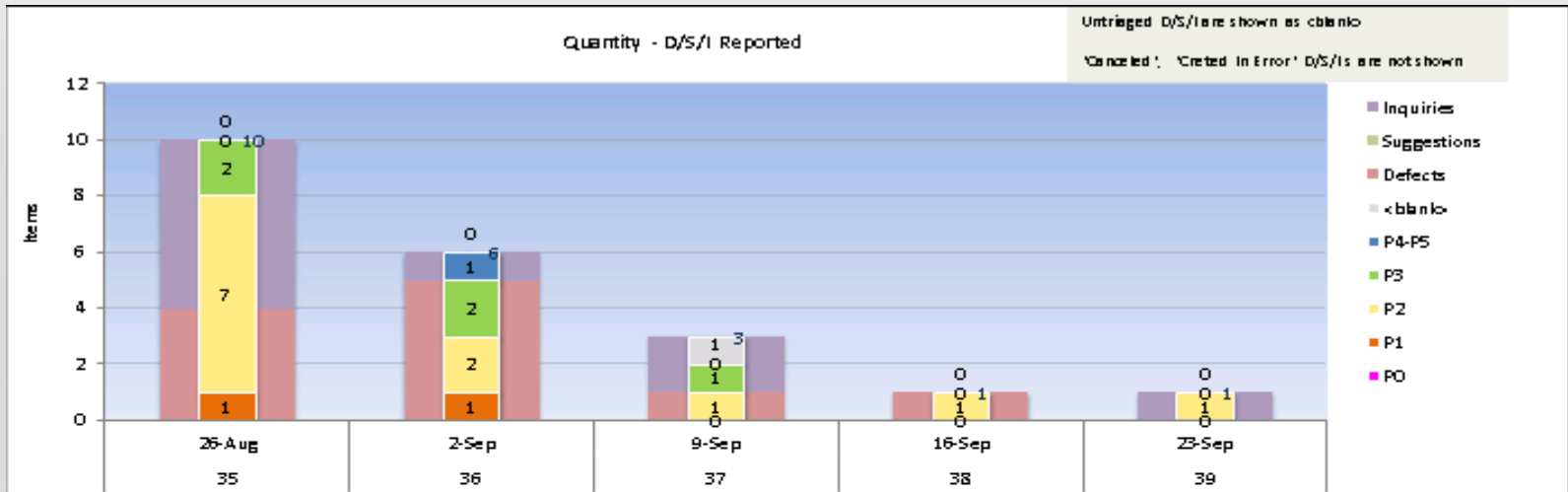
SQA Team Dashboard
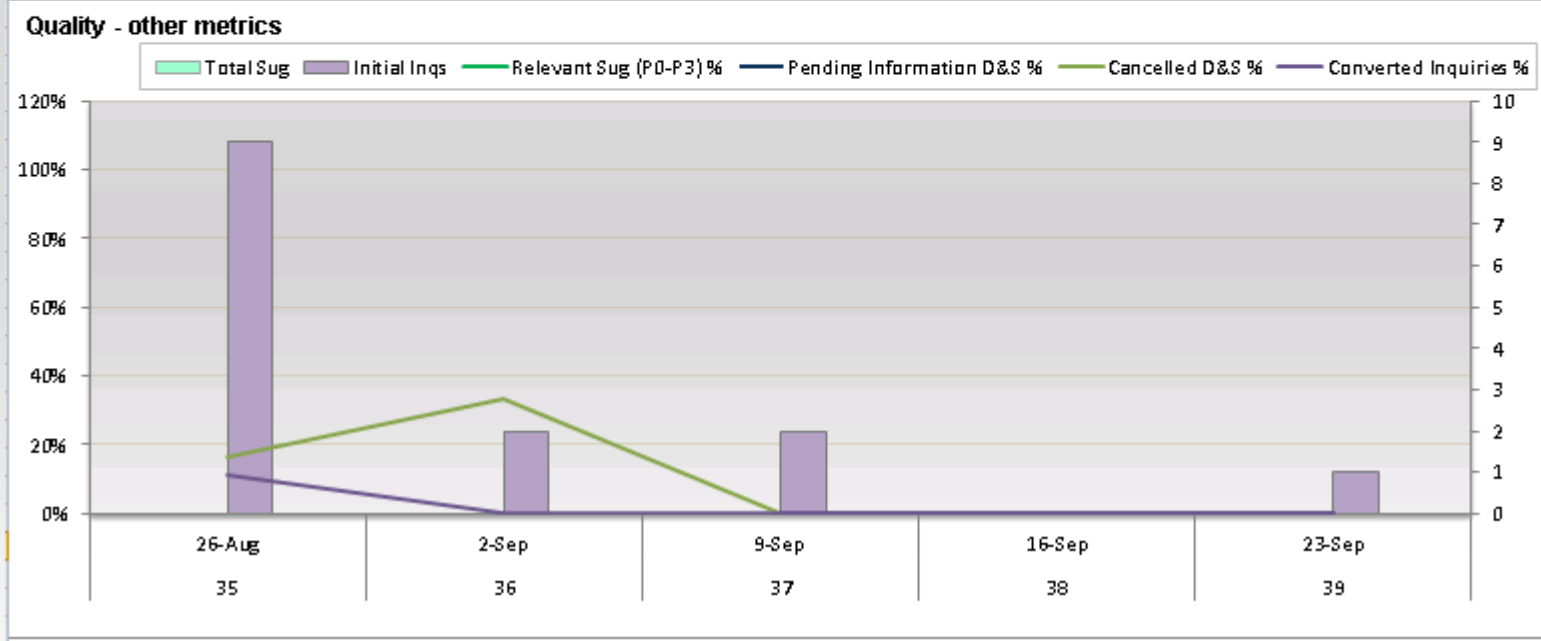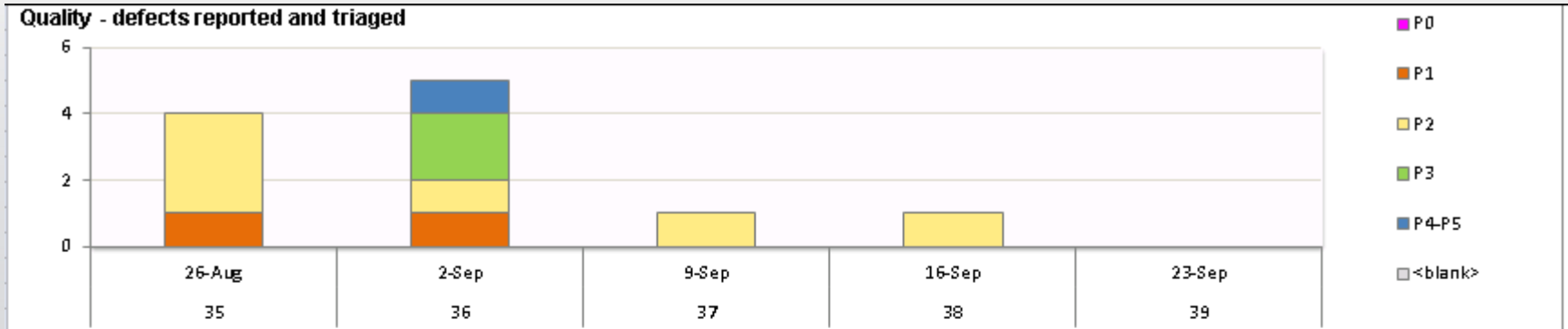
Maintenance Project
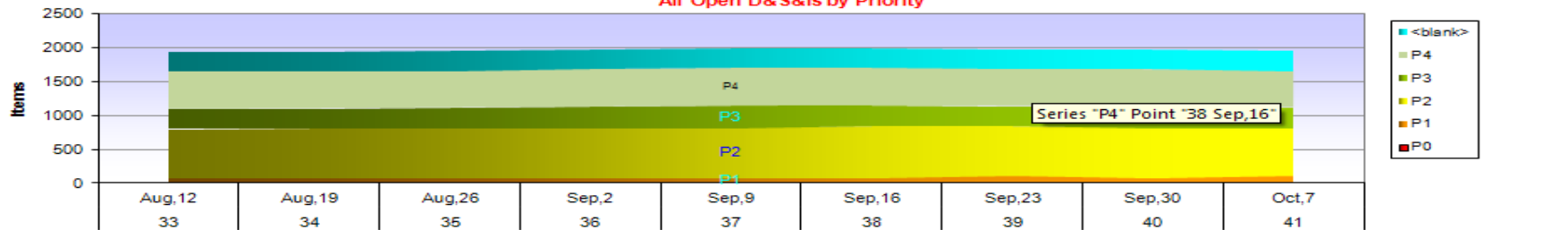
Analyze & Improve!

How to Improve?

Act!

- Communicate & Discuss
- Review & Inspect
- Monitor & Control
- Measure & Analyze
- Update the process
- Pilot updates
- Provide feedback
- Adjust the process

Process

Improvments Specifics

- Different development systems

- Different Testing teams

- Different product types

- Features implemented by different teams and combined into the end product

- Teams distribution and availability

- Testing approaches

Software

Quality

- Mostly depends on quality of software processes, like
  - Requirements
  - Inspections
  - Testing & Quality Assurance
  - Defects & Suggestions
  - Etc.
- Can be defined in terms of defects.
- Must be controlled through metrics.

Always search for ways of software quality and software processes improvement!

Learning on Mistakes

Data

- Project postmortems
- Project and Team level metrics
- Releases analysis
- Post-SQA defects review
- Customer Experience data analysis

Outputs

- Process analysis and update
- Updated Test Plans with new test cases
- Test coverage increase
- Performance and productivity increase

Summary

Presented

- Benefits of SQA Early involvement into the project
- Unified Testing Process and its customization
  - For any level of testing
  - For any development system
  - For any testing team
- Unified Test Plan and its benefits
- Improvements based on
  - Process and product quality measurement
  - Project and Team level metrics used in CQG

Thank You!

emmad@cqg.com
emma_danielyan@yahoo.com