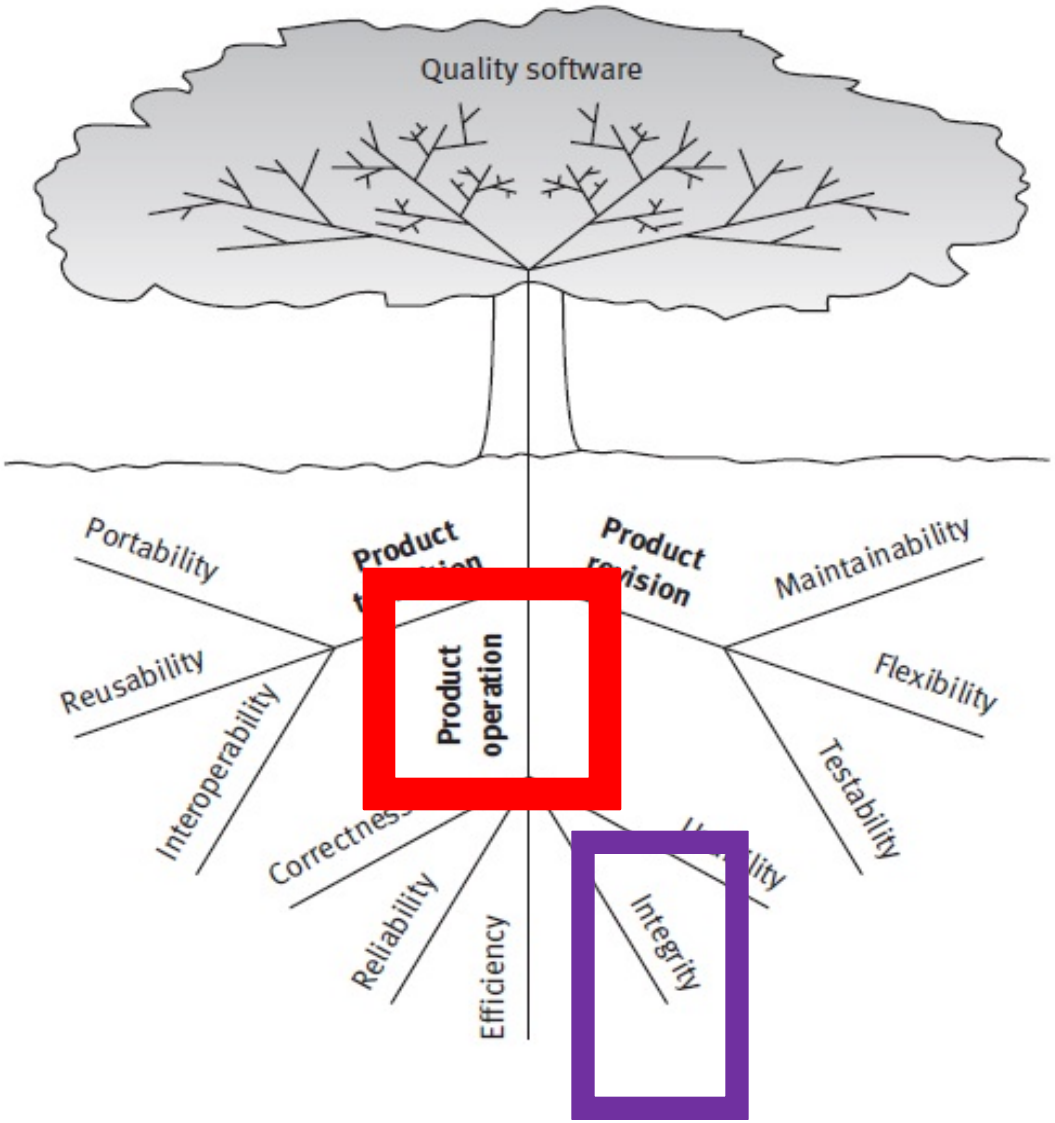
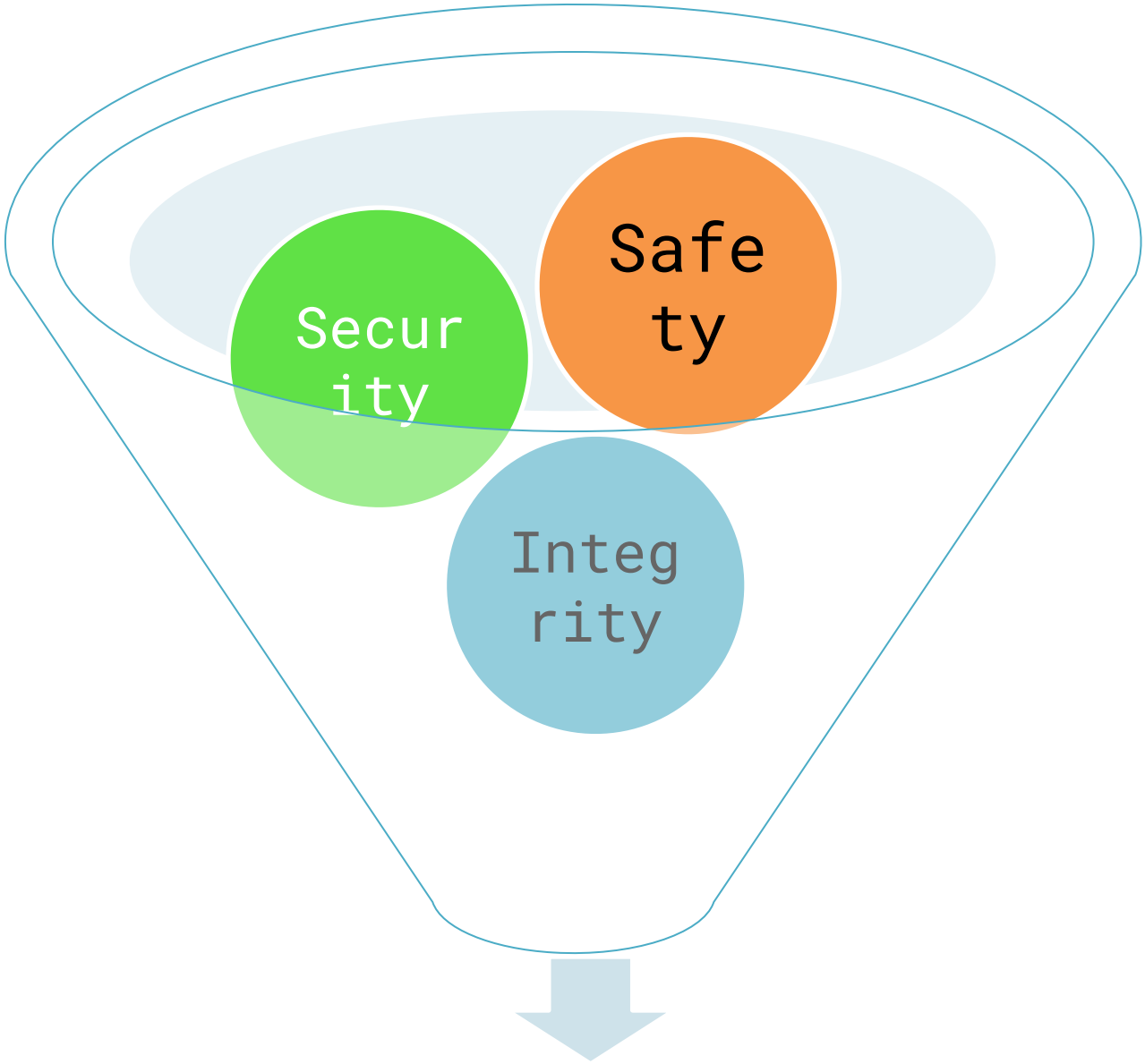
Course 6

Software Quality Factors in detail

Integrity / Security / Safety


Quality software
Portability
Reusability
Interoperability
Product operation
Product revision
Maintainability
Flexibility
Testability
Reliability
Efficiency
Integrity


Safe
ty
Secur
ity
Integ
rity

Integrity vs. Security vs. Safety

Integrity = ability to withstand attacks to its security

-Safety requires security: when unauthorized access results in an incident
-Security ⊆ safety – when any unauthorized access is considered an accident

Integrity vs. Security vs. Safety

Integrity

- Extent to which access to software or data by unauthorized persons can be controlled [McCall].
- Internal consistency or lack of corruption in electronic data and at the human level
- Prevents unauthorized or improper access & allow proper access

Security

- Protect data and resources contained in and controlled by the software
- Risk for privacy

Safety

- Freedom from software hazards
- Elimination of conditions hazardous to operation as a result of errors in process control software [Deutsch & Willis model]
- Risk for human lives or environment

Integrity [McCall]

- Criteria:
  - Access control
  - Access audit

- Measured in: analysis, design, implementation
- Impact in operation

Access Control

- User access control (ID, password)
- Database access control (authorization table, privacy)
- Memory protection across tasks (mobile app, partitions)

Access Audit

- Support for recording and reporting access
- Support for awareness of access violation

Security

- Remarks:
  - Weakest link
  - Environment dependable

- Approach - Risk analysis:
  - Identify risks
  - Evaluate ways to protect against it
- Detect security vulnerabilities

Security vulnerabilities

1. Buffer overflow / overrun:
   - write outside the limits of data structure
   - C/C++
2. Race conditions: an atomic operation is performed in 2 steps
   - File access permission
3. API functions – insecure interfaces

Security vulnerabilities

4. Untrusted input: accept and use data from untrusted sources
5. Result verification:
   - function and system calls fail
   - Traceroute / tracert (Windows) - track the route of internet packages
6. Data and privilege leakage

Solutions

- Java approach:
  - Associating permissions with code and checking them – security manager
- Tools:
  - Flawfinder – C/C++ code
    - Looking for functions known for security vulnerabilities associations
    - http://www.dwheeler.com/flawfinder/
  - Splint – C code
    - Checks for security vulnerabilities and coding mistakes
    - Parse C code
    - Allows code annotation
    - http://lclint.cs.virginia.edu/

OWASP Top 10 Security Risks – 2017 / 2013

1. Injection
2. Broken Authentication and session management
3. Sensitive Data Exposure (former 6)
4. XML external entities (new)
5. Broken access control (new)
6. Security misconfiguration (former 8)
7. Cross-Site Scripting
8. Insecure deserialization (new)
9. Using components with known vulnerability
10. Insufficient Logging & Monitoring

Show CWE distribution

Categories	Vulnerabilities		Security Hotspots		
			Open	In Review	Won't Fix
A1 - Injection	0	A	17	0	0
A2 - Broken Authentication	0	A	0	0	0
A3 - Sensitive Data Exposure	2	B	6	0	0
A4 - XML External Entities (XXE)	0	A	0	0	0
A5 - Broken Access Control	0	A	79	0	0
A6 - Security Misconfiguration	4	E	1	0	0
A7 - Cross-Site Scripting (XSS)	0	A	4	0	0
A8 - Insecure Deserialization	0	A	0	0	0
A9 - Using Components with Known Vulnerabilities	0	A	0	0	0
A10 - Insufficient Logging & Monitoring	0	A	0	0	0
Not OWASP	1	D	0	0	0

New

- OWASP Mobile Security Project – top 10 (2016)

Software Security Measurement and Assurance

- CWE – Common Weakness Enumeration cwe.mitre.org
  - Community-developed formal list of common software weaknesses
- CERT – Computer Emergency Center – Carnegie Melon Univ.

  http://www.cert.org/
  - Software security assurance
  - Security measurements and analysis
  - Frameworks and protocols
  - Methods and tools
- IBM, Microsoft, RedHat, Apple,…

CERT Risk analysis

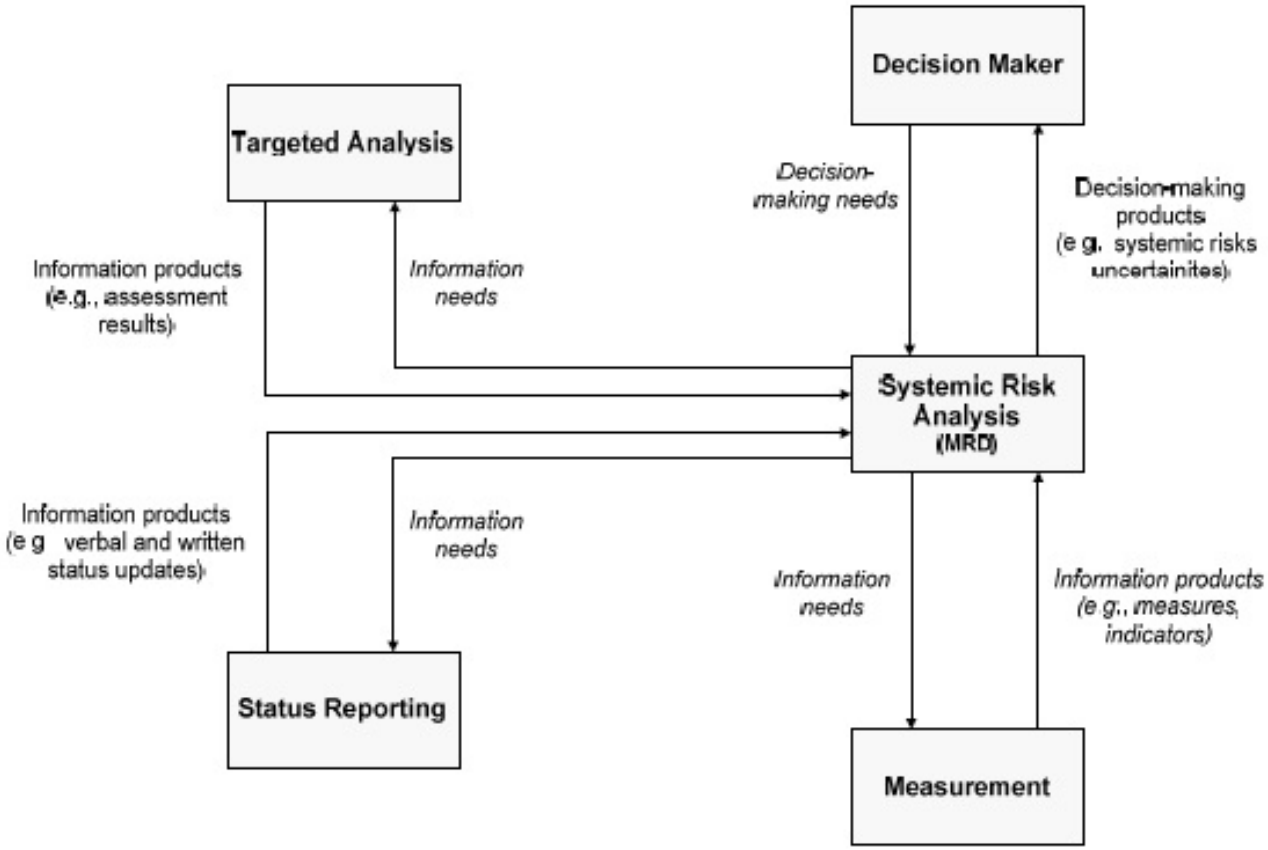
- 2 approaches:
  1. Tactical
     - risk = probability that an event will lead to a negative consequence or a loss
     - Goal: evaluate systems components for potential failure
  2. Systemic
     - Risk = probability of mission failure

CERT - IMAF

- Integrated Measurement and Analysis Framework
- Application in software security

[C. Alberts, J. Allen, R. Stoddard – Risk-Based Measurement and Analysis: Application to Software Security]

CERT - IMAF


Decision Maker
Targeted Analysis
Decision-making needs
Decision-making products (e.g., systemic risks uncertainites)
Information products (e.g., assessment results)
Information needs
Systemic Risk Analysis (MRD)
Information products (e.g., verbal and written status updates)
Information needs
Information needs
Information products (e.g., measures, indicators)
Status Reporting
Measurement

Case study Microsoft – security checklists

[M. Howard, D. LeBlanc – Writing Secure Code]

- Designer
- Developer
  - General
  - Web & database-specific
  - RPC
  - ActiveX, COM and DCOM
  - Crypto and Secret Management
  - Managed Code
- Tester

Snapshot Designer's Security Checklist

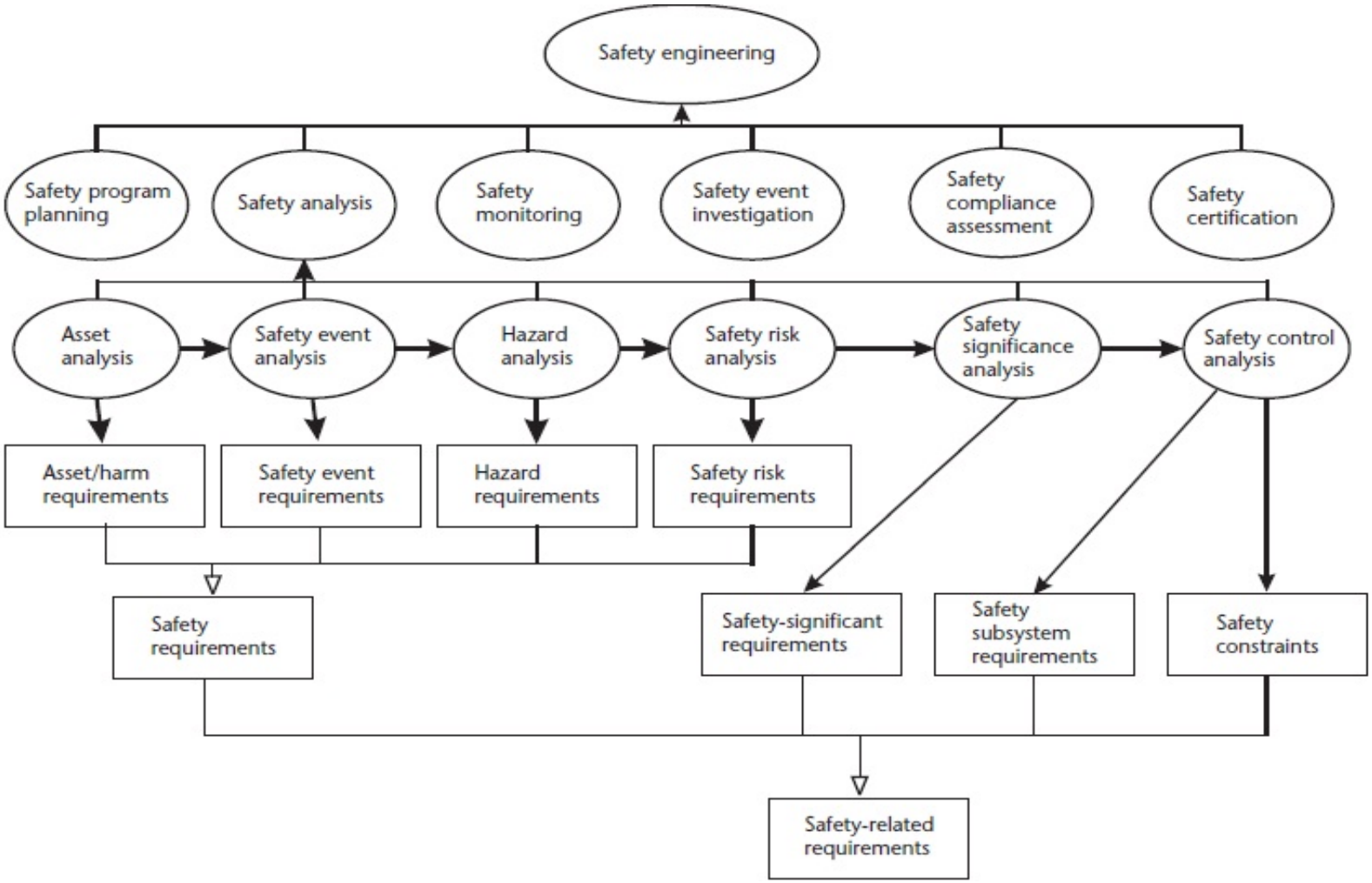
- Person responsible for bug tracking (BugTrack - http://www.bugtrack.net/)
- Competitor's vulnerabilities analyzed
- Past vulnerabilities in previous versions analyzed
- Safe-for-scripting controls reviewed
- Sample code reviewed for security issues
- Default install is secure
- Threat models complete for design phase
- Security failures logged for later analysis
- . . .

Safety

- Freedom of conditions that can cause death, injury, damage, loss, environmental harm
- Important in safe critical systems
- Harm severity – classification:
  - Catastrophic
  - Critical (severe)
  - Major
  - Minor
  - No effect (negligible)

System Safety Plan

[Firesmith, D., Engineering Safety-Related Requirements for Software-Intensive Systems]



- 1228-1994 - IEEE Standard for Software Safety Plans
- Software Safety Analysis Procedures – USA, 2014