

# Quality

SE423: Software Project Management

# Outline

- Software Quality
- Quality Assurance
- Quality Control
- Project Success
- Measurement and Metrics

# Software Quality

# Software Quality

- A definition of quality should emphasize three important points:
  1. Software requirements are the foundation from which quality is measured. Lack of conformance to requirement is lack of quality.
  2. Specified standards define a set of development criteria that guide the manner in which software is engineered. If the criteria are not followed, lack of quality will almost surely result.
  3. There is a set of implicit requirements that often goes unmentioned (e.g. good maintainability). If software conforms to its explicit requirements but fails to meet implicit requirements, software quality is suspect.

# Software Testing

- The purpose of software testing is to assess and evaluate the quality of work performed at each step of the software development process.
- Although it sometimes seems that way, the purpose of testing is NOT to use up all the remaining budget or schedule resources at the end of a development effort.
- The goal of testing is to ensure that the software performs as intended, and to improve software quality, reliability and maintainability.

*Software testing is a full-life-cycle assessment of quality*

# Quality and Testing

- A good development process, tools, methods, and people go far in providing quality products
- Testing is one aspect of assuring software quality
  - It is a measure of quality, it does not deliver quality
- “Quality cannot be tested into a product”
- Software Quality Assurance includes
  - Software engineering process improvement
    - Prevent the insertion of defects
  - Fault tolerant software design
    - Tolerate the existence of defects
  - All aspects of software verification and validation
    - Including testing

# Quality Assurance

# Goal of Quality Assurance

- Quality assurance (QA) activities strive to ensure:
  - Few, if any, defects remain in the software system when it is delivered
  - Remaining defects will cause minimal disruptions or damages

# Planning Quality

- The following need to be considered: Scope, Stakeholders, Risks, Internal and External Environmental Factors, Process
- Project-specific standards and procedures are created
  - Based on quality standards for each deliverable
  - Includes how PM activities themselves should be done
  - Plans/Project must comply with external standards (CISG, ISO 9000, OSHA, etc)
  - Plans/Project must comply with organizational standards
  - Plans/Project must meet the customer's quality standards
  - Tracking / Proof may be needed (metrics, measurements, etc.)

# Quality Control

# Perform quality control

*Perform quality control* is concerned with monitoring specific project results for compliance with quality standards

- Performed throughout project
- May also include taking control actions to correct causes of quality problems

*Perform quality assurance* is the process that provides the framework of activities and standards for performing quality control

# Software Quality Assurance Group

- The group should prepare an SQA plan for a project.
- The plan identifies
  - evaluations to be performed
  - audits and reviews to be performed
  - standards that are applicable to the project
  - procedures for error reporting and tracking
  - procedures for change management
  - documents to be produced by the SQA group
  - amount of feedback provided to the software project team

# Software Quality Assurance Group

- Participates in the development of the project's software process description
- Reviews software engineering activities to verify compliance with the defined software process.
- Audits designated software work products to verify compliance with those defined as part of the software process.
- Ensures that deviations in software work and work products are documented and handled according to a documented procedure.
- Records any noncompliance and reports to senior management.

# Project Success

# Think Small

- Keep requirements tight & focused
- One milestone at a time
- Smaller, incremental chunks
- As simple as possible but no simpler

# Process Spectrum

Too much medicine can kill the patient

Process  
Spectrum



Chaos

Bureaucracy

Balance is crucial

# Miscellaneous

## Be Disciplined and Professional

- Learn to say “No”
  - Be polite but firm
- The Value of Versions
  - “We will put that in phase 2”
- An Ounce of Prevention

## Paralysis

- Analysis Paralysis
  - Over-process
  - Nothing gets finished
  - 65% of software professionals have experienced this
- Paralysis Paranoia
  - Fear of over-process = process avoidance

# Miscellaneous

- MBWA – Management by Walk About
  - Shows you're actually involved day-to-day
  - Recognizes individuals may say more 1-on-1
  - Allows spontaneity
  - Finds personnel problems sooner
- Delegate
  - Don't be a "Control Freak"
  - You need to be the "hub" but not everything
- Project Home Page
  - Give your project an intranet page
  - Central repository for project status, documents and other resources

# Success Metrics

1. On schedule
  - Requires good: plan; estimation; control
2. Within budget
  - Again: planning, estimation & control
3. According to requirements
  - Importance of good requirements
  - Perception & negotiation critical
4. High quality. May or may not be same as item 3

**Only real measure:**

**Is the customer happy?**

Customer satisfaction!!

# Why Do Projects Succeed?

- How to identify a project's success potential
  - What metrics could you look at?
    - Project size
    - Project duration
    - Project team size

# Why Do Projects Succeed?

- **Executive support**
- User involvement
- Experienced project manager
- Clear business objectives
- Minimized scope
- Standard software infrastructure
- Firm basic requirements
- Formal methodology
- Reliable estimates

# Why Executive Support?

- Top management can help to:
  - Secure adequate resources
  - Get approval for unique project needs in a timely manner
  - Receive cooperation from people throughout the organization
  - Provide leadership guidance

# State of the Practice in Software Management

- Factors that may influence the success or failure of the software projects could be:
  1. Social Factors
  2. Technology

# State of the Practice in Software Management

## Technologies on Unsuccessful Projects

- No historical software measurement data
- Failure to use automated estimating tool
- Failure to use automated planning tool
- Failure to monitor progress or milestones
- Failure to use effective architecture
- Failure to use effective development methods
- Failure to use design reviews
- Failure to use code inspections
- Failure to include formal risk management
- Informal, inadequate testing
- Manual design and specification
- More than 30% creep in user requirements

## Technologies on Successful Projects

- Accurate software measurement
- Early use of estimating tools
- Continuous use of planning tool
- Formal progress reporting
- Formal architecture planning
- Formal development methods
- Formal design reviews
- Formal code inspections
- Formal risk management
- Formal testing methods
- Automated design and specification
- Automated configuration control
- Less than 10% creep in requirements

# State of the Practice in Software Management

## Social Factors on Unsuccessful Projects

- Excessive schedule pressure
- Executive rejection of estimates
- Severe friction with clients
- Divisive corporate politics
- Poor team communications
- Naïve senior executives
- Project management malpractice
- Unqualified technical staff
- Generalists used for critical tasks: quality assurance, testing, planning, estimating

## Social factors on Successful Projects

- Realistic schedule pressure
- Executive understanding of estimates
- Cooperation with clients
- Congruent management goals
- Excellent team communications
- Experienced senior executives
- Capable Project management
- Capable technical staff
- Specialists used for critical tasks: quality assurance, testing, planning, estimating

# How to ensure a project fails

- Do the same things you did on the last project. Over and over and over.
- Don't listen to your experts. After all the last project worked out okay (mostly)
- Don't measure progress with metrics. The only thing that counts is "did you meet the delivery date?".
- Set delivery dates with the customer but not with the developers. Developers can meet any schedule we ask for.
- Don't use new tools. Keep using the ones we used ten, twenty years ago.
- Spend your time making sure people do it your way.
- Office politics and vendettas are more important than the project.

# Measurement and Metrics

# Measurement and Metrics

- “Not everything that can be counted counts, and not everything that counts can be counted.”  
- Albert Einstein

# Why Measurement and Metrics?

- Software measurement is concerned with deriving a quantitative (numeric) value for an attribute of a software product or process (largely qualitative)
- This allows for objective comparisons between techniques and processes
- Although some companies have introduced measurement programs, the systematic use of measurement is still uncommon
- There are few standards in this area

# Definitions

- **Measure**
  - provides a quantitative indication of the size of some product or process attribute
- **Measurement**
  - the act of obtaining a measure
- **Metric**
  - a quantitative measure of the degree to which a system, component, or process possesses a given attribute

# Software Metric

- Any type of measurement which relates to a software system, process or related documentation
  - Lines of code in a program, number of person-days required to develop a component
- Allow the software and the software process to be quantified
- Measures of the software process or product
- May be used to predict product attributes or to control the software process

# Categories

- Product
  - Assess the quality of the design and construction of the software product being built.
- Process & Project
  - Quantitative measures that enable software engineers to gain insight into the efficiency of the software process and the projects conducted using the process framework

# Process Metrics

- Private process metrics (e.g., defect rates by individual or module) are only known to by the individual or team concerned.
- Public process metrics enable organizations to make strategic changes to improve the software process.
- Metrics should not be used to evaluate the performance of individuals.
- Statistical software process improvement helps and organization to discover where they are strong and where they are weak

# Product Metrics

- A quality metric should be a predictor of product quality
- Classes of product metric
  - Dynamic metrics which are collected by measurements made of a program in execution
  - Static metrics which are collected by measurements made of the system representations
  - Dynamic metrics help assess efficiency and reliability; Static metrics help assess complexity, understandability and maintainability

# Project Metrics

- A software team can use software project metrics to adapt project workflow and technical activities
- Project metrics are used to avoid development schedule delays, to mitigate potential risks, and to assess product quality on an on-going basis
- Every project should measure its inputs (resources), outputs (deliverables), and results (effectiveness of deliverables)

# Metrics Assumptions

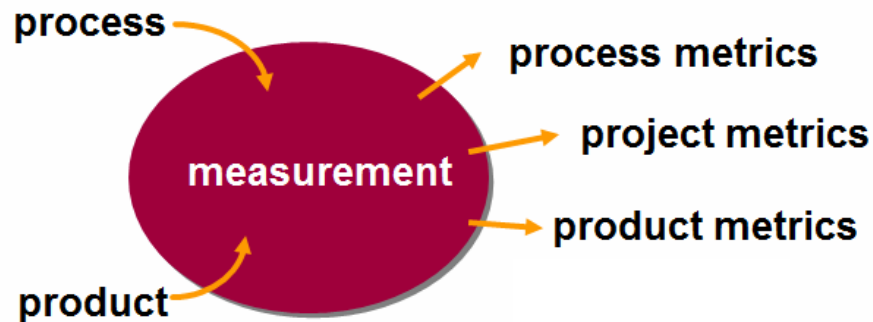
- A software property can be measured
- The relationship exists between what we can measure and what we want to know
- This relationship has been formalized and validated
- It may be difficult to relate what can be measured to desirable quality attributes

# Integrating Metrics with Process

- Many software developers do not collect measures.
- Without measurement it is impossible to determine whether a process is improving or not
- Baseline metrics data should be collected from a large, representative sampling of past software projects
- Getting this historic project data is very difficult, if the previous developers did not collect data in an on-going manner

# Software Measurement

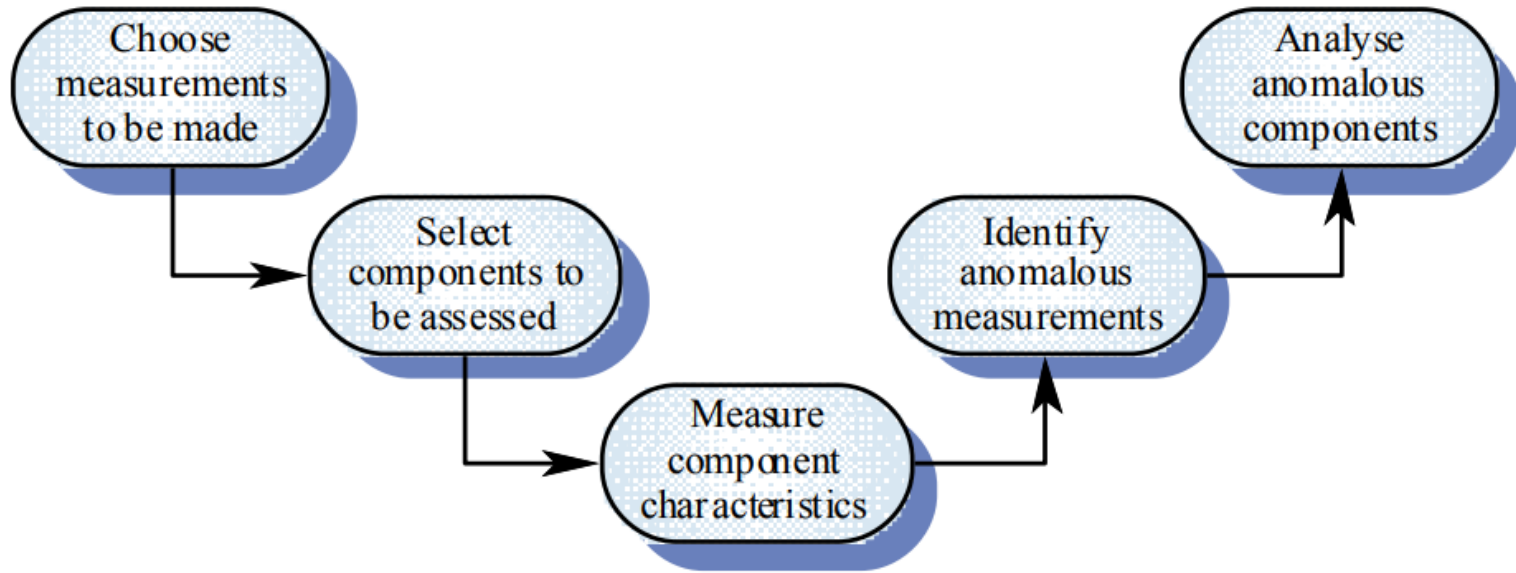
- Direct measures of a software engineering process include cost and effort
- Direct measures of the product include lines of code (LOC), execution speed, memory size, defects reported over some time period
- Indirect product measures examine the quality of the software product itself (e.g., functionality, complexity, efficiency, reliability, maintainability)



# The Measurement Process

- A software measurement process may be part of a quality control process
- Data collected during this process should be maintained as an organizational resource
- Once a measurement database has been established, comparisons across projects become possible

# Product Measurement Process



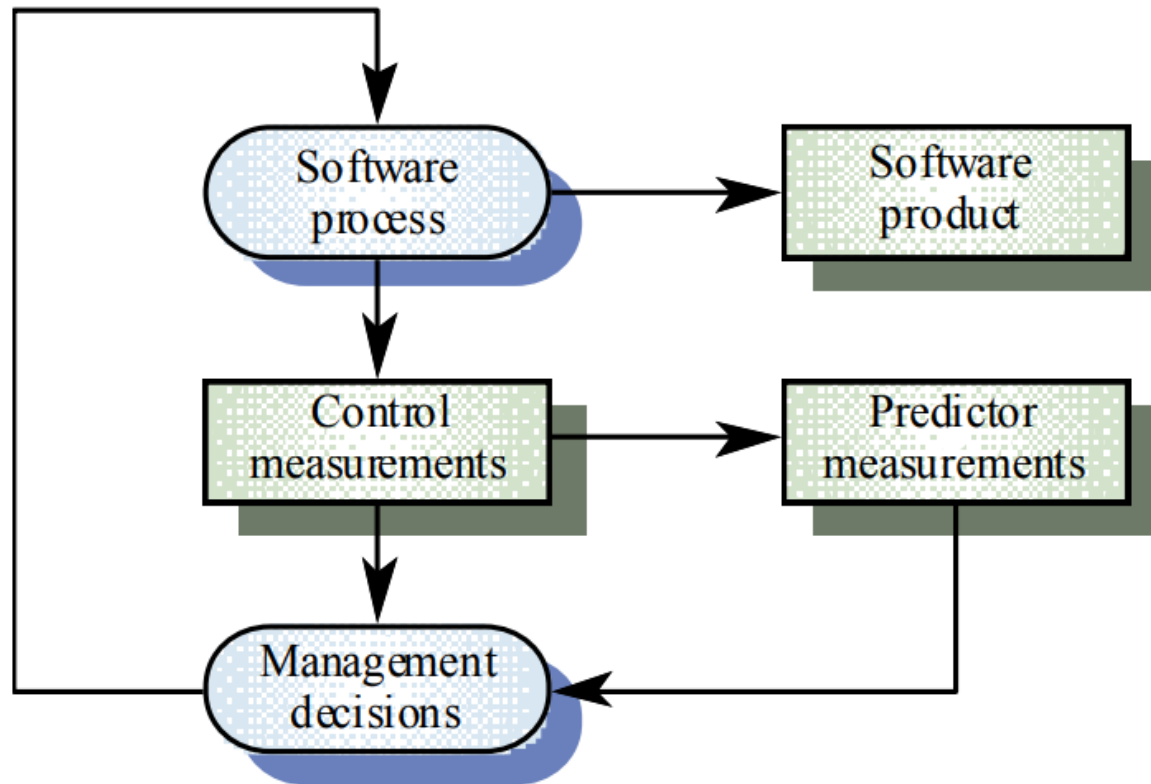
# Data Collection

- A metrics program should be based on a set of product and process data
- Data should be collected immediately (not in retrospect) and, if possible, automatically
- Three types of automatic data collection
  - Static product analysis
  - Dynamic product analysis
  - Process data collation

# Data Accuracy

- Don't collect unnecessary data
  - The questions to be answered should be decided in advance and the required data identified
- Tell people why the data is being collected
  - It should not be part of personnel evaluation
- Don't rely on memory
  - Collect data when it is generated not after a project has finished

# What Are Metrics For?



# Analysis

- It is not always obvious what data means
  - Analyzing collected data is very difficult
- Professional statisticians should be consulted if available
- Data analysis must take local circumstances into account

# Metric Types with Examples

- Size-oriented metrics
  - Defects, human effort, Line of Code LOC
- Function-oriented metrics
  - Function points
- Web Engineering
  - Number of static Web pages ( $N_{sp}$ ), Number of dynamic Web pages ( $N_{dp}$ ), Customization index:  $C = N_{sp} / (N_{dp} + N_{sp})$
- Product metrics
  - Cyclomatic complexity, Fan in/Fan-out