RISK AND RESILIENCE BOOTCAMP

TEKsystems Global Services

WORKFORCE DEVELOPMENT

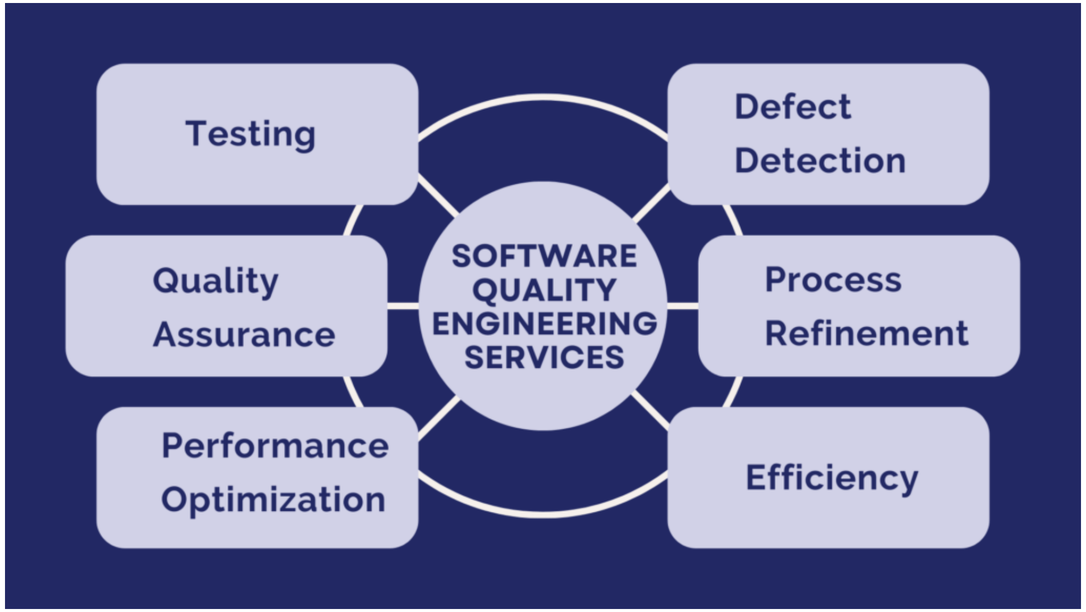
QUALITY ENGINEERING AND RISK TESTING

This section is an introduction to

- Software Quality Engineering (SQE)
- Risk Testing

This will not be a deep dive into the general topics of quality or testing

- We will just focus on aspects of these topics that are important in understanding QE and risk testing



DEFINITIONS

- Defining quality
  - Defining exactly what quality is has been a topic of debate for centuries
  - The definition used today expresses quality in terms of quality attributes
    - Every product has features
    - Quality is assigned to a feature by stakeholders based on their requirements
    - Quality is a subjective decision
  - For example, a cell phone plan has unlimited calling to Europe
    - Whether or not this improves the quality of the plan depends on whether or not a stakeholder knows and talks to people in Europe
    - The quality is not a property of the product but rather how well it meets the quality requirements of the stakeholders

DEFINITIONS

- The following concepts are often confused
  - There is an overlap in their areas of responsibility
  - But each has a specific focus and objective
    - Quality control
    - Quality assurance
    - Quality engineering
    - Reliability engineering

DEFINITIONS

- Quality control (QC)
  - QC is a product-oriented activity that involves testing, inspection, and validation to verify that deliverables meet quality specifications
  - Scope
    - Focused on identifying and correcting defects in outputs
    - These include: code, configurations, components, or operational deliverables
    - Reactive in nature: detects problems after creation but before release or operation
    - Filters out defective products as a last stage in production
  - Common techniques
    - Functional testing, integration testing, performance benchmarking
    - Inspection and validation of production releases
    - Regression and acceptance testing
  - Analogy
    - QC is like inspecting a finished bridge for cracks or misaligned bolts before cars drive across

DEFINITIONS

- Quality assurance (QA)
  - QA is a process-oriented discipline ensuring that methods, procedures, and standards are followed correctly so that the resulting product or service is likely to meet requirements
  - Scope
    - QA focuses on process conformance
    - Verifying that defined policies, design reviews, and documentation steps are executed
    - Typical outputs: checklists, audit results, process capability assessments.
  - Risk and resilience
    - QA ensures governance controls exist and are followed
    - Reduces operational risk by ensuring process discipline before incidents occur
  - Analogy
    - QA is like ensuring the bridge was designed and built according to engineering codes and safety regulations

DEFINITIONS

- Quality engineering (QE)
  - QE discipline of integrating quality practices, automation, and data-driven validation across the entire system lifecycle to prevent defects and ensure reliability and resilience
    - One of the key characteristics is that QE deals with computational models of system performance and behavior
  - Scope
    - Encompasses quality assurance, control, and reliability
    - Embeds continuous feedback loops, risk awareness, and automation
    - For example: CI/CD pipelines, observability
    - Combines software, systems, and operational perspectives
    - Especially relevant in DevOps, SRE, and regulated financial IT environments
  - Key practices
    - Defines quality models according to industry standards
    - Automates quality gates for attributes like security, performance, code analysis
    - Implements risk-based testing focusing on high-impact scenarios
    - Integrates observability metrics like availability, latency, MTTR, SLA adherence

DEFINITIONS

- Quality engineering (QE)
  - Risk and resilience
    - QE aligns directly with preventive risk management
    - Ensures processes and systems are built to withstand failure
    - Improves operational resilience through predictable performance, early defect detection, and continuous monitoring
  - Non-functional requirements
    - Primarily used to engineer the non-functional performance properties of a system
    - These properties tend to be common across *all* systems, irrespective of their functionality
    - Stress, load, throughput, response time, transaction latency, mean time to failure, etc.
  - Because of the common context for operational characteristics
    - There are standard models, processes, statistical and mathematical models used in QE
    - Forces these properties to be rigorously quantified during development
  - Analogy
    - QE is like building a bridge with sensors, predictive maintenance, and self-healing materials
    - Quality by design

DEFINITIONS

- Reliability engineering (RE)
  - A common distinction between QE and RE is
    - QE defines and quantifies performance characteristics that meet the requirements
    - RE focuses on analyzing, predicting, and improving system performance over time
    - Goal is to minimize failures and ensure consistent availability under operational stress
  - Scope
    - Studies how and why systems fail
    - Uses statistical models, fault analysis, and redundancy design
    - Concerned with probability of failure-free operation over a specified time
  - Risk and resilience
    - Enhances resilience engineering, focusing on maintaining function despite component failures
    - Transforms resilience from reactive recovery to predictive, measurable dependability

DEFINITIONS

- Reliability engineering (RE)
  - Key techniques
    - FMEA (Failure Modes and Effects Analysis)
    - Fault Tree Analysis (FTA)
    - Reliability Growth Modeling
    - Stress and Endurance Testing
    - Resilience Engineering (study of adaptive capacity in complex systems)
  - Standards
    - IEEE 1413 (Reliability Predictions)
    - MIL-STD-1629A (FMEA)
    - ISO 9001 and ISO 25010 reliability criteria
    - NIST SP 800-160 Vol.2 (Systems Security and Resilient Engineering)
  - Analogy
    - Calculating how long the bridge will last under different loads and ensuring it can withstand storms, traffic surges, or material fatigue

COMPARISON

Discipline	Core Question	Focus Area
Quality Engineering (QE)	"How do we *build quality and resilience in* from the start?"	End-to-end systems and process design
Quality Assurance (QA)	"Are we *following the right processes* to ensure quality?"	Process compliance and prevention
Quality Control (QC)	"Are the *outputs* defect-free?"	Product testing and inspection
Reliability Engineering (RE)	"Will it *keep performing reliably over time* under real-world stress?"	Long-term system performance and failure analysis

QUALITY ENGINEERING

- Quality vs. Reliability
  - Quality
    - Conformance to requirements, customer expectations, and specifications
  - Reliability
    - The probability that a system will perform its intended function without failure over a specified period
- Quality engineering works with
  - Quality planning
    - Defining quality goals, metrics, and acceptance criteria
  - Quality assurance
    - Establishing processes and standards to prevent defects
  - Quality control
    - Monitoring outputs and detecting defects
  - Continuous improvement
    - Using analytics and feedback to refine processes

LIFECYCLE INTEGRATION

Phase	QE Activities	Resilience Link
Requirements / Design	Define service-level objectives (SLOs), security and compliance criteria.	Embeds resilience targets early (e.g., RTO, RPO).
Development	Automated testing, code analysis, security testing.	Builds prevention into software.
Deployment	Configuration validation, release gates, automated approvals.	Prevents unstable releases that reduce availability.
Operations	Monitoring, feedback loops, incident reviews.	Continuous quality + resilience metrics.

ISO/IEC 25010: SOFTWARE QUALITY ITEMS

Category	Examples	Resilience Implication
Functional Suitability	Accuracy, completeness	Reliable business transactions
Performance Efficiency	Response time, throughput	Service continuity under load
Compatibility	Interoperability	Resilient integration with partners and cloud
Usability	Accessibility, learnability	Smooth user recovery during incidents
Reliability	Fault tolerance, recoverability	Directly tied to resilience and uptime
Security	Integrity, confidentiality, accountability	Mitigates cyber and data risks
Maintainability	Modularity, testability	Faster recovery and adaptive changes
Portability	Adaptability, replaceability	Cloud / hybrid migration resilience

QUALITY METRICS

Category	Common Metrics	Purpose
Reliability / Stability	MTBF (Mean Time Between Failures), Uptime %, Error Rates	Measure operational resilience
Performance	Response Time, Throughput, Resource Utilization	Validate service performance under load
Maintainability	MTTR (Mean Time To Repair), Change Success Rate	Assess recovery efficiency
Security / Integrity	Vulnerability Density, Incident Frequency	Monitor risk control effectiveness
Customer Impact	SLA Breaches, User Complaints, CSAT	Reflect quality perception and trust
Process Quality	Defect Escape Rate, Test Coverage, Automation %	Gauge maturity of engineering process

RISK TESTING

- Risk testing process
  - Identify risks
    - From risk register, past incidents, threat modeling, and BIAs
  - Prioritize scenarios
    - Based on impact and likelihood
  - Design test cases
    - Simulate control failure or degradation
    - What happens if a patch fails? backup corrupts?
  - Execute tests
    - Controlled or sandboxed environments, or live “chaos” experiments
  - Document results
    - Capture effectiveness of controls and response time
  - Improve controls
    - Feed lessons into SOPs and process maturity models

TYPES OF RISK TESTING

Type	Description	Resilience Application
Functional Risk Testing	Tests that validate business-critical workflows.	Confirms mission-critical services (e.g., payments) remain functional.
Performance / Load Testing	Simulates user or transaction load.	Ensures capacity and response within tolerance under peak conditions.
Stress Testing	Pushes systems beyond limits to see where failure occurs.	Establishes resilience threshold (tipping point).
Security Testing (Penetration / Vulnerability)	Identifies exploitable weaknesses.	Reduces cyber risk and operational exposure.
Failover / Recovery Testing	Tests backup systems and DR procedures.	Confirms recovery time (RTO) and data integrity (RPO).
Chaos Engineering	Intentionally injects faults in production to test resilience.	Builds “resilience by design” mindset (Netflix pioneered).
Regulatory / Compliance Testing	Verifies controls required by standards (e.g., SOX, PCI-DSS).	Ensures compliance even during disruptive events.

QE, RISK TESTING AND RESILIENCE

Resilience Element (DRII)	Quality Engineering Contribution	Risk Testing Contribution
Prevention	Enforces standards, automates quality gates	Validates proactive control effectiveness
Response	Ensures systems degrade gracefully	Tests incident detection and escalation pathways
Recovery	Defines and verifies recovery steps	Tests restoration and failover accuracy
Adaptation / Improvement	Uses post-incident metrics to improve quality	Re-tests modified controls and configurations

Example:

In a banking context, risk testing ensures that a **payment switch** can failover to a secondary node within the RTO defined in the **BIA**. Quality engineering ensures that the failover process is automated, verified, and documented.

BEST PRACTICES

- Best practices and governance alignment
  - Shift quality and risk testing left
    - Integrate QE and risk testing early in DevOps pipelines
    - Automate quality gates
    - For example: performance, security scans before deployment
  - Adopt continuous testing
    - Align with CICD workflows using automated regression and resilience tests
  - Use risk-based test prioritization
    - Focus test effort on high-risk, high-impact components
    - For example: critical services, security controls, compliance functions
  - Measure and report
    - Tie test outcomes to enterprise risk metrics (e.g., number of high-risk defects closed before release)
    - Map QE and risk testing activities to COBIT processes

COMMON CHALLENGES

Challenge	Impact	Mitigation Approach
Overemphasis on functional testing	Missed resilience gaps	Incorporate risk-based and non-functional testing.
Poor traceability between risks and tests	Incomplete coverage	Use risk-to-test mapping matrices.
Manual testing processes	Delayed feedback loops	Automate testing and reporting in pipelines.
Siloed QE and operations teams	Quality gaps in production	Integrate SRE/QA/DevOps collaboration.
Lack of metrics and governance	Weak resilience measurement	Establish KPIs/KRIs aligned to resilience goals.

RECAP

Dimension	QA	QC	QE	RE
Focus	Process compliance	Product verification	Systemic prevention	Long-term dependability
Orientation	Preventive	Detective	Preventive + Adaptive	Predictive
Typical Metrics	Process compliance %, audit pass rate	Defect rate, test coverage	Automation %, MTTR, SLA success	MTBF, reliability %, uptime
Risk Control Type	Preventive	Detective	Preventive + Corrective	Predictive + Corrective
Resilience Impact	Builds procedural resilience	Ensures immediate defect detection	Embeds resilience-by-design	Sustains resilience over time

Q&A AND OPEN DISCUSSION