Lecture 27

Quality Management

- Quality concepts
- Software quality assurance
- Software reviews
- Statistical software quality assurance
- Software reliability, availability, and safety
- SQA plan

Statistical Software Quality Assurance

Process Steps

1) Collect and categorize information (i.e., causes) about software defects that occur

2) Attempt to trace each defect to its underlying cause (e.g., nonconformance to specifications, design error, violation of standards, poor communication with the customer)

3) Using the Pareto principle (80% of defects can be traced to 20% of all causes), isolate the 20%

A Sample of Possible Causes for Defects

- Incomplete or erroneous specifications
- Misinterpretation of customer communication
- Intentional deviation from specifications
- Violation of programming standards
- Errors in data representation
- Inconsistent component interface
- Errors in design logic
- Incomplete or erroneous testing
- Inaccurate or incomplete documentation
- Errors in programming language translation of design
- Ambiguous or inconsistent human/computer interface

Six Sigma

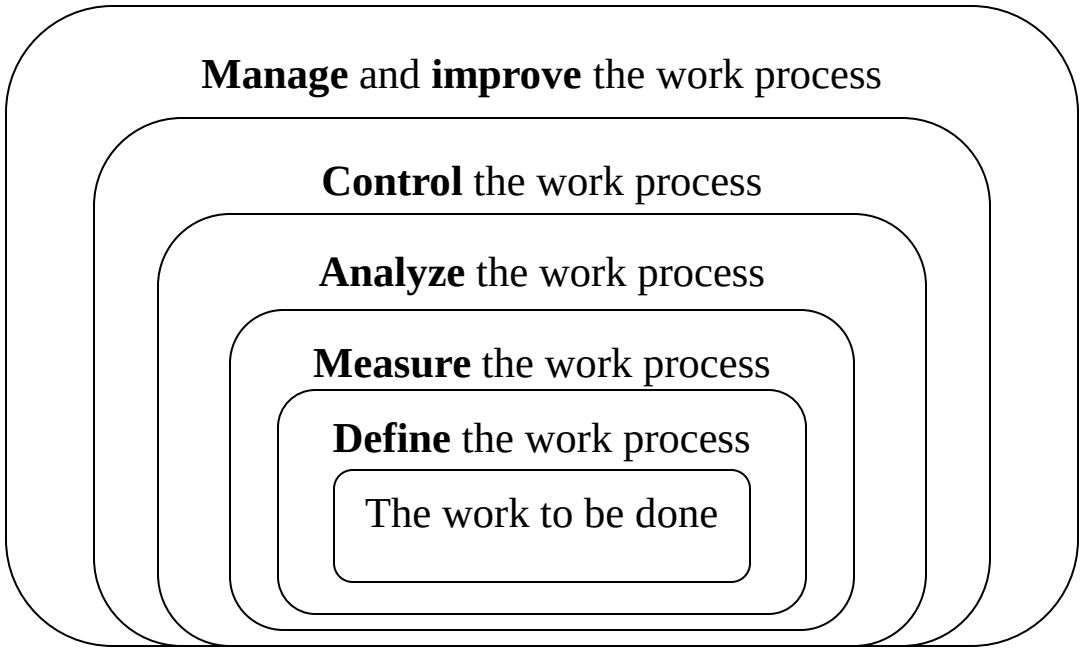
- Popularized by Motorola in the 1980s
- Is the most widely used strategy for statistical quality assurance
- Uses data and statistical analysis to measure and improve a company's operational performance
- Identifies and eliminates defects in manufacturing and service-related processes
- The "Six Sigma" refers to six standard deviations (3.4 defects per a million occurrences)

Six Sigma (continued)

- **Three core steps**
  - Define customer requirements, deliverables, and project goals via well-defined methods of customer communication
  - Measure the existing process and its output to determine current quality performance (collect defect metrics)
  - Analyze defect metrics and determine the vital few causes (the 20%)
- Two additional steps are added for existing processes (and can be done in parallel)
  - Improve the process by eliminating the root causes of defects
  - Control the process to ensure that future work does not reintroduce the causes of defects

Six Sigma (continued)

- All of these steps need to be performed so that you can manage the process to accomplish something
- You cannot effectively manage and improve a process until you first do these steps (in this order):



ISO 9000 Quality Standards

- ISO 9000 describes quality assurance elements in generic terms that can be applied to any business.
- It treats an enterprise as a network of interconnected processes.
- To be ISO-complaint processes should adhere to the standards described.
- Elements include organizational structure, procedures, processes and resources.
- Ensures quality planning, quality control, quality assurance and quality improvement.

ISO 9001

- An international standard which provides broad guidance to software developers on how to Implement, maintain and improve a quality software system capable of ensuring high quality software
- Consists of 20 requirements…
- Differs from country to country..

ISO 9001 (cont'd)..requirements

- Management responsibility
- Quality system
- Contract review
- Design Control
- Document and data control
- Purchasing
- Control of customer supplied product
- Product identification and traceability
- Process control
- Inspection and testing
- Control of inspection, measuring and test equipment

ISO 9001 (cont'd)..

- Inspection and test status
- Control of non-confirming product
- Corrective and preventive action
- Handling, storage, packaging, preservation and delivery
- Control of quality records
- Internal quality audits
- Training
- Servicing
- Statistical techniques

Software Reliability

- Defined as the probability of failure free operation of a computer program in a specified environment for a specified time.
- It can measured, directed and estimated
- A measure of software reliability is *mean time between failures* where
- MTBF = MTTF + MTTR
- MTTF = *mean time to failure*
- MTTR = *mean time to repair*

Software Availability

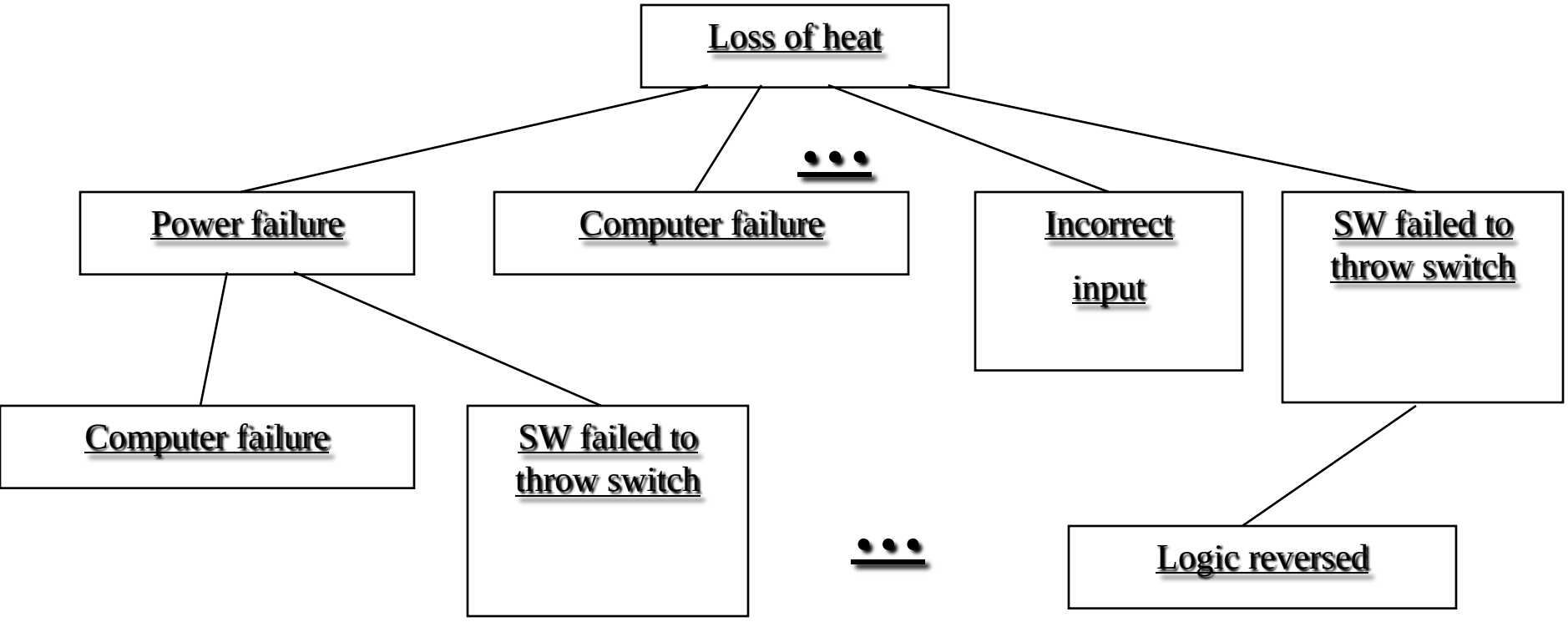
- Availability =MTTF/(MTTF + MTTR) * 100%
- *Software availability* is the probability that a program is operating according to requirements at a given point in time

Software Safety

- Processes that help reduce the probability that critical failures will occur due to SW

↗ Hazard analyses
↗ Identify hazards that could call failure
↗ Develop fault tree
↗ Identify all possible causes of the hazard
↗ Formally review the remedy for each
↗ Redundancy
↗ Require a written software safety plan
↗ Require independent verification & validation

Example Fault Tree -- Thermal



Software Safety

- Redundancy
  - Replicated at the hardware level
  - Similar vs.. dis-similar redundancy

↗ **Verification**

↗ **Assuring that the software specifications are met**

↗ **Validation**

↗ **Assuring that the product functions as desired**

↗ **Independence**

SQA Plan

Purpose and Layout

- Provides a road map for instituting software quality assurance in an organization
- Developed by the SQA group to serve as a template for SQA activities that are instituted for each software project in an organization
- Structured as follows:
  - The purpose and scope of the plan
  - A description of all software engineering work products that fall within the purview of SQA
  - All applicable standards and practices that are applied during the software process
  - SQA actions and tasks (including reviews and audits) and their placement throughout the software process
  - The tools and methods that support SQA actions and tasks
  - Methods for assembling, safeguarding, and maintaining all SQA-related records
  - Organizational roles and responsibilities relative to product quality