Measurements and Checkpoints

Bob Altizer

BASYS Consulting
Phoenix, AZ

bob.altizer@basysconsulting.com
http://www.basysconsulting.com

BASYS Consulting

Measurements and Checkpoints *Agenda*

- **Effective measurements**
- **Selecting and using checkpoints**
- **Summary of key points and benefits**

Effective Measurements

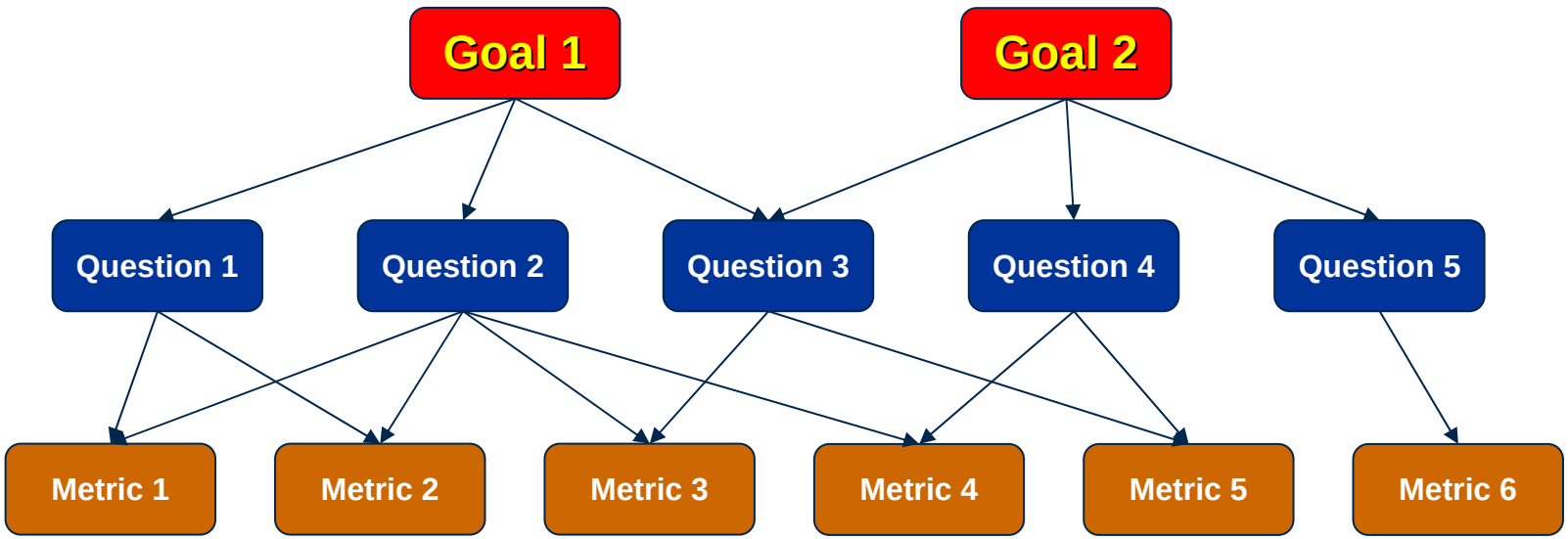
BASYS Consulting

Attempts to Use Metrics Often Fail

- Organizations try to define “standard” metrics with no relationship to business goals
- Data are sporadic or incomplete; computations misleading; reporting poor
- Management fails to use the metrics they have, or to demand more useful metrics
- Nobody says “Yeah, so what?” to drive change
- Metrics are viewed as non-essential, overhead activity
- Resistance to metrics in fear of personal retribution

Goal-Question-Metric (GQM) Model

- Business goals are the fundamental drivers
- Operational questions assess goal achievement
- Use metrics to answer operational questions



Source: Basili et al., "The GQM Approach"

A Goal-Question-Metric Example

- **GOAL: Minimize defects in released products**

- **QUESTIONS**
  - How many post-release defects are there?
  - What kinds of defects show up most frequently?

- **METRICS**
  - Customer-reported unique defect (CRUD) count
  - Defect type distribution
  - Defect trends (count, type) over time

Defining Business Goals

- **Goals must lead to a desirable business outcome, and be explicitly stated and adopted**
- **Goals can be externally focused (customer and product) or internally focused (process)**
  - Develop more new products for the market (external)
  - Enhance existing products (external)
  - Improve delivery cycle time (internal)
  - Reduce cost of poor quality (internal)
- **Goals are owned by an individual with the responsibility and authority to ensure they're met**
- **Progress toward goals is supported by measurement**
  - Track current status and accomplishments
  - Drive corrective and improvement actions

GQM Example: Requirements

- **Enterprise Goal**
  - Six-month development and release cycle time
- **Requirements Goal**
  - Requirements process supports a six-month development and release cycle time
- **Questions**
  - Does the existing system meet the six-month goal?
  - Just how efficient is our current requirements process?
  - What requirements-based problems do we have?
  - What is the cycle time impact of requirements changes to a project?
  - What kind of improvements should we focus on?

Continued …

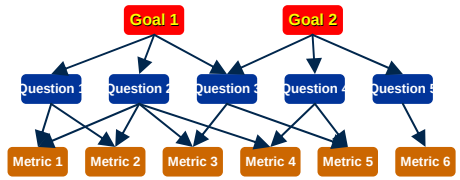
Goals and Questions Lead to Metrics

Some standard requirements-oriented metrics:

- Count (per project, bundle, release, etc.)
- Type (internal or external; product or process; etc.)
- Priority and risk distribution
- Complexity (I/Os, user interactions, interfaces, files affected)
- Integrity (change requests; changed vs. unchanged)
- Volatility (new, dropped)
- Quality (standards compliance; rework)
- Performance to plan (number planned, accepted, completed)
- Defects (number of defects; defect type and distribution)
- Phase containment effectiveness (early detection of defects)

“Yeah, now what?”

- **Accomplish business goals through quantitative management of requirements:**
  - Create a measurement infrastructure: initially set up in a project, ongoing operation as a service
  - Define requirements goals of process owners and stakeholders, at organizational and project level:
    - Business group champions and sponsors
    - Business and technical project managers
    - Process Assurance function



  - Develop requirements-related questions, measurement plans, and metrics (organization wide and per-project)
  - Start collecting, reporting, and using measurements
  - Use measurement-based, goal-oriented decisions in organization and project management

Coda: Six Sigma for Software

- **Bad News:**
  - Classical Six Sigma measures developed for control and improvement of a repetitive manufacturing process are often not directly applicable to other processes
- **Good News:**
  - The fundamental principle of improvement through quantitative analysis, and many of the individual tools and practices, are highly applicable to software development
    - DFSS (Design for Six Sigma)
    - DMADV (Define, Measure, Analyze, Design, Verify)
    - DMAIC (Define, Measure, Analyze, Improve, Control)
- **Use DFSS principles in both process and product or project development**
  - "The DFSS methodology can greatly improve timely discovery of latent or hidden requirements …" (Gack & Robison, in *Software Quality Professional*)

Measurements Summary

1. **Every goal and metric must have an owner and a clear business purpose**
2. **Effective measurement drives progress toward business goals, product and process improvement**
3. **Use the proven Goal-Question-Metric approach, a recognized best practice**
4. **Plan, budget, and schedule for effective measurement (organizationally, per-project)**
5. **Make use of the measurements, or don't do them ("Yeah, so what?")**
6. **Set up the measurement infrastructure in a project, then run it as a service**
7. **Use Six Sigma principles and tools as appropriate**

Measurements Benefits

- **Management at all levels can make better decisions when informed by effective measurements**
  - Organizational: product and process improvement
  - Per-Project: focus on highest priority needs
- **Every goal and metric has an owner and a clear business purpose**
- **All measurement is based on the GQM best practice**
- **Measurement infrastructure is usable across all process and life cycle activities**

Checkpoints

BASYS Consulting

What are Checkpoints?

- **Management views into the status of a project, connected with some well-defined unit of work, milestone, or time frame (schedule or event driven)**
  - Regular project status reviews
  - Phase-end reviews
  - May involve a go/no-go decision (Risk: social promotion)

- **Technical reviews of project work products for a particular purpose**
  - Walk-throughs, ballpark and detailed sizing, inspections, architecture and feasibility reviews, etc.
  - May identify need for rework (Risk: rework deferred)

Example: Requirements Process Checkpoints

- **Need and Estimation**
  - Sizing and feasibility
  - Executive sponsor approval
- **Requirements Definition**
  - Management sign-off
  - Quality standards compliance
  - Peer review approval
  - System feasibility
- **Requirements Review and Draft Estimate**
  - Review SLA compliance
  - Draft estimate completeness
  - Target timeframe feasibility
- **Functional Specification and Sign-Off**
  - Functional specification completeness and accuracy
  - Prototype correctness and accuracy
  - Business and technical sign-off
- **Detail Sizing and Release Assignment**
  - Detail sizing completeness
  - Detail sizing SLA compliance
  - Release assignment acceptability

Measurement Opportunities

- **Management checkpoints**
  - Service Level Agreement (SLA) compliance
  - Completeness and accuracy
  - Process-oriented measurement and feedback
  - Process correction and improvement

- **Technical checkpoints**
  - Specific requirements and design elements
  - Inspection for defects
  - Validation
  - Product-oriented measurement and feedback
  - Specific item rework

Calibrate Checkpoint Types: Formality Spectrum

- **Formality attributes:**
  - Level of defined process and specified roles
  - Amount of formal training needed to conduct
  - Time required to prepare, conduct, and follow-up
  - Flexibility in scheduling
  - Amount of support needed
  - Opportunity for measurement
- **Example: Peer Reviews**
  - Inspection (highly formal)
  - Team Review
  - Pair Programming
  - Walkthrough (moderate)
  - Peer Desk Check
  - Pass-around
  - Ad Hoc Review (informal)

Checkpoint Opportunities

1. **Verify the proper checkpoints are used within the requirements workflow**
2. **Streamline checkpoints to optimize actual flow (not just the "happy path")**
3. **Formalize processes as needed for all checkpoints (e.g., peer review spectrum)**
4. **Include measurements as appropriate at all checkpoints, supporting Goal-Question-Metric (GQM) approach…**
5. **Then manage processes and products based on measurements**

Checkpoint Benefits and Risks

- **Benefit: The right checkpoints are used within the requirements workflow**
  - Streamlined for efficiency
  - Right level of formality
  - Optimal measurements
  - Effective product and process management

- **Checkpoint Mismanagement Risks**
  - Management Checkpoints
    - Inadequate information collection
    - Incorrect view of project or organization status
    - Social promotion
  - Technical Checkpoints
    - Inadequate information collection
    - Incorrect specification, design, construction
    - Defect escape

Summary of Benefits: Measurements and Checkpoints

- **The improvement efforts should include coordinated introduction of:**
  - Effective, goal-oriented measurements via GQM
  - Calibrated workflow checkpoints
  - Enhanced risk identification and mitigation
  - Common process, training, and tools

- **Fundamental benefits include:**
  - Better business decisions
  - Higher performance, lower risk workflow
  - Lower overall project risk
  - Better project planning and scheduling
  - Standard processes, tools, and training for all employees

Thanks for your kind attention!

Backup slides following …

BASYS Consulting

Measurement Backup Slides

BASYS Consulting

Effective Quantitative Management

- **An effective, quantitative management program includes:**
  - Defined quality attributes ("Critical-To-Quality requirements") with significant business value, commitment of a named owner
  - Business-oriented quality management plans at every level
    - Quality goals and metrics
    - Preventative and corrective action plans
    - Defect prevention plans
    - Strategic improvement plans
  - Reviews of performance against quality goals and action triggers
  - Using metrics to manage activities and make product and process change decisions

Practical Steps to Making Measurements Effective

- **Derive specific metrics from operational questions, using either known standard metrics or new ones as needed**
- **Define details for each metric:**
  - Information Type (attribute, base, derived, indicator, decision criteria, etc.)
  - Responsibility for collecting, compiling, computing, publishing
    - Algorithm and data
    - Data source; frequency of collection
    - Sample charts or graphs
  - Where, when, and how reported and reviewed
    - Standard review questions
    - Two-way responsibility: metric owner, metric provider
  - Product and process controls
    - Baseline performance and control limits; performance goals
    - Preventative and corrective action plans and triggers

More GQM for Requirements Assurance

- **GOAL: Improve requirements through an effective metrics program.**

- **QUESTIONS**
  - How effective is the current requirements process?
  - What are the highest priorities for immediate change?
  - What rate of improvement can we maintain?

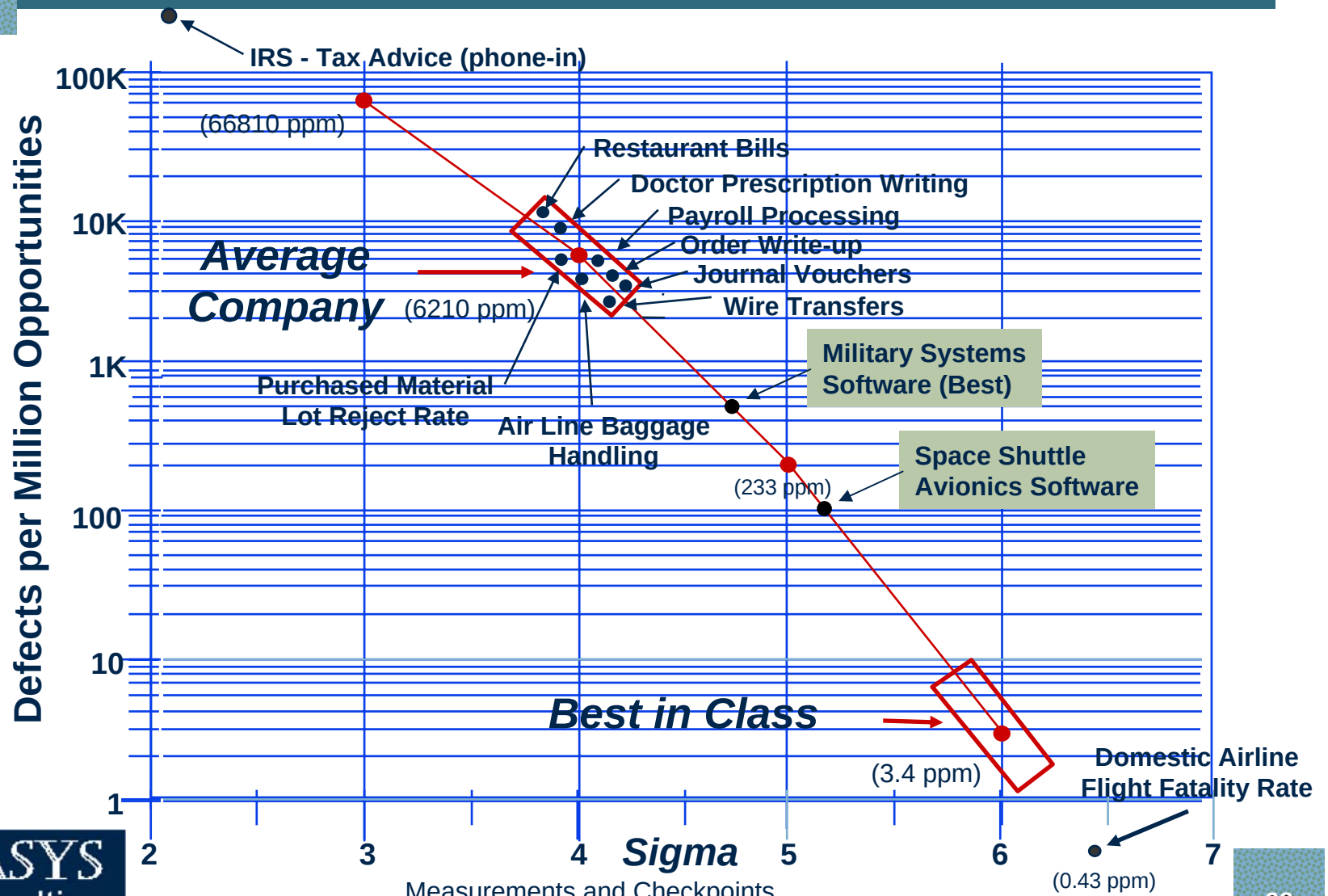
- **METRICS**
  - Cost and cycle time impact of poor requirements quality
  - Number of projects with requirements measurement plans
  - Number of requirement owners employing effective requirements reviews
  - Number and impact of requirements process changes

Common Hazards for Measurement Programs

- **Lack of sustainable measurement plans**
  - Make management with measurements a normal practice
- **Measurement per se is the goal**
  - Maximize ROI by targeting key decision making processes
- **Cultural issues treated as just resistance to change**
  - Accept concerns with measuring personal attributes or behaviors
- **Data are the responsibility of those being measured**
  - Build an environment of trust with consistent, correct use of data
- **Measurement results kept "secret" or not acted upon**
  - Be prepared to act on information revealed by measurement

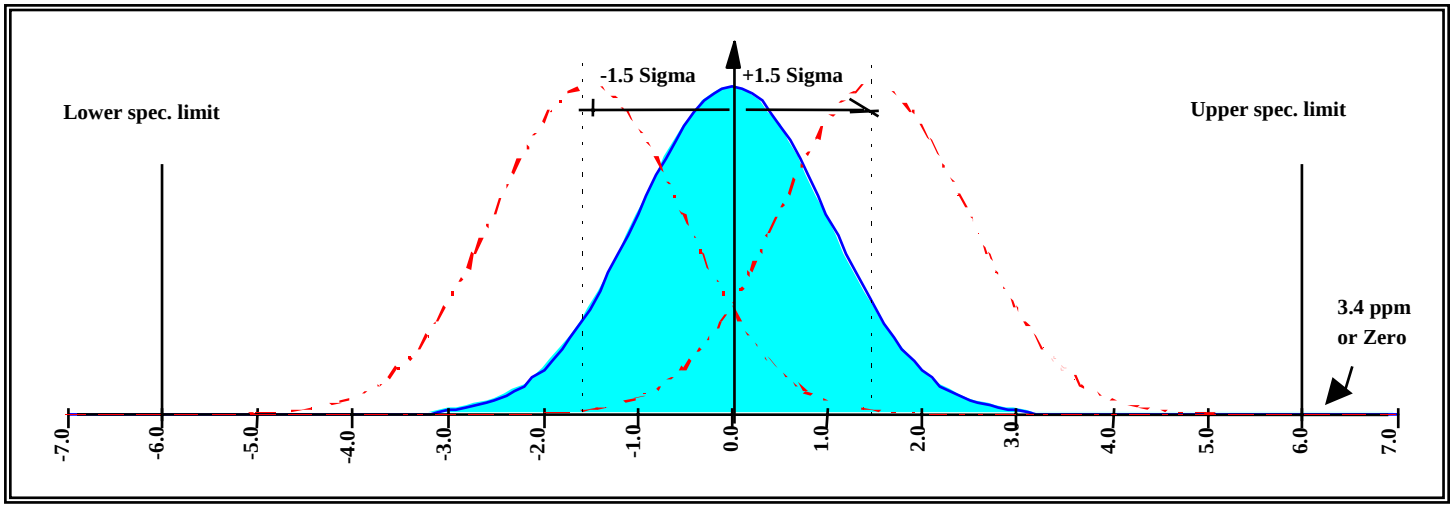
Source: McQuaid & Dekkers in *SQP*, 3/04

Benchmarking Quality Performance



BASYS Consulting

Six Sigma™ Measures Process Variance



- "Sigma" is a statistical measure of how a repeatable process complies of with specification limits
- In a "Six Sigma process," predictable sources of variation have been controlled or eliminated
- Process shifts due to unpredictable sources of variation can be accommodated without going out-of-spec

Information Types

Info Type	Description	Example
Attribute	**Inherent property of an entity relevant to information needs**	**Size of the hardware work product** **Time spent working (time cards)** **Defects reported** **Project roster**
Base Measure	**A measure of a single attribute by a specific method**	**Source lines of code (SLOC)** **Hours charged to the project** **Number of defects reported** **Employee's subject matter experience**
Derived Measure	**A function of two or more base measures**	**Productivity (SLOC/hr) per experience level** **Error Rate (Defects/SLOC) per experience level**
Indicator	**Estimate or evaluation that provides a basis for decision making; a function of two or more measures**	**Baseline (normal) productivity and error rates for similar projects** **Variation around normal: Standard deviation**
Decision Criteria	**Numerical thresholds, targets, and limits used to determine the need for action or investigation**	**Productivity control limits: Will the project be completed on time?** **Defect control limits: Is the project error rate too high?**

Source: McGarry et al., "Practical Software Measurement"

Risk Backup Slides

BASYS Consulting

What is a "Risk?"

- **Definitions**
  - Risk: The possibility of suffering a loss
  - Issue: A risk that has become a reality
- **A risk must be expressed clearly, describing:**
  - The current conditions that may lead to the loss (probability)
  - A description of the loss (impact)
- **Risk**
  - "The team will need re-training and may lose a week of development time if initial training is ineffective."
- **Not a risk**
  - "Our test environment resources can't handle all the functions in the upcoming release." (Already an issue!)

Risk Management Goals

- **Enhance and document current risk identification and management processes to include:**
  - Requirement-level risks
  - Multiple risk identification and assessment phases per project
  - Automated tool support
- **Include risks in feature and requirement prioritization; development and release planning**
- **Use risk management to drive iteration planning for multi-release projects (as in RUP)**
- **Employ use cases and use case flows as appropriate to help mitigate risks**
- **Track mitigation and risk status along with requirements**

Taxonomy of Software Risks

Requirements Risk Types:

- Stability
- Completeness
- Clarity
- Validity
- Feasibility
- Precedent
- Scale

Other important risk areas:

- Design
- Coding, integration, and test
- Development process
- Development system
- Management process
- Management methods
- Work environment
- Resources
- Program interfaces
  - Customer
  - Corporate management
  - Vendors
  - Politics

Source: Carr et al., *Taxonomy-Based Risk Identification*

Requirements Risk Questions

Stability:	Are requirements changing even as the product is being produced?
Completeness:	Are requirements missing or incompletely specified?
Clarity:	Are requirements unclear or in need of interpretation?
Validity:	Will the requirements lead to the product the customer has in mind?
Feasibility:	Are requirements infeasible from an analytical point of view?
Precedent:	Do requirements specify something never done before, or that the company has not done before?
Scale:	Do requirements specify a product larger, more complex, or requiring a larger organization than in the experience of the company?

Source: Carr et al., *Taxonomy-Based Risk Identification*