

# Applying Quality Principles to Software

David Walker

ASQ Kalamazoo/Battle Creek

January 13, 2009

David Walker SPCS, LLC



# Overview

- Software is Different
- Applying Conventional Quality Principles
- Solutions to Common Traps & Pitfalls
- Available Resources
- Professional Certification

# What Is Software?

- Instructions, Data

# What Is Software?

- Instructions, Data
- Layered Language Translations

# What Is Software?

- Instructions, Data
- Layered Language Translations
- The Variable Part of the System

# What Is Software?

- Instructions, Data
- Layered Language Translations
- The Variable Part of the System
- The Instantiation of a Process

# Characterization of Software

- Highly Complex, Somewhat Invisible  
*Difficult to mentally model all interactions*

# Characterization of Software

- Highly Complex, Somewhat Invisible  
*Difficult to mentally model all interactions*
- Highly Design and Labor Intensive

# Characterization of Software

- Highly Complex, Somewhat Invisible

*Difficult to mentally model all interactions*

- Highly Design and Labor Intensive

- Systematic Failure (random)

*Defects traceable to errors in design & development*

*Failure occurs without warning and at any time*

# Characterization of Software<sup>2</sup>

- Laws of Physics Do Not Apply

# Characterization of Software<sup>2</sup>

- Laws of Physics Do Not Apply
- SW Does Not Get Used Up or Wear Out

*Traditional products deteriorate over time*  
*Software gets better over time*

# Comparison

Most Products	Software
Predictable Behavior	Random Failure
Visible	Invisible
Difficult To Change	Easy To Change (superficially)
Production, Build	Design
Costs – Resources, Equipment, Supplies	Costs - Labor
Locality – Space, Time, Severity	No Inherent Locality
Defects in Single Units	Defects in Each Copy

# Fundamental Quality Principles Apply

## Customer Satisfaction

Determining who the customers are and what it takes to satisfy them.

# Fundamental Quality Principles Apply

## Problem Solving

Using four basic steps to implement solutions:

1. **Define the problem**
2. **Generate alternative solutions**
3. **Evaluate and select an alternative**
4. **Implement and follow up on the solution**

# Fundamental Quality Principles Apply

## Cost of Quality

The cost of NOT creating a quality product or service. Quality costs include:

1. **Prevention Costs**
2. **Appraisal Costs**
3. **Internal Failure Costs**
4. **External Failure Costs**

# Fundamental Quality Principles Apply

## Process Focus

Analyze how work gets done so that you can increase efficiency, effectiveness, and adaptability.

# Fundamental Quality Principles Apply

## Continuous Improvement

How to take your products, services and processes to the next level through an ongoing cycle of activities that capitalize on improvement opportunities.

# Fundamental Quality Principles Apply

## Quality Assurance and Quality Control

**Quality control**: The observation techniques and activities used to fulfill requirements for quality.

**Quality assurance**: The planned and systematic activities implemented in a quality system so that quality requirements for a product or service will be fulfilled.

# Fundamental Quality Principles Apply

## Supplier Quality

Select and work with suppliers in ways that will provide for high quality.

# Fundamental Quality Principles Apply

## Variation

Variation represents the difference between an ideal and an actual situation. Reducing the variation in stakeholders' experience is the key to quality and continuous improvement.

# Solutions To Unique Problems Complexity

- Modeling Tools

# Solutions To Unique Problems Complexity

- Modeling Tools
- Real Time Operating Systems

# Solutions To Unique Problems Complexity

- Modeling Tools
- Real Time Operating Systems
- Object Oriented Design and Development

# Solutions To Unique Problems Complexity

- Modeling Tools
- Real Time Operating Systems
- Object Oriented Design and Development
- Coding Standards and Standard Development Environments

# Solutions To Unique Problems

## Visibility

- Project Management Practices

# Solutions To Unique Problems

## Visibility

- Project Management Practices
- Configuration Management Practices

# Solutions To Unique Problems

## Visibility

- Project Management Practices
- Configuration Management Practices
- Modeling

# Solutions To Unique Problems

## Visibility

- Project Management Practices
- Configuration Management Practices
- Modeling
- Development Environments

# Solutions To Unique Problems

## Visibility

- Project Management Practices
- Configuration Management Practices
- Modeling
- Development Environments
- TDD (Test Driven Development)

# Solutions To Unique Problems

## Visibility

- Project Management Practices
- Configuration Management Practices
- Modeling
- Development Environments
- TDD (Test Driven Development)
- Frequent Customer Releases

# Solutions To Unique Problems Indirect Cause of Failure

- Inherent Defensive Design

# Solutions To Unique Problems Indirect Cause of Failure

- Inherent Defensive Design
- Design and Coding Standards

# Solutions To Unique Problems Indirect Cause of Failure

- Inherent Defensive Design
- Design and Coding Standards
- Reviews and Audits

# Solutions To Unique Problems Illusive & Latent Bugs

- Inherent Defensive Design

# Solutions To Unique Problems Illusive & Latent Bugs

- Inherent Defensive Design
- Tight Coding Structure and Memory Management

# Solutions To Unique Problems Illusive & Latent Bugs

- Inherent Defensive Design
- Tight Coding Structure and Memory Management
- Design and Coding Standards

# Solutions To Unique Problems Illusive & Latent Bugs

- Inherent Defensive Design
- Tight Coding Structure and Memory Management
- Design and Coding Standards
- Reviews and Audits

# Solutions To Unique Problems Illusive & Latent Bugs

- Inherent Defensive Design
- Tight Coding Structure and Memory Management
- Design and Coding Standards
- Reviews and Audits
- Development Environments

# Solutions To Unique Problems

## Synergy – Integrated Components

- Define Component Interface Requirements Early

# Solutions To Unique Problems

## Synergy – Integrated Components

- Define Component Interface Requirements Early
- Use Simulation to Ensure Integrity of Interface Descriptions

# Solutions To Unique Problems

## Synergy – Integrated Components

- Define Component Interface Requirements Early
- Use Simulation to Ensure Integrity of Interface Descriptions
- Maintain and Communicate Interface Descriptions

# Solutions To Unique Problems

## Synergy – Integrated Components

- Define Component Interface Requirements Early
- Use Simulation to Ensure Integrity of Interface Descriptions
- Maintain and Communicate Interface Descriptions
- Integration Testing

# Solutions To Unique Problems

## Testing – Too late and Insufficient

- Software Testing is Labor Intensive

# Solutions To Unique Problems

## Testing – Too late and Insufficient

- Software Testing is Labor Intensive
- Software Testing is NOT QA (SQA)

# Solutions To Unique Problems

## Testing – Too late and Insufficient

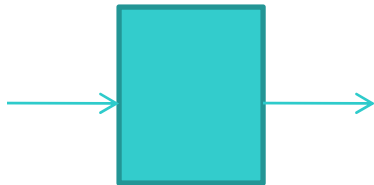
- Software Testing is Labor Intensive
- Software Testing is NOT QA (SQA)
- Quality Cannot be Tested into Software

# Solutions To Unique Problems

## Testing – Too late and Insufficient

- Software Testing is Labor Intensive
- Software Testing is NOT QA (SQA)
- Quality Cannot be Tested into Software
- Software Cannot be Fully Tested

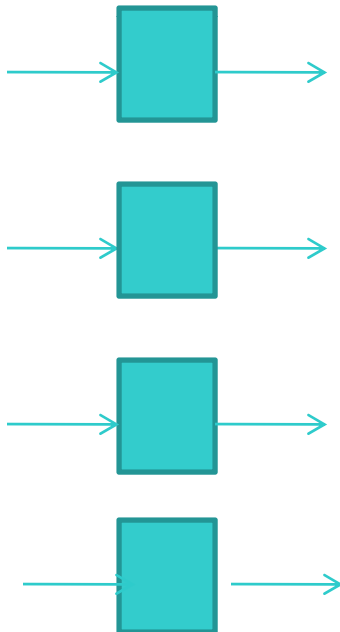
# Testing – System 1



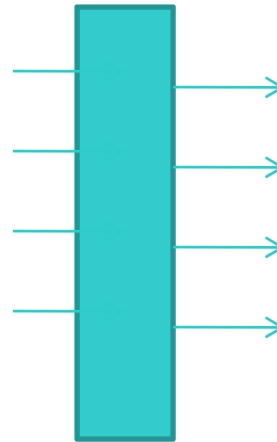
Two Tests: On/Off

# Testing – System 2

Eight Tests



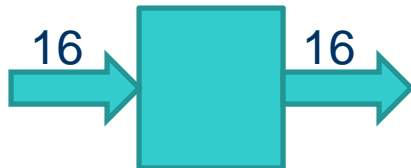
Sixteen Tests



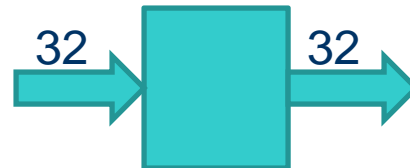
OFF	OFF	OFF	OFF
OFF	OFF	OFF	ON
OFF	OFF	ON	OFF
OFF	OFF	ON	ON
OFF	ON	OFF	OFF
OFF	ON	OFF	ON
OFF	ON	ON	OFF
OFF	ON	ON	ON
ON	OFF	OFF	OFF
ON	OFF	OFF	ON
ON	OFF	ON	OFF
ON	OFF	ON	ON
ON	ON	OFF	OFF
ON	ON	OFF	ON
ON	ON	ON	OFF
ON	ON	ON	ON

# Testing – System 3

65,000 Tests

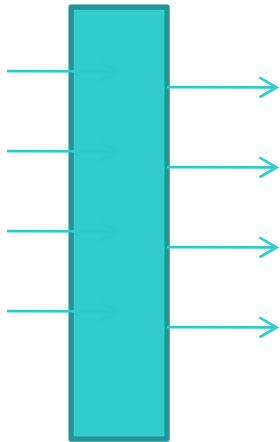


4.3 Billion Tests



1 Test Every 2 Seconds Would Take 272 Years

# Testing – System 2 Again



OFF	OFF	OFF	OFF
OFF	OFF	OFF	ON
OFF	OFF	ON	OFF
OFF	OFF	ON	ON
OFF	ON	OFF	OFF
OFF	ON	OFF	ON
OFF	ON	ON	OFF
OFF	ON	ON	ON
ON	OFF	OFF	OFF
ON	OFF	OFF	ON
ON	OFF	ON	OFF
ON	OFF	ON	ON
ON	ON	OFF	OFF
ON	ON	OFF	ON
ON	ON	ON	OFF
ON	ON	ON	ON

16! Sequences =

20 Trillion Tests

Would take 1.3 Million  
Years To Test (2s each)

# Solutions To Unique Problems

## The Death March

- Edward Yourdon's *Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*
- Process Focus and Measurement can profoundly improve decision making

# Available Resources

- ASQ Software Division WEB Site  
[www.asq.org/software](http://www.asq.org/software)

News, Conferences, Training, Online Library,  
Discussions Boards, Links, Publications,  
Jobs Page, Standards Updates,  
Certification Information, Speaker Information

# Available Resources

- SW QA and Testing Resource Center  
<http://www.softwareqatest.com/>

FAQs, Job Board, QA and Testing Tools Info,  
Book Recommendations, Term Index

# Available Resources

- “Handbook of Software Quality Assurance”  
4<sup>th</sup> Edition – Gordon Schulmeyer – 2007

ISBN-10: 1596931868

# Available Resources

- [StickyMinds.com](http://StickyMinds.com)
- Practical Software and Systems Measurement
- Better Software Magazine
- IEEE Software
- Great Lakes SW Excellence Conference
- CMMI Yahoo Discussion Board

# ASQ Certified Software Quality Engineer (CSQE)

The CSQE understands software quality development and implementation, software inspection, testing, verification and validation; and implements software development and maintenance processes and methods.

# CSQE BOK (Body of Knowledge)

## I. General Knowledge

- A. Quality principles
- B. Ethical and Legal Compliance
- C. Standards and models
- D. Leadership skills
- E. Team Skills

# CSQE BOK (Body of Knowledge)

## II. Software Quality Management

- A. Quality Management System
- B. Methodologies
- C. Audits

# CSQE BOK (Body of Knowledge)

## III. Systems and Software Engineering Processes

- A. Lifecycles and process models
- B. Systems architecture
- C. Requirements engineering
- D. Requirements management
- E. Software analysis, design, and development
- F. Maintenance management

# CSQE BOK (Body of Knowledge)

## IV. Project Management

- A. Planning, scheduling, and deployment
- B. Tracking and controlling
- C. Risk management

# CSQE BOK (Body of Knowledge)

## V. Software Metrics and Analysis

A. Metrics and measurement theory

B. Process and product measurement

C. Analytical techniques

# CSQE BOK (Body of Knowledge)

## VI. Software Verification and Validation (V&V)

- A. Theory
- B. Test planning and design
- C. Reviews and inspections
- D. Test execution documentation
- E. Customer deliverables

# CSQE BOK (Body of Knowledge)

## VII. Software Configuration Management

- A. Configuration infrastructure
- B. Configuration identification
- C. Configuration control and status accounting
- D. Configuration audits
- E. Product release and distribution

# Summary

- Software is Highly Complex and Mostly Design Activity
- Fundamental Quality Principles Apply, Tactics are Different
- There are Solutions to the Challenges Presented in Assuring Quality in Software
- BOK Has Been Defined for Software Quality Assurance Engineering - ASQ

# Discussion



David Walker SPCS, LLC



# Materials

These Slides [www.davidwalkerspcs.com/downloads](http://www.davidwalkerspcs.com/downloads)

David Walker, ASQ CSQE,  
SEI Authorized CMMI Instructor  
[dave@davidwalkerspcs.com](mailto:dave@davidwalkerspcs.com)  
269-352-9000

David Walker SPCS, LLC

