Construx®

Software Development Best Practices

Starting a Measurement Program

by Pamela Perrott, Construx

http://www.construx.com

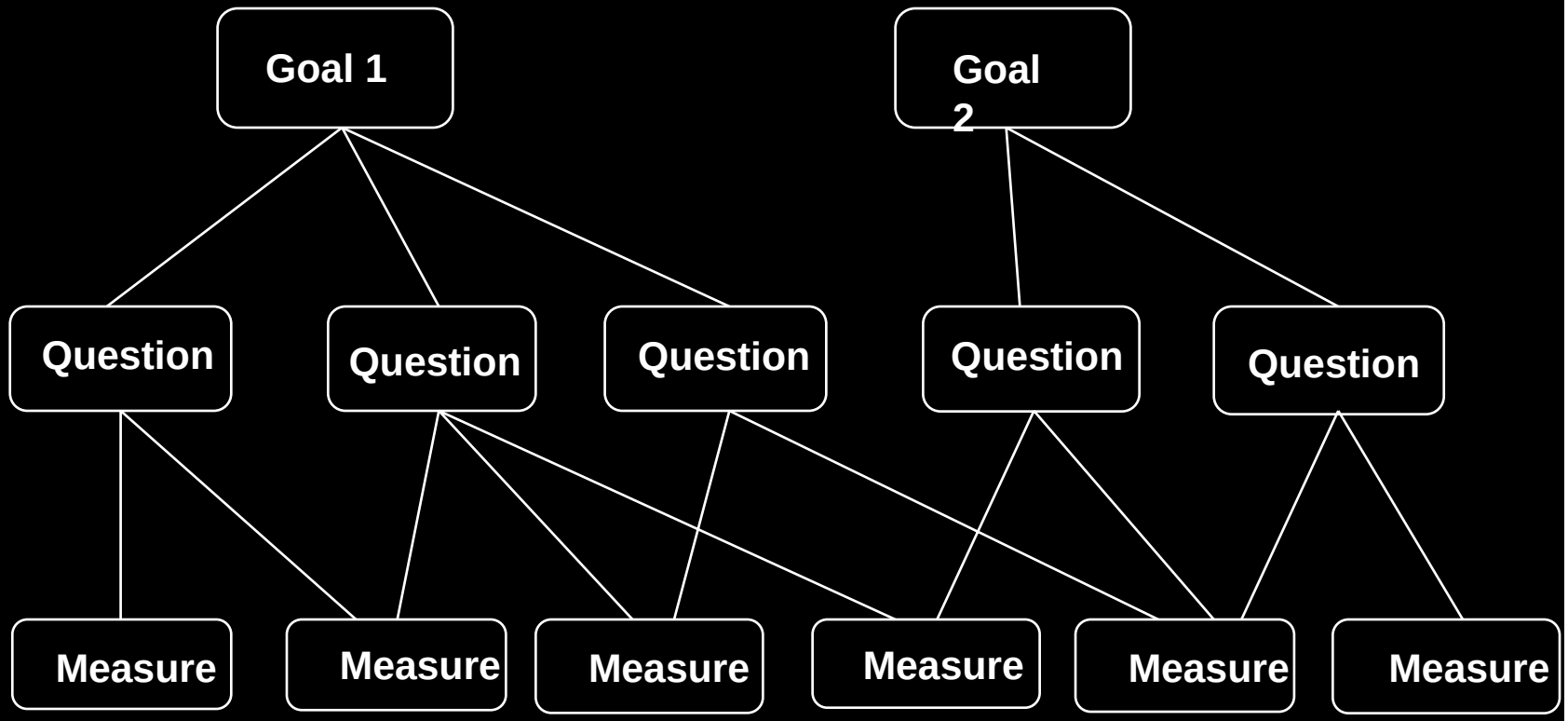
What Should We Measure?

- The real question is not *what* to measure, but *why* to measure, that will help determine what to measure
  - Determine the goals
  - Identify what questions we need to ask to know if we are meeting the goal
  - Create measures that will answer our question

Good Goals

- A goal should be SMART
  - Specific
  - Measurable/Testable
  - Attainable
  - Relevant
  - Time-bound
- Can use a *Purpose, Issue, Object* format

GQM Hierarchy



GQM Example

Goal	Purpose Issue Object (process) Viewpoint	Improve by 10% the timeliness of change request processing from the project manager's viewpoint by end of year 2007
Question		What is the current change request processing speed?
Measures		Average cycle time Standard Deviation % cases outside the upper limit
Question		Is the performance of the process improving?
Measures		Current average cycle time * 100 Baseline average cycle time Subjective rating of manager's satisfaction

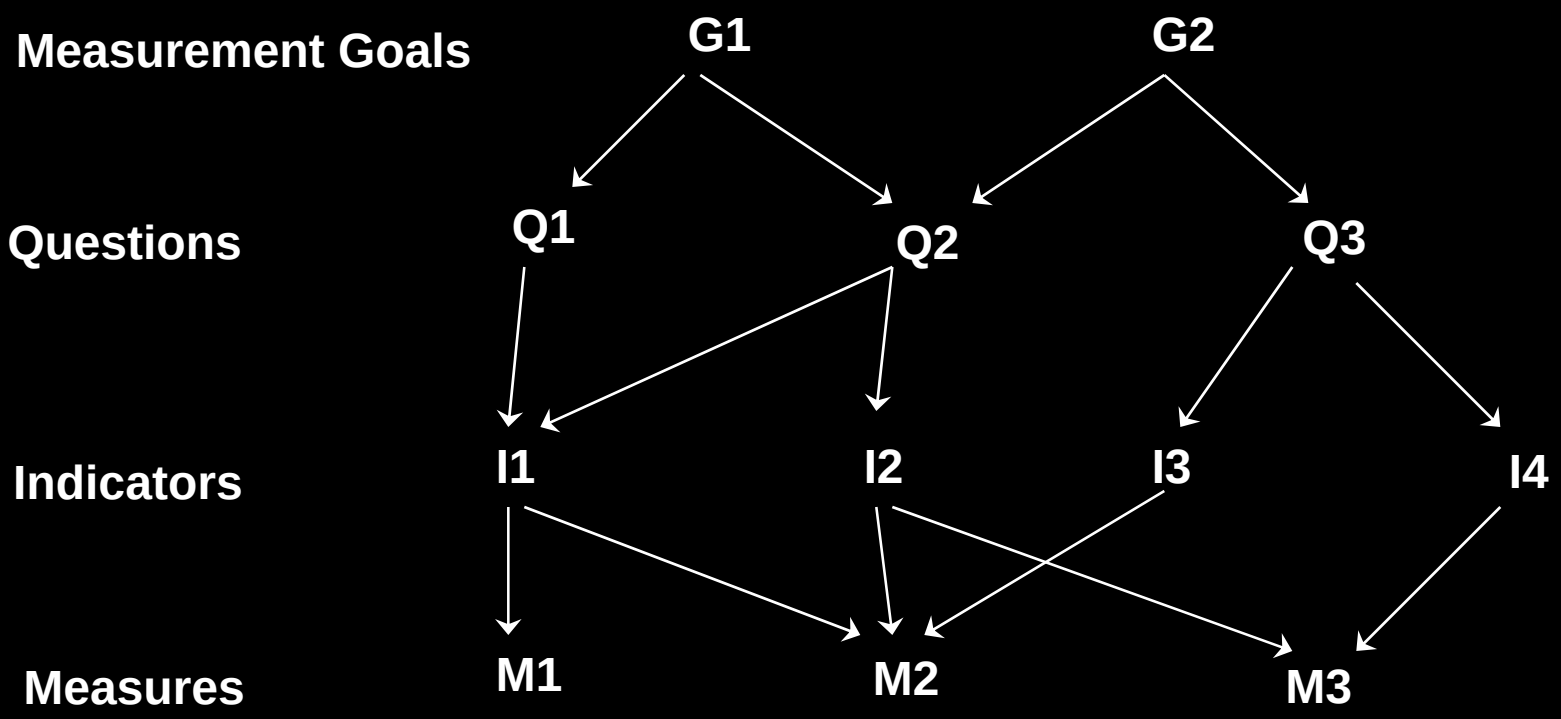
AirTouch example

- Goal: do an inspection (review) of every change to current systems to reduce defects
- Question: How many deliverables are being inspected relative to how many could have been inspected?
- Measures:
  - A) count inspections turned in
  - B) count changes moved to production
  - C) Match changes to inspections
  - D) percentage of changes with inspection

GQ (I) M

- The SEI suggests an enhancement of GQM, GQ(I)M
- The I stands for Indicator
- An Indicator is a graph or table – it's the way we will want to see our data displayed
- It helps to know what data to collect, i.e. measures, if you know what graph will be useful

GQ(I)M continued



From Robert E. Park, Wolfhart B. Goethert, William A. Florac, Goal-Driven Software Measurement – A Guidebook, SEI-96-HB-002, 1996

Construx®

Software Development Best Practices

Starting and Operating Measurement Program

Why Start a Measurement Program?

- Measuring the software process provides a rigor and understanding of what is going on
- Measurement activities and tools together lead to a greater level of sophistication in software engineering techniques
- The use of common terminology and sharing of success stories leads to more consistent use of the most effective development environments and tools
- You have a way to determine your progress

Merely Curious

- Collecting data is not the goal
- Measurement must be part of an overall strategy for software development process improvement
- This requires management support!
- Both staff and managers are afraid of being measured and will resist – you need management support for your efforts

A Suggested Process

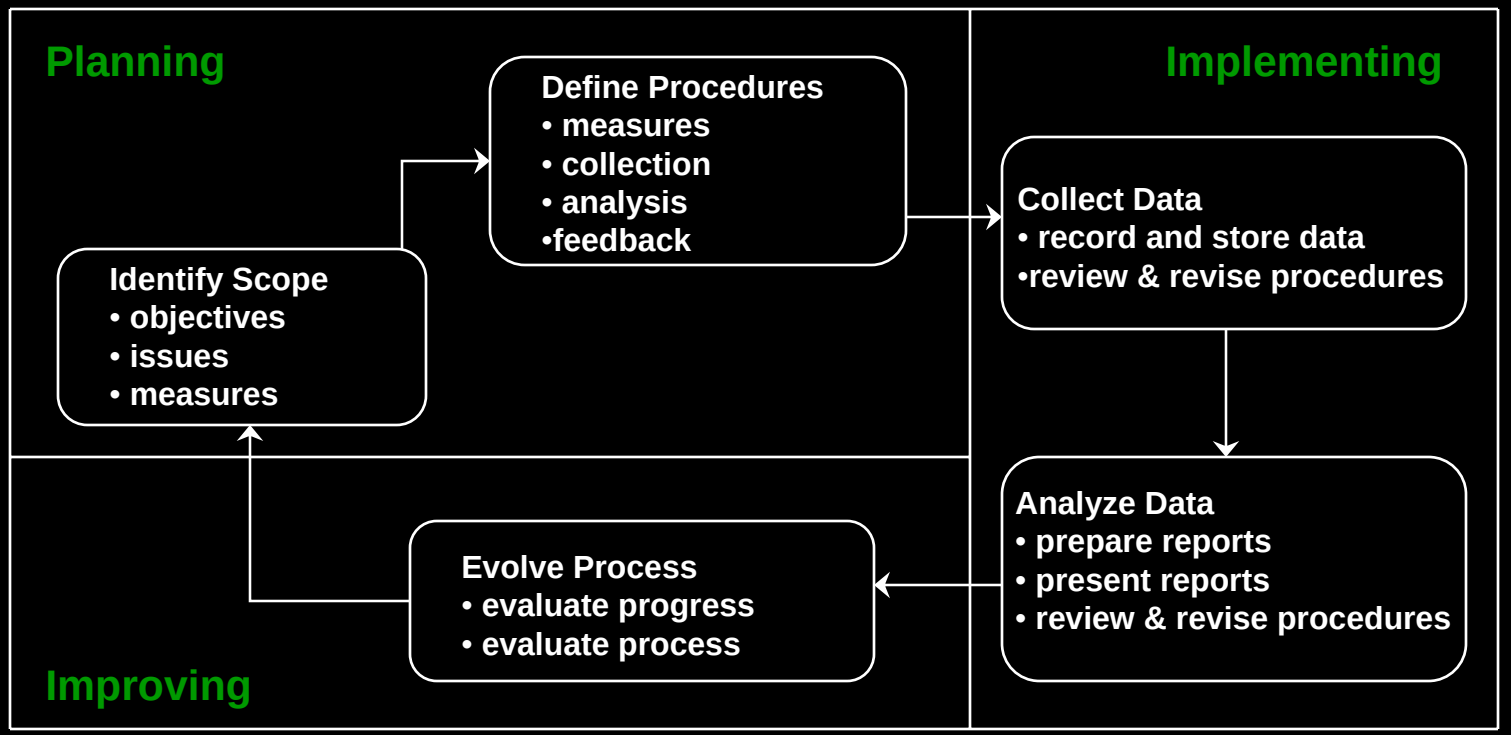
- 1. Define your goals – what does the organization want to accomplish through measurement?
- 2. Define Scope
  - Projects included
  - Phases included
  - Staff included

A Suggested Process continued

- 3. Roles, Responsibilities and Structure
  - There are usually three distinct roles
    - The source of the data
    - Analysis and packaging
    - Technical support
- 4. Define initial measurements to collect
- 5. Budget for the cost of the measurement program

A Suggested Process

Developing a Measurement Process

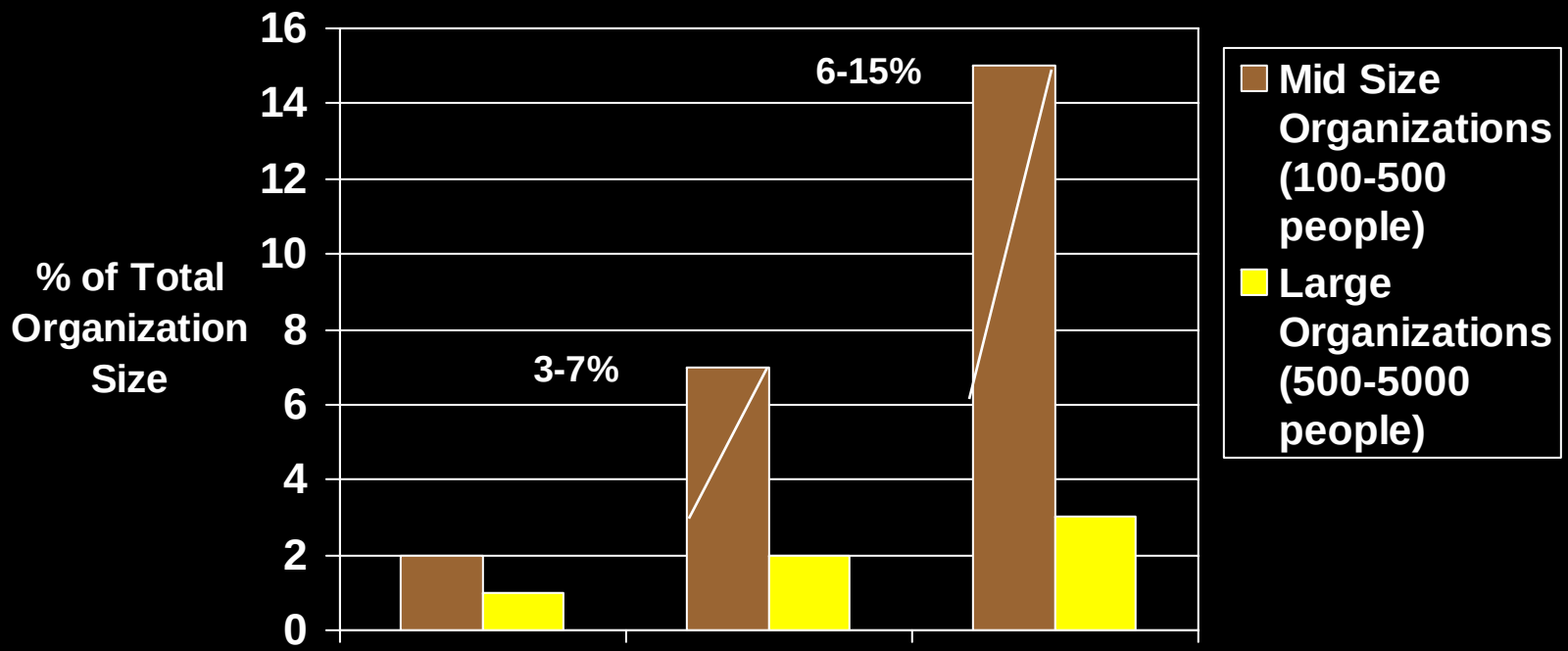


From Establishing a Software Measurement Process by Donald R. McAndrews

Cost of a Measurement Program

- Capers Jones says the cost of a measurement program is often about 4 – 5% of the software budget
  - 2 % for productivity measurements
  - 2 – 3% for quality and user satisfaction measurements
- The cost at the NASA SEL was:
  - About 2% for filling out forms and providing data
  - About 3-7% for archiving results, maintaining the database and QA
  - 6-15% for analyzing the results, training staff, defining experiments, developing models for medium sized organizations

Cost of Software Measurement



- **Source of Data**
  - •Fill out forms
  - •Provide data
- **Technical Support**
  - •Archive results
  - •Maintain database
  - •QA
- **Analysis & Packaging**
  - •Develop models (processes)
  - •Analyze results
  - •Train staff
  - •Define experiments

From NASA Software Measurement Guidebook, Revision 1, 1995

Look for Data Already Available

- Some data may already be collected
  - Total defects we've found
  - Defects fixed
  - Defects still open
  - Effort data may also be available – but may not have the level of detail desired
- Some data may be easy to automate the collection of
  - Lines of code checked into configuration management system
- Look around and see what's available now
- Don't collect it just because it's there

Need a High Level Sponsor

- For your measurement program to continue, you need a high level management sponsor
- If you lose your management sponsor, the program will probably not continue

Dedicated Resources

- Your measurement program will need dedicated resources
- People who understand software measurement
- Someone needs to analyze the data submitted for obvious defects and check back with the person submitting it
- If the organization is large, people need to be appointed in each division as measurement experts

Dedicated Resources continued

- Someone needs to run monthly reports, design the measurement database, collect data, and design graphs to display the data
- Someone needs to chair regular meetings of the measurement council or measurement divisional representatives – which considers if the measurements being collected are still what we want, should new ones be defined, etc.
- Someone needs to use the data for process improvement – to think about possible process improvements, try an improvement, test if it's better, etc.

Storing Data

- For very small starting measurement programs, something like Excel may be enough
- Most organizations will need to design a multiuser database
- May be able to buy a commercial tool
- Someone needs to revise it as the needs of the organization change
- Reporting – graphs – will be important

Construx®

Software Development Best Practices

Pitfalls and Keys to Success

It's Hard to Establish a Measurement Program

- Historically, most fail within 2 years
- Only about 20% succeed
- We've given you the tools to succeed:
  - Get management support
  - Collect only what you'll use
  - Involve members of the organization in deciding what to collect
  - Use the goals of the organization in deciding what to collect
  - Never use data for individual performance reviews
  - Don't let data disappear into a black hole

Pitfalls

- Collecting data which isn't used
- Not feeding data back to the people who collected it
- Collecting the wrong data
- Using data in performance reviews – it should be collected for process improvement and for tracking
- Using it to target individuals in any way
  - If you do they'll subvert accurate data gathering to make themselves look as good as possible
  - There may be good reasons for productivity differences
- Not having upper management support

Keys to Success

- Set solid objectives and plans
- Make measurement a part of your process and not a ‘fad of the year’
- Gain a thorough understanding of what measurement is all about, including its benefits and limitations
- Focus on the cultural issues. Involve staff in defining measures to be used so they will buy in
- Have an internal champion of the program

Keys to Success continued

- Measurement programs are successful because people allow them to succeed
- Use a GQM approach to decide what data to collect
- Create a safe environment for collecting and reporting true data (don't shoot the messenger)
- Actively use the data to find root causes
- Use a set of complementary measurements – a 'dashboard' – you don't want productivity to rise at the expense of quality

Keys to Success continued

- Start small and add more measurements over time
- Feed the data back to practitioners so they see it being used
- Automate data collection where possible
- Train practitioners on your measurement program
- Have a dedicated measurement team

NASA Lessons Learned

- Data collection requires a rigorous process and professional staff
- Compromise - ask for only as much information as is feasible to obtain
- Staff training in data collection never ends
- Data collection takes second place to deadlines
- The accuracy of the measurement data will always be suspect

NASA Lessons Learned continued

- Establishing a baseline of an organization's products, processes, and goals is critical
- Having upper management support is important for continued success
- If you lose management support the measurement program may not continue
- The organization trying to improve their process has to own the improvement process

Data Should be Visible

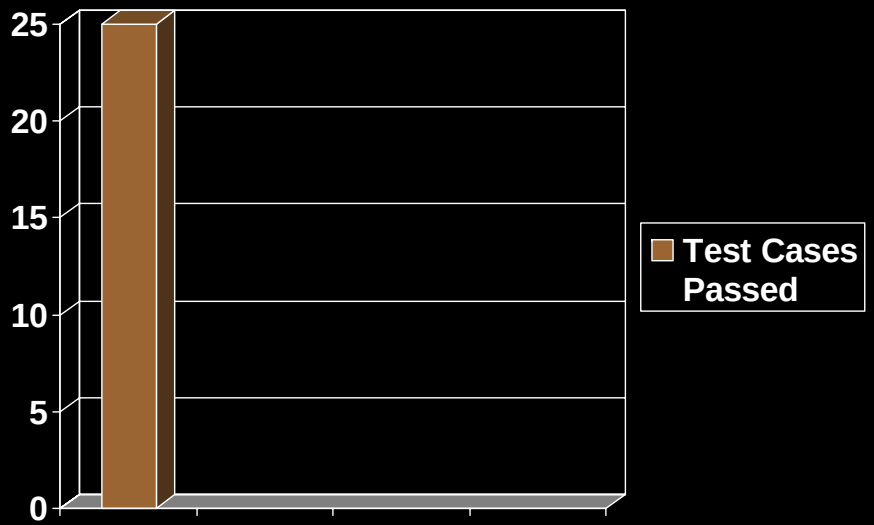
- Feed the data collected back to those who submitted it
- They'll be motivated to collect it accurately if they see how it's used
- Post the results prominently
- Put them in a findable place electronically
- Especially for sensitive data – data about productivity, or defects – post only grouped results for a project or an organization

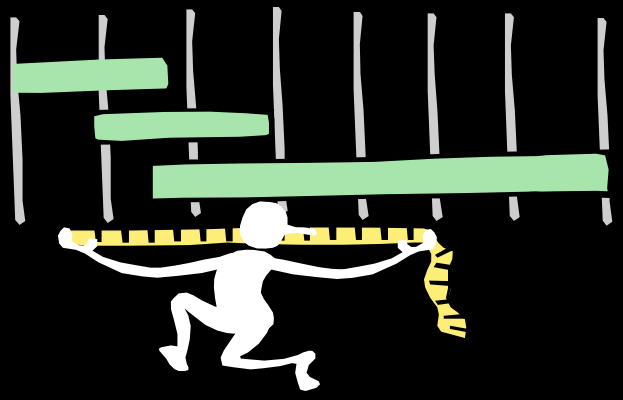
Types of Measurements

- Categorize
  - The fundamental data gathered
    - Main problem is to decide what to collect and how to define the measurement
- Evaluate
  - Look at changes, often over time
- Predict
  - When have gathered enough data, predict the future

To Characterize

- Establish a way to quantify an item of concern
  - Objectify
  - Baselines





Characterize examples

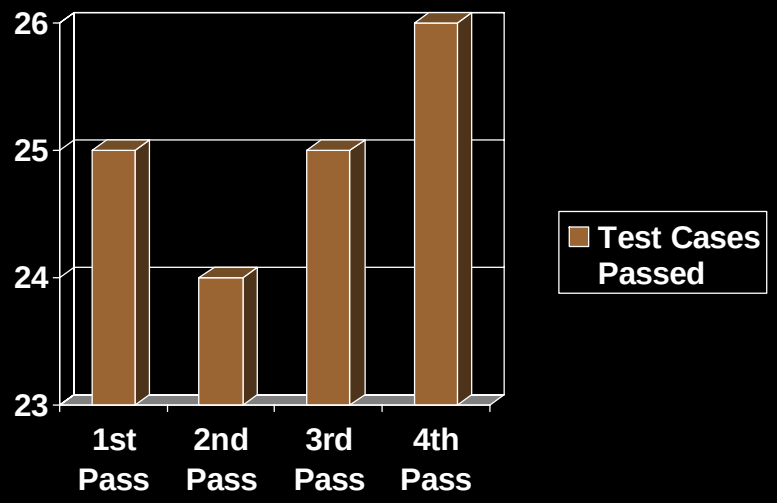
- Effort
  - Do you count overtime?
  - Does your current time tracking system force people to enter 40 hours/week no matter how many hours they worked?
  - If so, how will you collect actual effort?
  - Whose effort will be counted? Managers? Project managers? Administrative people? Testers? Business analysts? VP's?
  - When will effort tracking begin? Are early tasks like requirements development included?
  - Will you separate development and maintenance?

Characterize examples

- Defects
  - What is a defect?
  - What will you do with duplicates? With problems that can't be repeated?
  - Will you include defects found in reviews and inspections before you get to testing?
  - Will you include defects found in unit testing?
  - What about defects in user manuals and other documentation?
  - What about enhancements? Will you count enhancements as defects?
  - What if the user doesn't like it? Is that a defect?

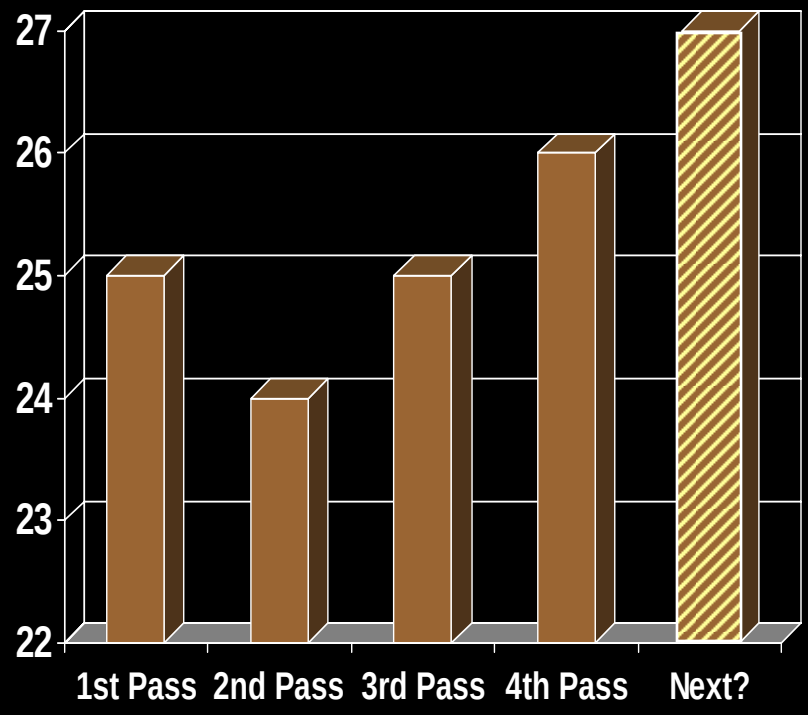
To Evaluate

- Compare characterizations against each other or against time
  - Determine status with respect to plans
  - Relationship between process & products
  - Assess the impacts of technology and process improvements
- Trends

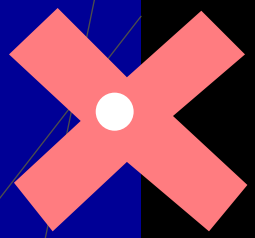


To Predict

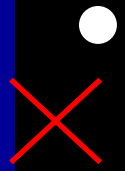
- Extrapolation of an evaluation to anticipate future measures
  - Planning
  - Set goals
  - Analyze risks
  - Design/cost tradeoffs



Precision vs. Accuracy



- Accuracy
  - Saying PI = 3 is accurate, but not precise
  - I'm 2 meters tall, which is accurate, but not precise
- Precision
  - Saying PI = 4.378383 is precise, but not accurate
  - Airline flight times are precise to the minute, but not accurate
- Number of significant digits is the key



Precision vs. Accuracy

- People make assumptions about accuracy based on precision
- “365 days” is not the same as “1 year” or “4 quarters” or even “52 weeks”
- “10,000 staff hours” is not the same as “5 staff years”
- Unwarranted precision is the enemy of accuracy (e.g., 395.7 days +/- 6 months)

Some Example Measures

- In the following slides, I've listed some example measures
  - For projects
  - For a product
  - For a process
  - That an individual might use to measure him or herself

Construx®

Software Development Best Practices

Project Measures

Section Outline

- Key Characterizations
  - Size
  - Effort
  - Time
  - Quality
- Useful Evaluations
  - Project Tracking
  - Amount of Rework
  - Earned Value
  - Retrospectives
- Typical Predictions
  - Estimates
  - Risk

Project Tracking Measures

- Time, i.e. schedule
- Effort
  - Cost can be derived from effort
- Functionality
  - A product characterization
- Quality
  - A product characterization
- Productivity
  - A process characterization
- Amount or percent rework
  - A process characterization

Construx®

Software Development Best Practices

Product Measures

Section Outline

- Key Characteristics
  - Size
  - Quality
  - Complexity
- Useful Evaluations
  - Change rate of ‘x’ over time
  - ‘X’ vs. defect rate
- Typical Predictions
  - Impact estimation
  - Remaining defects

Construx®

Software Development Best Practices

Process Measures

Section Outline

- Key Characteristics
  - Defect
  - Status
  - Compliance
- Useful Evaluations
  - Defect phase containment
  - Natural Variation
- Typical Predictions
  - Statistical Process Control
  - Capability

Construx®

Software Development Best Practices

Personal Measures

PSP Data

- What a developer measures
  - Estimated & actual size by object
  - Estimated & actual direct effort ("Time On Task") by phase
  - Estimated & actual defects, injected and removed by phase
- Metrics derived from the measurements (for starters!)
  - Productivity (LOC/Hour)
  - % Time on Task
  - Defect density (Defects/LOC)
  - Cost Of Quality (COQ) (design review time + code review time/ time in compile and test including time to fix and retest any defects found)
  - Yield (Defects found/Defects present)
  - Review Rates (LOC/hr. More than about 150-200 loc/hr results in missed defects)
  - Process Quality Index (PQI)
  - Defect Removal Leverage (DRL) – relative effectiveness of two defect removal phases, for example design reviews/unit test

A Few Suggested Measures

- Requirements
  - What percent of changes were
    - The business changed; we couldn't have known about it in advance
    - We got it wrong
  - Measures how good your requirements are
- Defect rework percentage of total effort
  - In the absence of Inspections, about 30-80 percent of total effort is rework
  - It's the single biggest productivity improvement
- Earned Value or Velocity
  - Gets planning and tracking under control

Contact Information

Consulting@construx.com

www.construx.com

(425) 636-0100

Construx

Delivering Software Project Success

Consulting

Seminars

sales@construx.com

www.construx.com