Proactive SQA™ Overcomes the 'Traffic Cop' Mentality

Robin F. Goldsmith, JD

GO PRO MANAGEMENT, INC.

SYSTEM ACQUISITION & DEVELOPMENT

QUALITY/TESTING

PRODUCTIVITY

BUSINESS ENGINEERING

TRAINING

22 CYNTHIA ROAD

NEEDHAM, MA 02494-1412

INFO@GOPROMANAGEMENT.COM

WWW.GOPROMANAGEMENT.COM

(781) 444-5753 *VOICE/FAX*

Are You Familiar with QA as 'Traffic Cop'

www.123rf.com/photo_12808921

- Enforcing compliance
  - Document formats
  - Following procedures
- Obstacle to
  - Progress
  - Delivery
- Understandable ***RESISTANCE***

Objectives

- Distinguish system/software quality, quality assurance (SQA), and quality control (SQC).
- Analyze conventional SQA/standards and why they so often are resisted, ignored, and/or fail.
- Describe the six functions Proactive SQA™ performs so
  - Involved parties understand and willingly participate in meaningful methods to assure software quality
  - Resisted practices are reduced, such as being a 'traffic cop'
  - Higher quality software truly is delivered quicker and cheaper.

Proactive SQA™ is a key basis of significant value-enhancing revisions to IEEE SQA Std. 730-2014

Exercise: What is System Quality?

System Quality

meets the needs of your customer ecosystem friendly
Doesn't catch on fire Meets design spec cultural integration
Complete deliveable product operates efficiently
Scalable and performance discoverable meets cost
Reliable user-friendly meets reqs of the software
Sum of parts integrate together, hardware and software
Achieves objectives, sys reqs recommendable

Software Quality

Doesn't crash all of above except sum of parts integrated
System reqs allocated to software are achieved
Doesn't waste lots of user time maintainable accurate results
Organized well reusable

Exercise: What is SQA?

System Quality Assurance

Finding unexpected behaviors thoughtful approach and well-defined process to assure delivery of product that meets cust expectations provide info about whether reqs and acc criteria have been met

Using that data to improve processes

Holistic approach of where defects occur in system and why is product doing so terrible and who is resp for fixing them

Does sys do what we need it to do? Defect prevention

Quality Control

Testing and capturing data about testing

System vs. Software Quality Relevance to SQC/SQA

- At which life cycle phase is it decided whether solution includes hardware?
  - Requirements
  - Design
  - Build and test

- What impact on quality activities
  - If system vs. software initially misidentified?
  - If system vs. software subsequently changes?

Is system vs. software distinction relevant, useful?

Quality Is Key to Delivering Quicker and Cheaper

- "Quality is free"
- Cost of (poor) quality
  - Assessment (appraisal)
  - Prevention
  - Failure
    - » Internal
    - » External

-- Philip Crosby

Some Common Definitions of Quality

- Customer satisfaction
- Meets or exceeds customer expectations
- Optimization, value
- Conformance to requirements (Philip Crosby)
- Percent of (a sample of) products passing inspection for defects; lack of defects (~Deming)
- Minimal variation within specification (Six Sigma)
- Fitness for use (Joseph Juran)

Any problems with these definitions? Relation to systems?

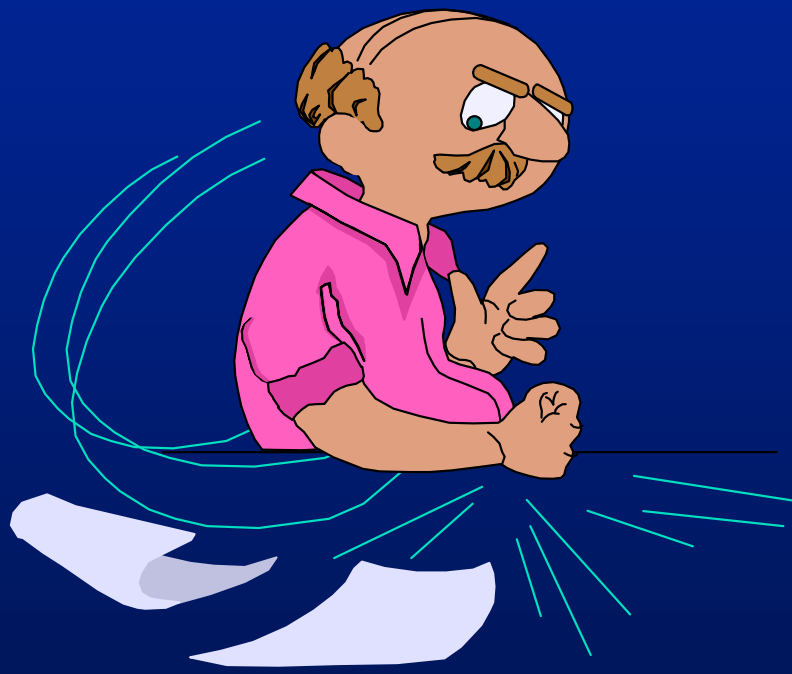
What We Mean By System Quality

- Fits system specs
- Runs efficiently
- Doesn't blow up
- Follows standards
- Current technology
- Modern techniques
- Easily modified
  - without code change
  - when code changes

What We Others Mean By System Quality

- Fits system specs
- Runs efficiently
- Doesn't blow up
- Follows standards
- Current technology
- Modern techniques
- Easily modified
  - without code change
  - when code changes
- Does what needs to be done correctly
- Performs adequately
- Reliable/consistent
- Easy to use
- Supported quickly and correctly
- On-time, in budget

Until We Share a Common Definition of System Quality...



- ✓ Users, managers, developers, and Quality professionals will continue to disappoint each other
- ✓ Each has a different idea of what to deliver and how to tell whether it has been delivered adequately
- ✓ Each thinks the others don't care about Quality

✪ *Quality Dimension: Quality of **Design** (What's it need to do)*

- Required functions, capabilities, and performance levels defined appropriately
  - needs of all stakeholders identified
  - definitions accurate and complete
  - meaningful common understanding
- Design suitably meets requirements
- Costs/benefits/schedules are accurate
- Trade-offs based on adequate information

✪ *Quality Dimension: Quality of* ***Conformance*** *(How it's produced)*

- Products conform to design
- Products apply standards/conventions
- Workers use expected skill and care
- Workers apply defined methods, tools
- Management uses appropriate practices
- Product is delivered on-time, in-budget

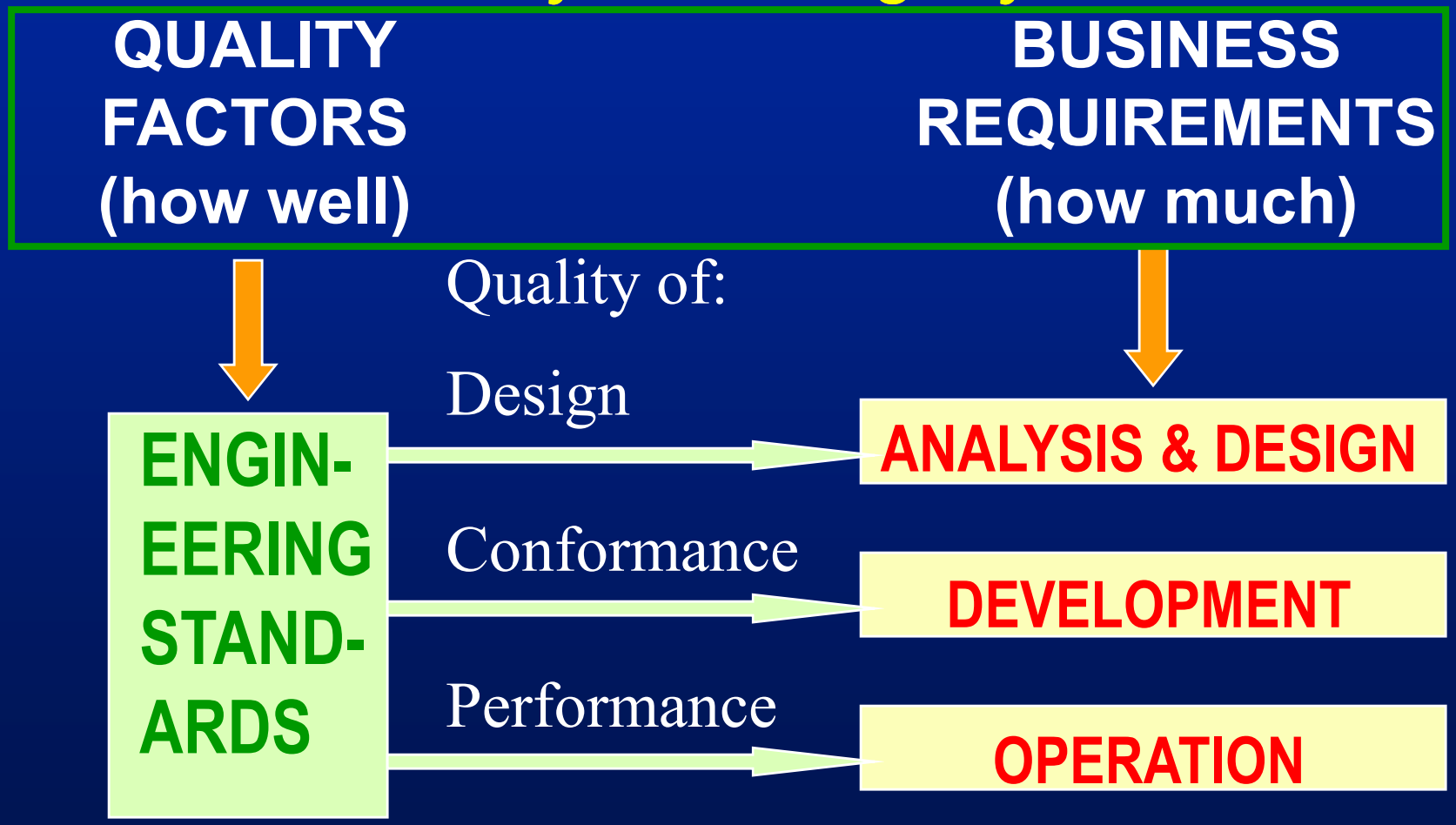
✪ *Quality Dimension: Quality of **Performance** (How it's delivered)*

- Product is available as needed for use
- Product works in intended manner
- Product works reliably and accurately
- Product handles workload adequately
- Product is supported and maintained responsively

✪ Addressing Quality Factors

Factors:
- Exterior
- Interior
- Future

Turning Requirements Into a Quality Working System



Our Working Definition of System Quality

The **extent** to which **it meets** weighted stated and implied exterior, interior, and future **REAL business requirements** of all affected internal and external stakeholders **consistent with standards** of design, workmanship, and performance.

The more of the relevant requirements which are met, and the more demanding the standards are with respect to meeting those requirements, the higher the quality.

Quality is absolute. The amount of quality one receives is governed by available resources, priorities, and other constraints.

Value is the perceived benefit of quality received relative to the costs of producing and receiving it.

Engineered Deliverable Quality™

How Much

How Well

Deliverable Capability	Weight/ Priority	Minimum	Desirable	Ideal

Quality Assurance (QA) vs. Quality Control (QC)/Testing

Dynamic Code Execution

- QC/Testing examines end products, typically for conformance to specifications (but which often are referred to as 'requirements')
- QA assures the processes producing the end products produce quality products

Static Reviews of Requirements and Designs

  - To some, that means examining intermediate products within the development process
  - Often checks compliance of documents/procedures to standards/guidelines ("traffic cop")

These are QC too—examining products

IEEE Std 12207-2008 Systems and software engineering —Software life cycle processes 1/4

7.2.3 Software Quality Assurance Process

7.2.3.1 Purpose

The purpose of the Software Quality Assurance Process is to provide assurance that work products and processes comply with predefined provisions and plans.

Starting point for revision of IEEE Std. 730 for SQA

IEEE Std 12207-2008 Systems and software engineering —Software life cycle processes 2/4

7.2.3.2 Outcomes

As a result of successful implementation of the Software Quality Assurance Process:

a) a strategy for conducting quality assurance is developed;

b) evidence of software quality assurance is produced and maintained;

c) problems and/or non-conformance with requirements are identified and recorded; and

d) adherence of products, processes and activities to the applicable standards, procedures and requirements are verified.

7.2.3.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Quality Assurance Process.

IEEE Std 12207-2008 Systems and software engineering —Software life cycle processes 3/4

7.2.3.3.1 Process implementation. This activity consists of the following tasks:

7.2.3.3.1.1 A quality assurance process suited to the project shall be established. The objectives of the quality assurance process shall be to assure that the software products and the processes employed for providing those software products comply with their established requirements and adhere to their established plans.

7.2.3.3.1.2 The quality assurance process should be coordinated with the related Software Verification (subclause 7.2.4), Software Validation (subclause 7.2.5), Software Review (subclause 7.2.6), and Software Audit (subclause 7.2.7) Processes.

7.2.3.3.1.3 A plan for conducting the quality assurance process activities and tasks shall be developed, documented, implemented, and maintained for the life of the contract. The plan shall include the following:

Original Std. 730 scope

a) Quality standards, methodologies, procedures, and tools for performing the quality assurance activities (or their references in organization's official documentation).
b) Procedures for contract review and coordination thereof.
c) Procedures for identification, collection, filing, maintenance, and disposition of quality records.
d) Resources, schedule, and responsibilities for conducting the quality assurance activities.

IEEE Std 12207-2008 Systems and software engineering —Software life cycle processes 4/4

e) Selected activities and tasks from supporting processes, such as Software Verification (subclause 7.2.4), Software Validation (subclause 7.2.5), Software Review (subclause 7.2.6), Software Audit (subclause 7.2.7), and Software Problem Resolution (subclause 7.2.8).

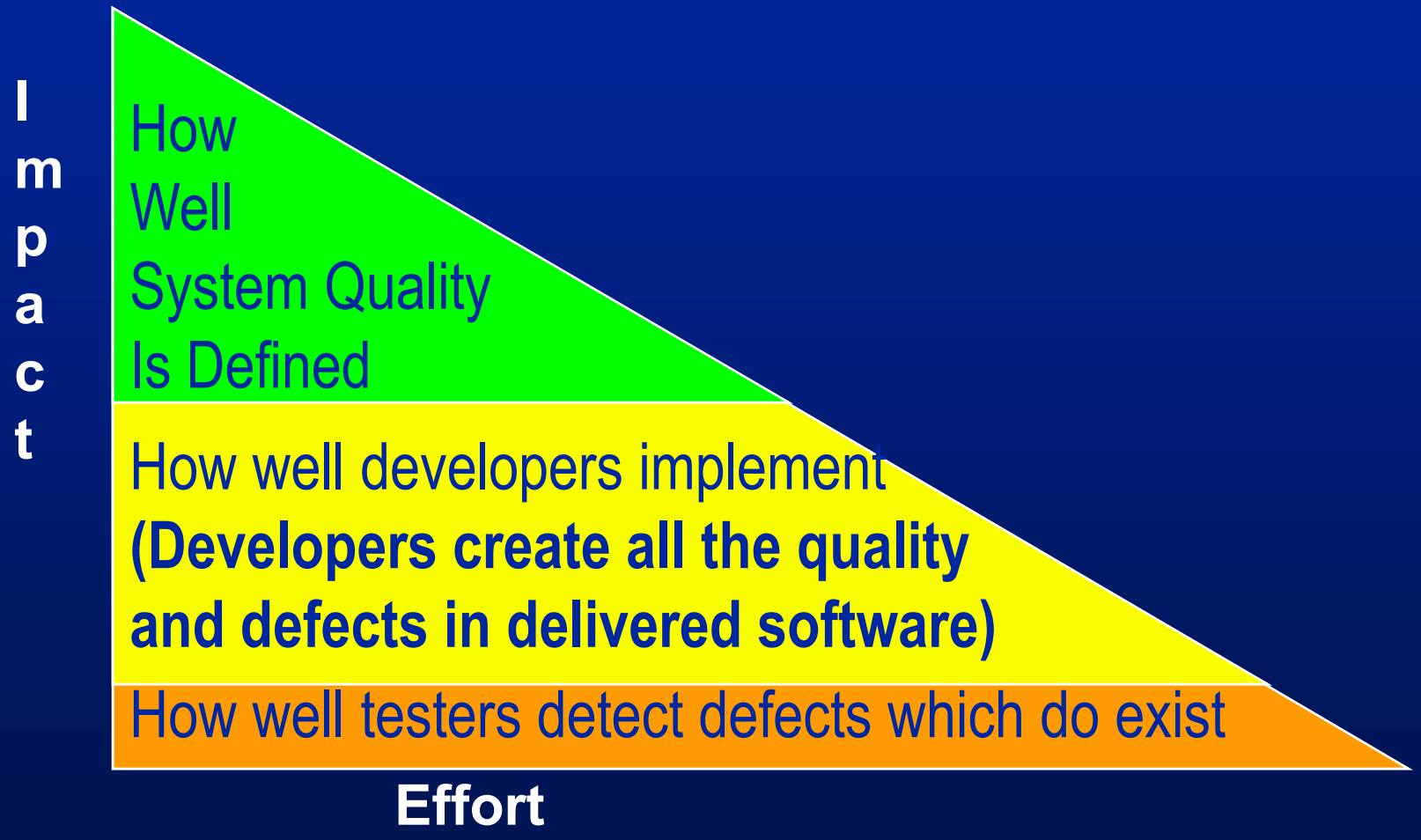
7.2.3.3.1.4 Scheduled and on-going quality assurance activities and tasks shall be executed. When problems or non-conformances with contract requirements are detected, they shall be documented and serve as input to the Problem Resolution Process (subclause 7.2.8). Records of these activities and tasks, their execution, problems, and problem resolutions shall be prepared and maintained.

7.2.3.3.1.5 Records of quality assurance activities and tasks shall be made available to the acquirer as specified in the contract.

7.2.3.3.1.6 It shall be assured that persons responsible for assuring compliance with the contract requirements have the organizational freedom, resources, and authority to permit objective evaluations and to initiate, effect, resolve, and verify problem resolutions.

How similar is this to what your organization does?

System Quality Results From

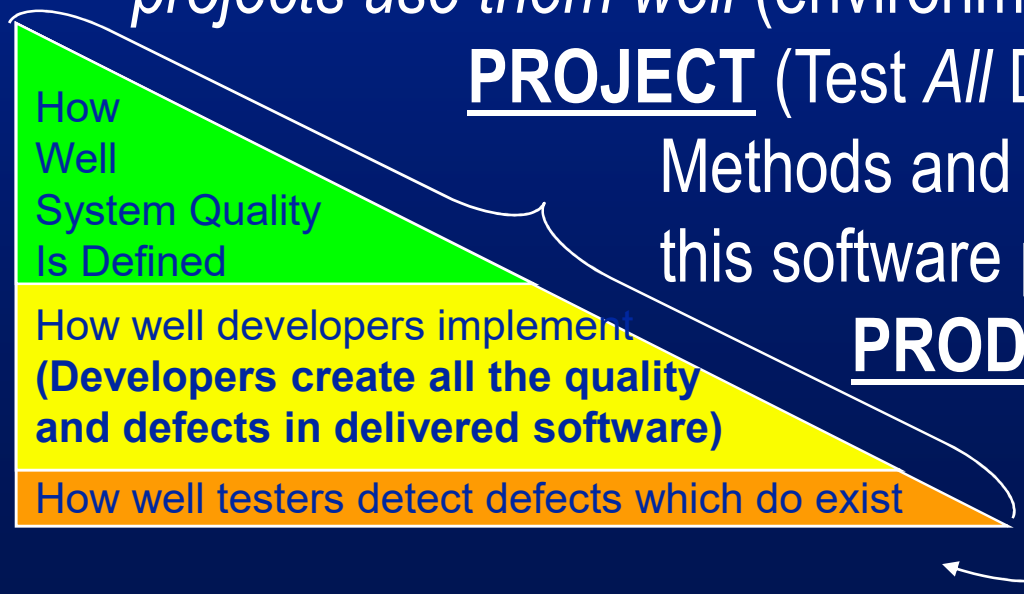


Proactive System Quality Assurance (SQA)™ Direction of New IEEE Std. 730

Value

PROCESS

***Define** appropriate methods and techniques and **assure** all projects use them well* (environment that promotes quality)

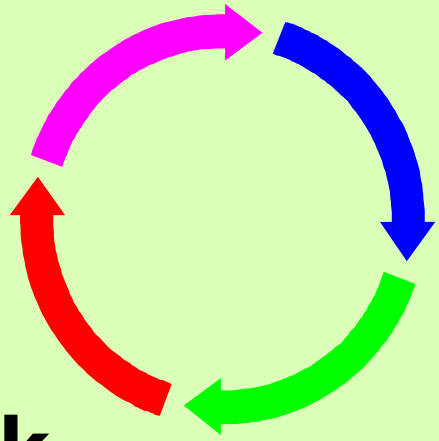


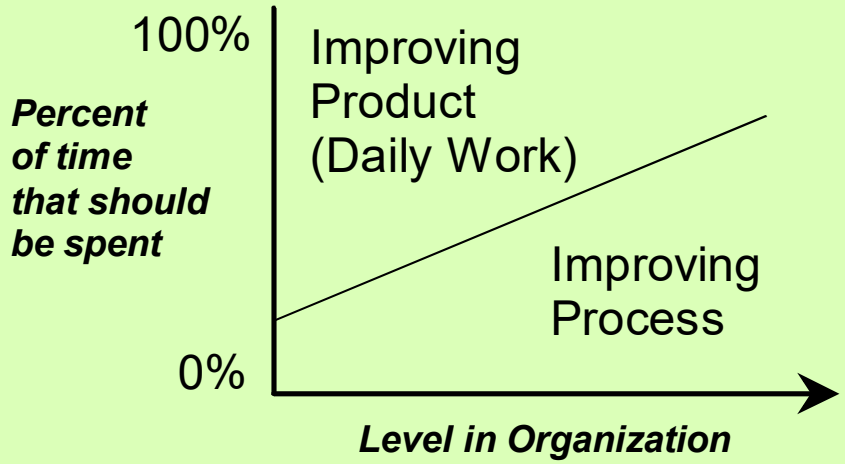
PROJECT (Test *All* Development Deliverables)
Methods and techniques used to create this software product were appropriate

PRODUCT (Testing the Code)
Delivered software works properly

Proactive SQA™: Establishes an Environment that Promotes Quality

Key Quality Environment Approaches





W. Edwards Deming, Walter Shewhart

Use data to fix the problem and the source of the problem

The 6 Functions of SQA

Actually HELP

Assure Vs. Do

① Define Quality Assurance Plans (What to do)

② Define, methods, practices, and standards (How to do it well)

③ Assure systematic quality controls of processes and products (Make sure it gets done right)

④ Maintain quality records (Keep track of it)

⑤ Analyze and report on quality (Learn from it)

⑥ Direct attention to improving quality (Encourage it)

1 Define Quality Assurance Plans

- The project plan for QA, becomes part of overall project plan—tasks, resources, budget, schedule
- Identifies every task and other information needed to assure software product quality
  - Templates, common to all projects
  - Tasks unique to project
  - Balanced with risk, needs, and constraints

Consider instead focusing on needed deliverables

- Used to monitor/control progress

Entire focus of IEEE Std. 730 until current revision

QA Plan, Very Detailed Deliverables

Mil. Std. 2167 Requirements Analysis Phase

Computer Software Configuration Item
- Functional Requirements
- Performance Requirements
- Interface Requirements
- Qualification Requirements
- Software Requirements Specification
- Interface Requirements Specification

Software Development Plan
Software Standards and Procedures Manual
Software Configuration Management Plan
Software Quality Evaluation Plan
Operational Concept Document
Software Specification Review
Allocated Baselines for each CSCI
- Authenticated SRS
- Authenticated IRS(s)

Ongoing Internal Reviews Verification

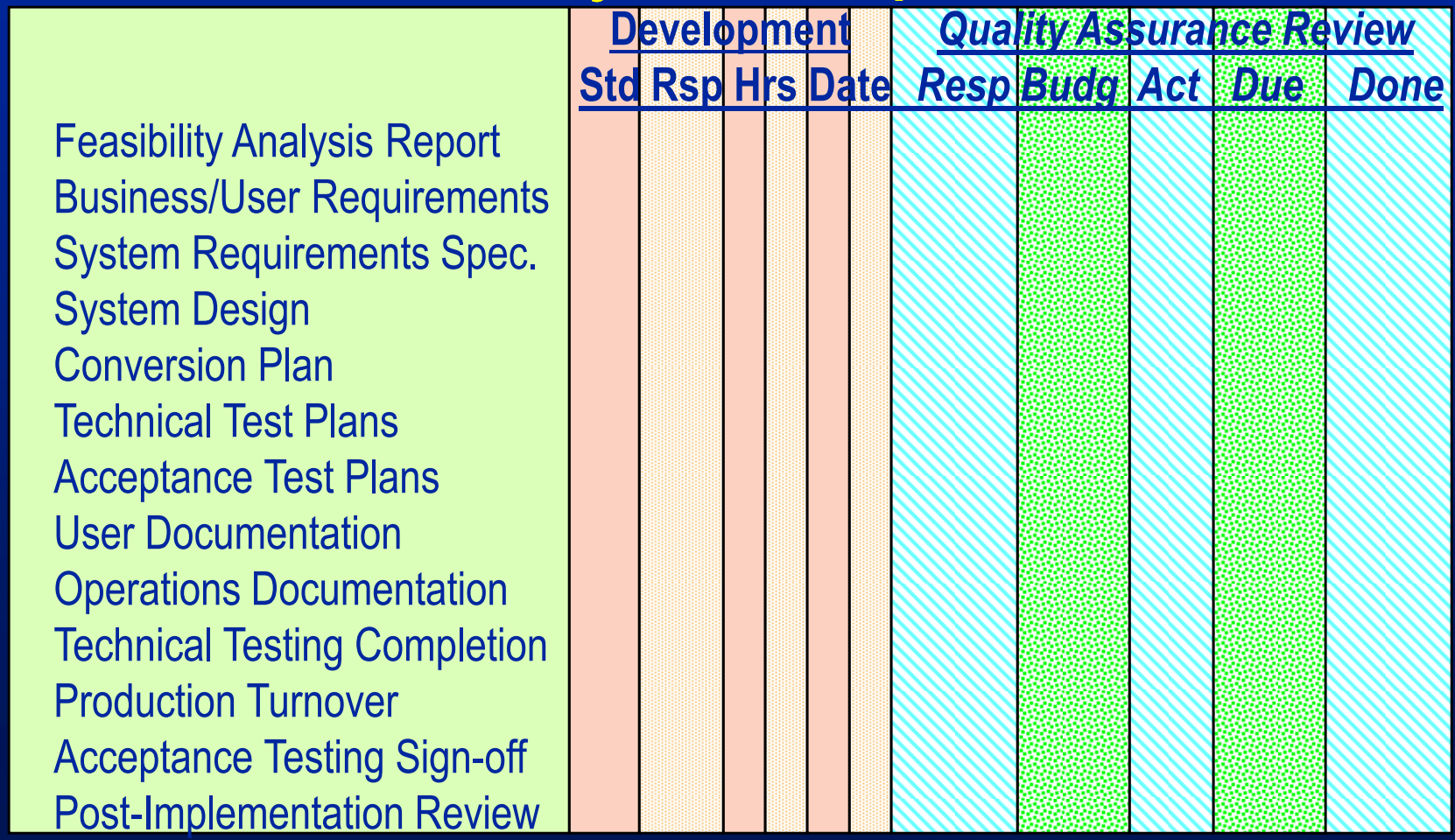
QA Plan Deliverables & Checklist
Generic Quality Checkpoints

	Date Completed
Feasibility Analysis Report	
Business/User Requirements	
System Requirements Spec.	
System Design	
Conversion Plan	
Technical Test Plans	
Acceptance Test Plans	
User Documentation	
Operations Documentation	
Technical Testing Completion	
Production Turnover	
Acceptance Testing Sign-off	
Post-Implementation Review	

QA Plan Deliverables & Action Plan Generic Quality Checkpoints

	Applicable Standards	Resp	Budget Hours	Actual Hours	Date Due	Date Done
Feasibility Analysis Report						
Business/User Requirements						
System Requirements Spec.						
System Design						
Conversion Plan						
Technical Test Plans						
Acceptance Test Plans						
User Documentation						
Operations Documentation						
Technical Testing Completion						
Production Turnover						
Acceptance Testing Sign-off						
Post-Implementation Review						

QA Plan Deliverables, QA Action Plan Generic Quality Checkpoints



Exercise: Managing SQA Tasks, Resources

How would you handle and account for?

Development deliverable is delivered after SQA review was scheduled to begin

SQA review finds a development deliverable inadequate and needs the deliverable to be corrected and re-reviewed

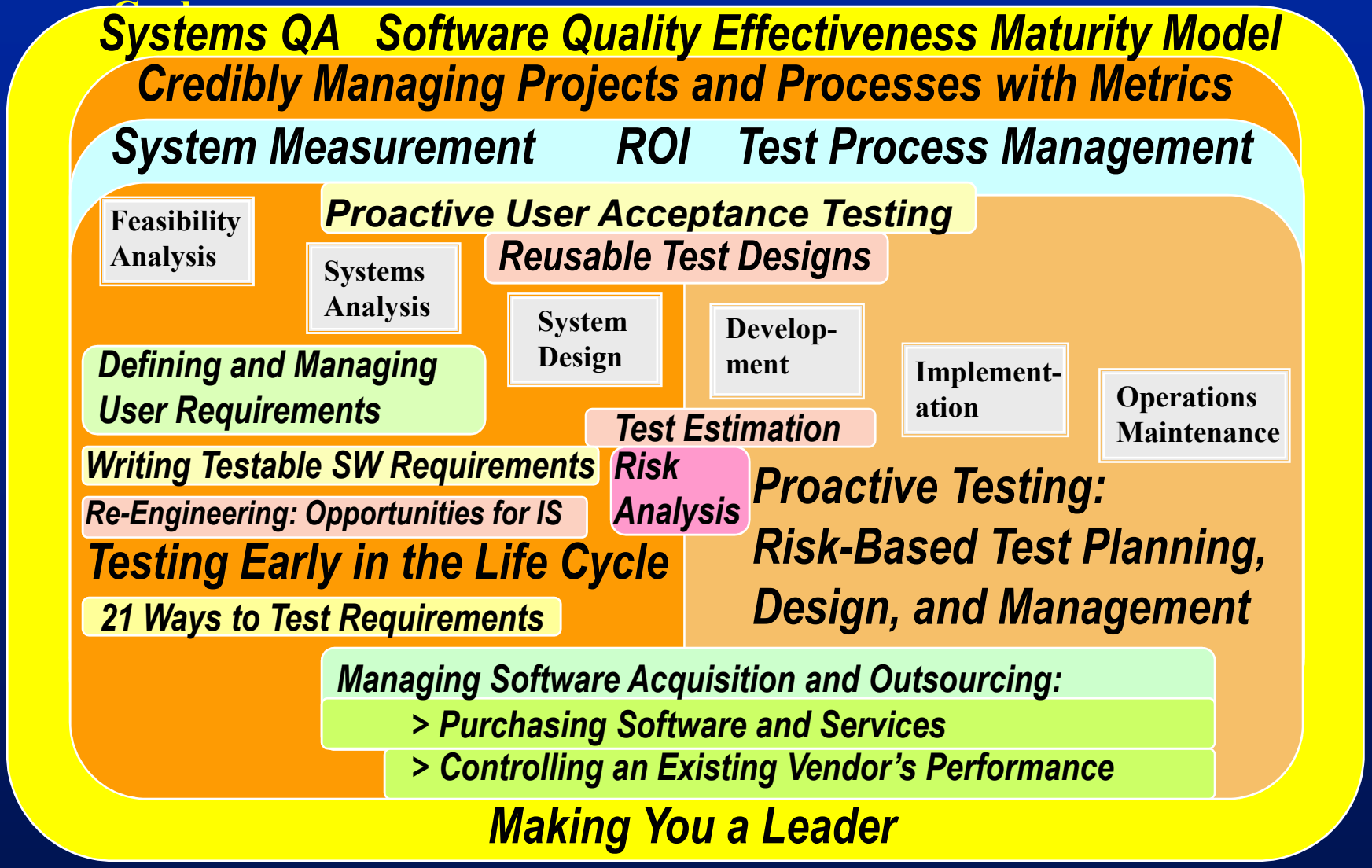
The SQA review takes longer and/or more effort than planned

Objectives

- Distinguish system/software quality, quality assurance (SQA), and quality control (SQC).
- Analyze conventional SQA/standards and why they so often are resisted, ignored, and/or fail.
- Describe the six functions Proactive SQA™ performs so
  - Involved parties understand and willingly participate in meaningful methods to assure software quality
  - Resisted practices are reduced, such as being a 'traffic cop'
  - Higher quality software truly is delivered quicker and cheaper.

Proactive SQA™ is a key basis of significant value-enhancing revisions to IEEE SQA Std. 730-2014

Go Pro Management, Inc. Seminars/Consulting--Relation to Life



Robin F. Goldsmith, JD

robin@gopromanagement.com (781) 444-5753 *www.gopromanagement.com*

- President of Go Pro Management, Inc. consultancy since 1982, working directly with and training professionals in business engineering, requirements analysis, software acquisition, project management, quality and testing.
- Partner with ProveIT.net in REAL ROI™ and ROI Value Modeling™.
- Previously a developer, systems programmer/DBA/QA, and project leader with the City of Cleveland, leading financial institutions, and a "Big 4" consulting firm.
- Degrees: Kenyon College, A.B.; Pennsylvania State University, M.S. in Psychology; Suffolk University, J.D.; Boston University, LL.M. in Tax Law.
- Published author and frequent speaker at leading professional conferences.
- Formerly International Vice President of the Association for Systems Management and Executive Editor of the *Journal of Systems Management*.
- Founding Chairman of the New England Center for Organizational Effectiveness.
- Member of the Boston SPIN and SEPG'95 Planning and Program Committees.
- Chair of record-setting BOSCON 2000 and 2001, ASQ Boston Section's Annual Quality Conferences.
- TechTarget, SearchSoftwareQuality requirements and testing subject expert.
- Member IEEE Std. 829-2008 for Software Test Documentation Standard Revision Committee.
- Member IEEE 730-2014 Working Group rewriting IEEE Std. 730-2002 for Software Quality Assurance Plans.
- International Institute of Business Analysis (IIBA) Business Analysis Body of Knowledge (BABOK) subject expert.
- Admitted to the Massachusetts Bar and licensed to practice law in Massachusetts.
- Author of book: ***Discovering REAL Business Requirements for Software Project Success***
- Author of forthcoming book***: Cut Creep—Put Business Back in Business Analysis to Discover REAL Business Requirements for Agile, ATDD, and Other Projects***