New CISQ/OMG Automated Source Code Technical Debt Standard
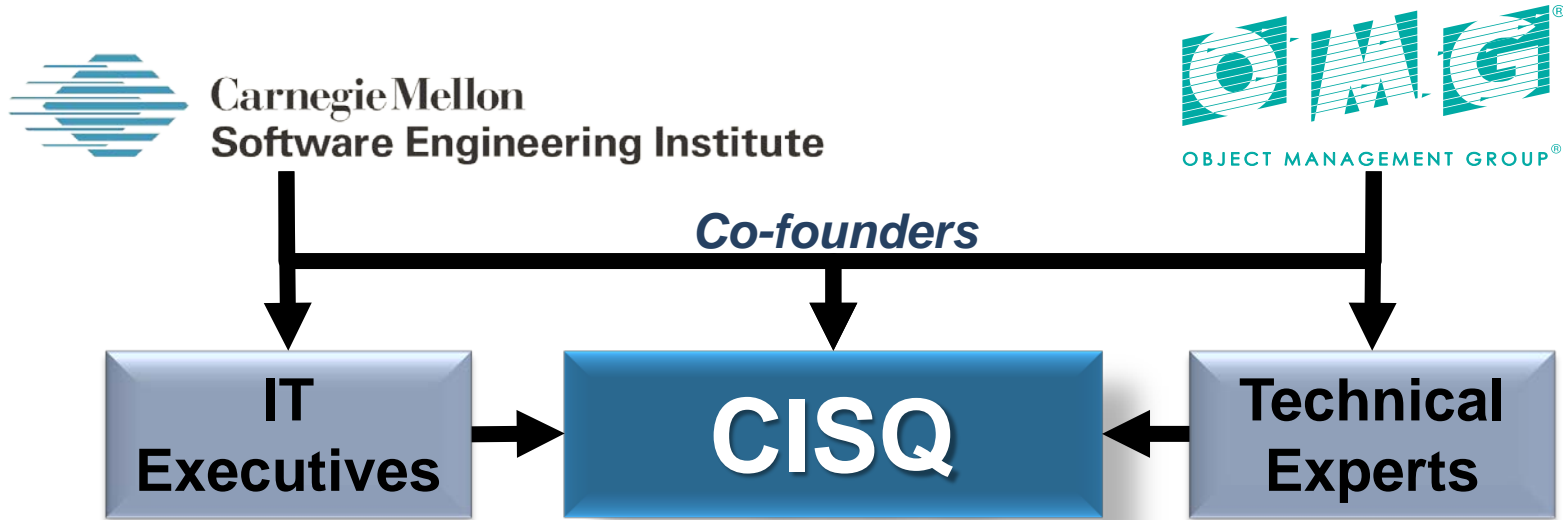
Dr. Bill Curtis, Executive Director

CISQ
Consortium for IT Software Quality

January 16, 2018

What is CISQ?

CISQ — Consortium for IT Software Quality

Carnegie Mellon Software Engineering Institute

OMG — Object Management Group®

Co-founders

IT Executives → **CISQ** ← Technical Experts

OMG Special Interest Group

CISQ is chartered to define automatable measures of software size and quality that can be measured in the source code, and promote them to become Approved Standards of the OMG®

CISQ Sponsors

Cognizant SYNOPSYS®

CAST — ACHIEVE INSIGHT. DELIVER EXCELLENCE. CGI

What Is Technical Debt?
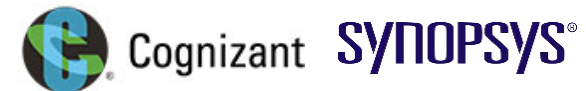
"Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite…Every minute spent on not-quite-right code counts as interest on that debt."

- Ward Cunningham, OOPSLA, 1992
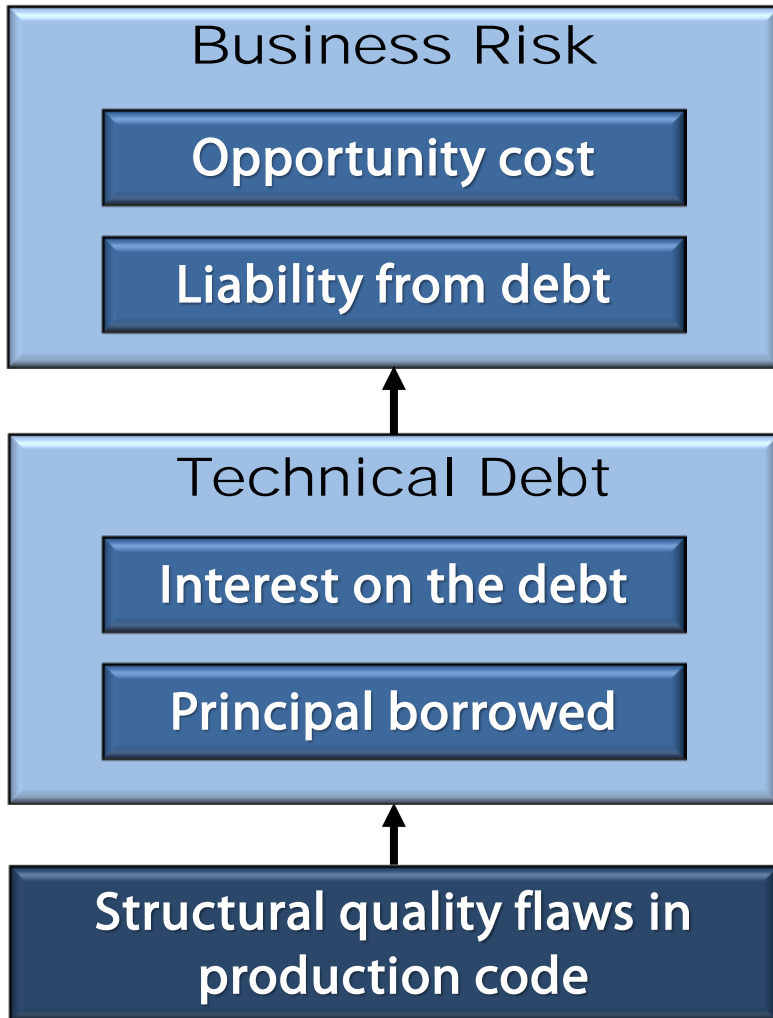
Strategic tradeoff

The cost of shipping sub-optimal code versus the business benefit derived

Although the Agile Alliance describes defects as "Quality Debt", industry most often treats defects and sub-optimized design jointly as 'Technical Debt' because they incur a corrective maintenance cost.

Industry Use

The correction costs and interest related to flaws that must be removed

Technical Debt Metaphor

Technical Debt —— future cost of fixing severe architectural and coding flaws in released software, part of cost of ownership

Business Risk

- Opportunity cost
- Liability from debt

Technical Debt

- Interest on the debt
- Principal borrowed

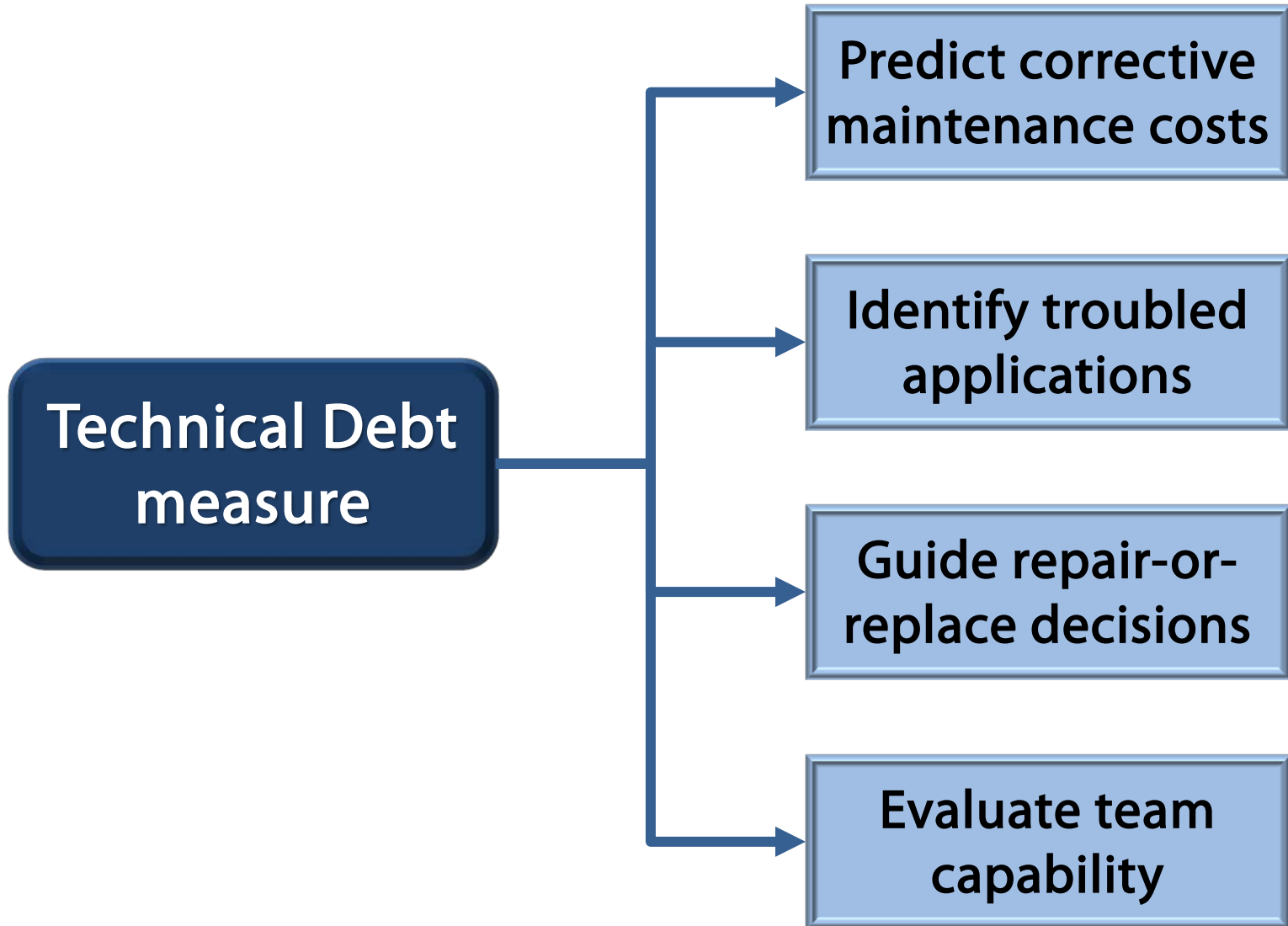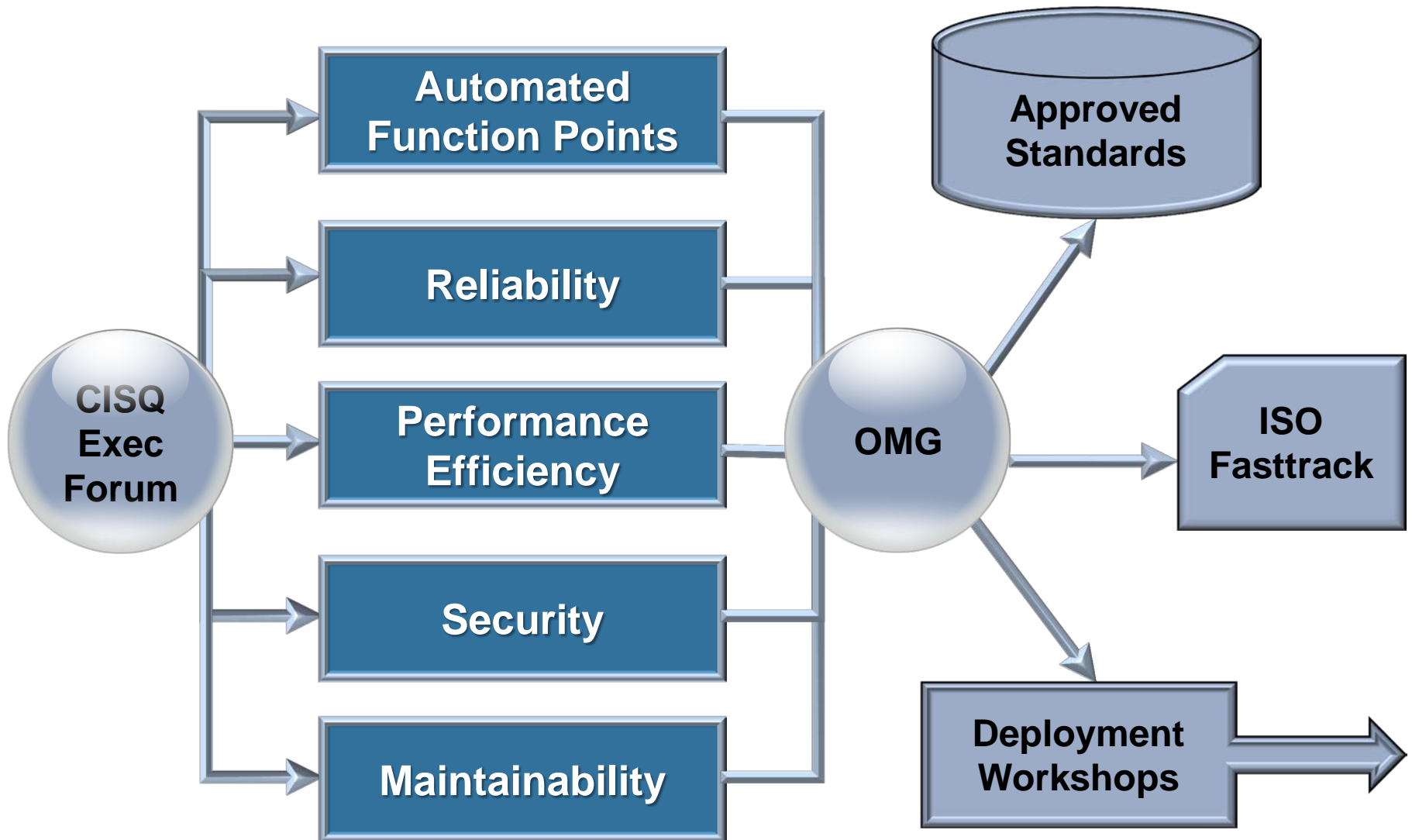## Structural quality flaws in production code

Opportunity cost—benefits that could have been achieved had resources been devoted to new capabilities rather than retiring technical debt

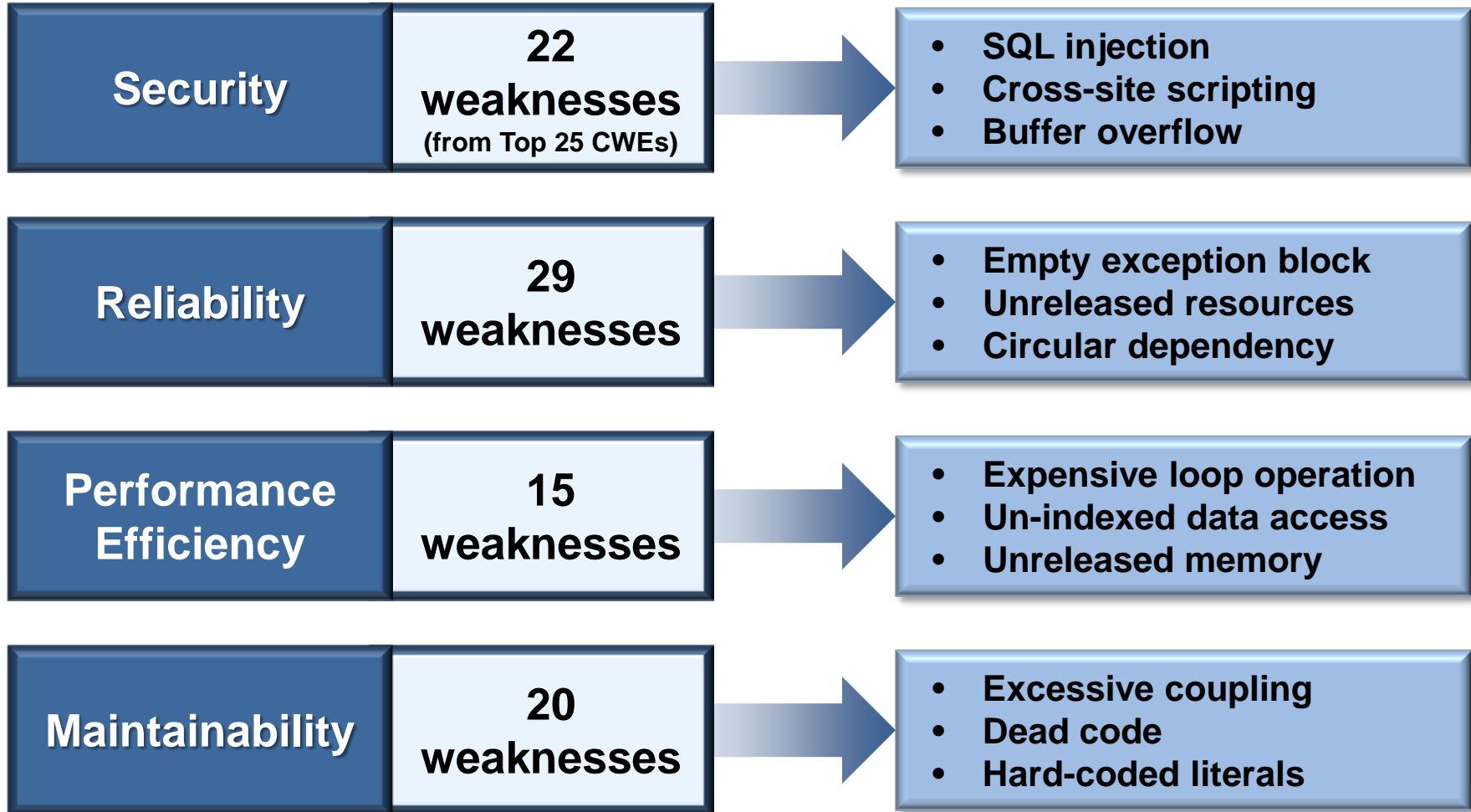**Liability**—business costs related to outages, breaches, corrupted data, etc.

Interest—continuing IT costs attributable to the flaws causing technical debt, i.e, greater developer effort, greater resource usage, etc.

Principal—cost of fixing architectural and coding flaws in released code that must be removed

Curtis, et al. (2012). IEEE Software.

What IT Management Wants

CISQ
Consortium for IT Software Quality

Technical Debt measure

- Predict corrective maintenance costs
- Identify troubled applications
- Guide repair-or-replace decisions
- Evaluate team capability

Content of CISQ Measures

CISQ Quality Characteristic Measures

Example architectural and coding weaknesses composing CISQ measures

Security	**22 weaknesses** (from Top 25 CWEs)

- SQL injection
- Cross-site scripting
- Buffer overflow

Reliability	**29 weaknesses**

- Empty exception block
- Unreleased resources
- Circular dependency

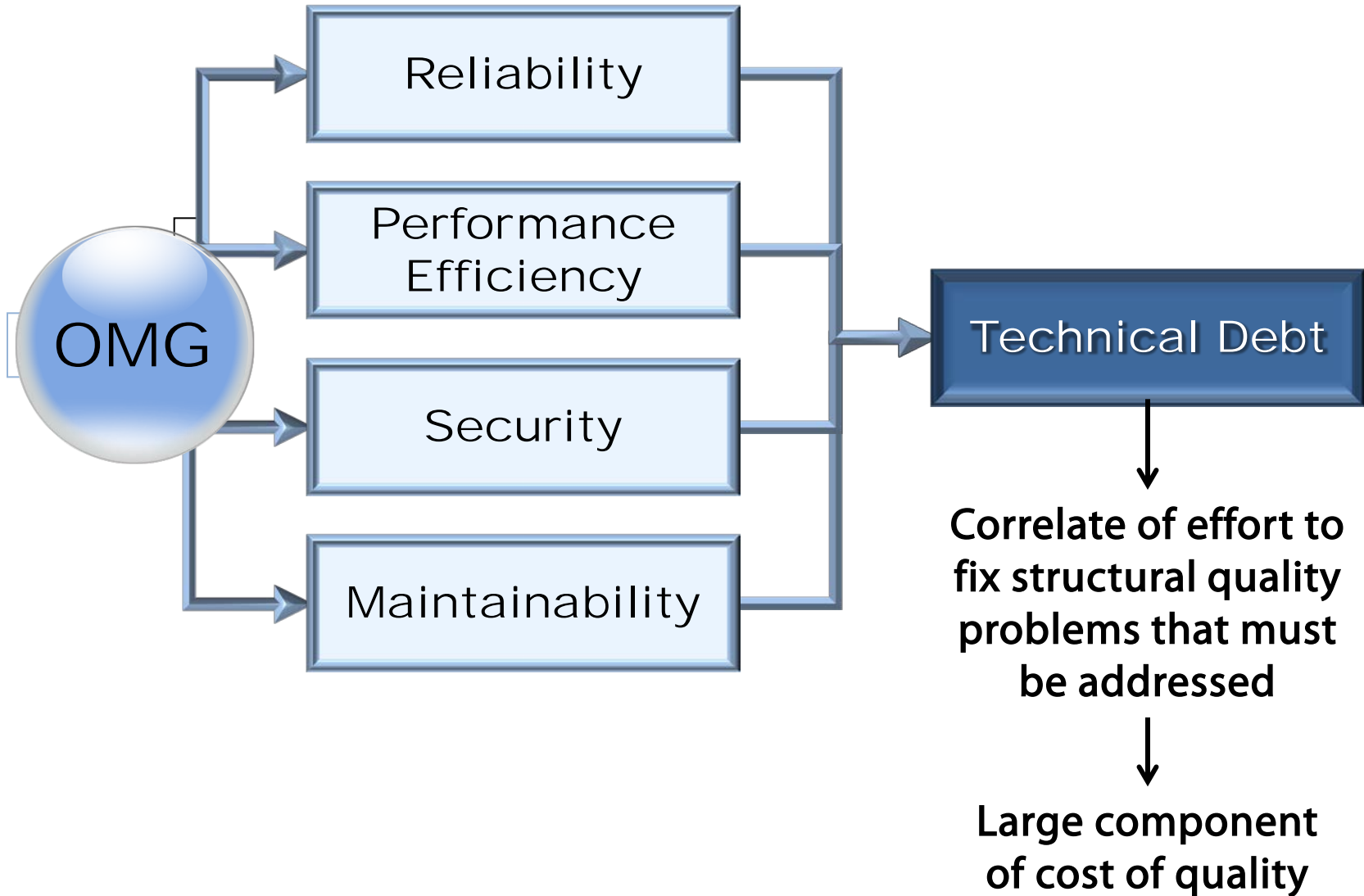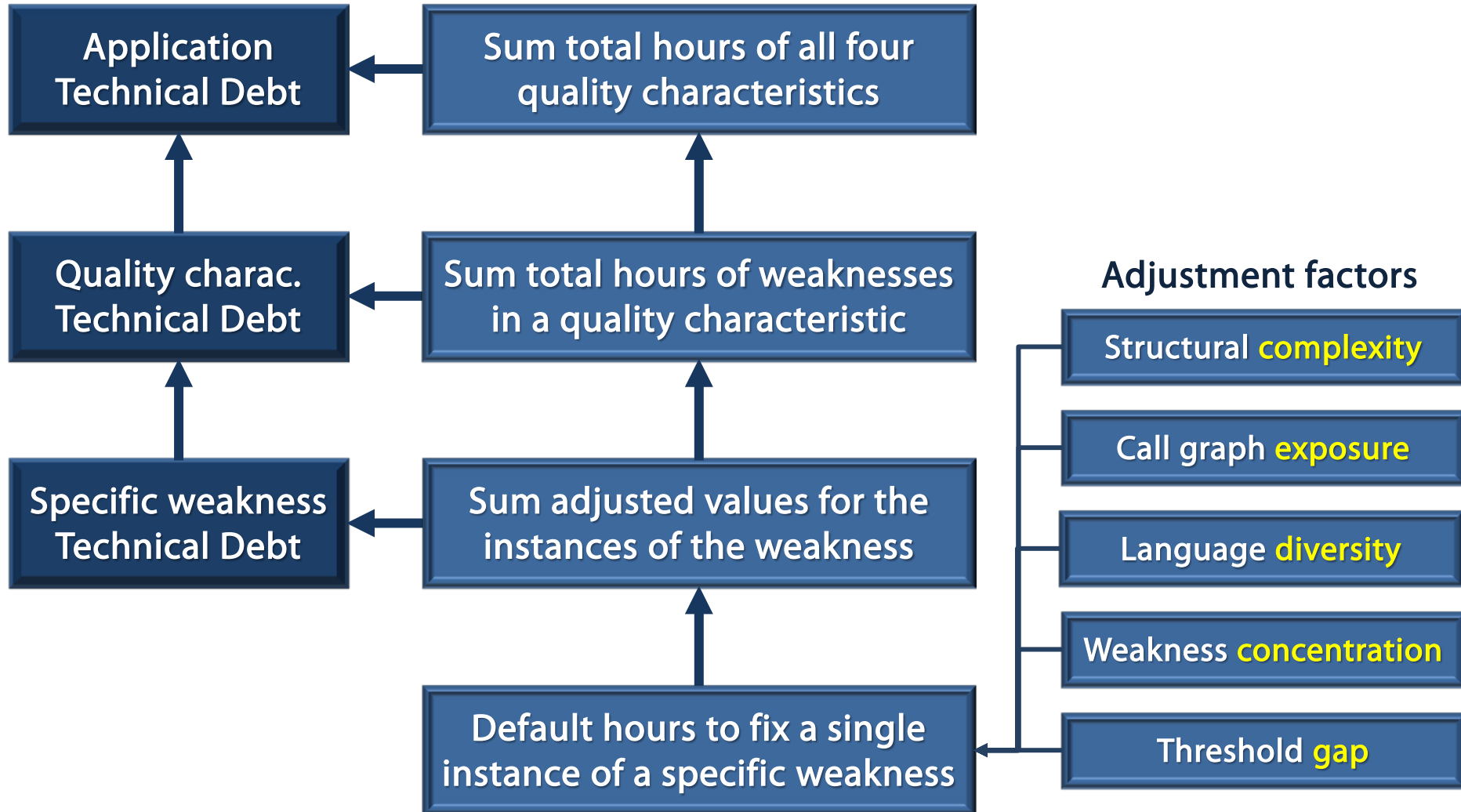| Performance Efficiency	**15 weaknesses**

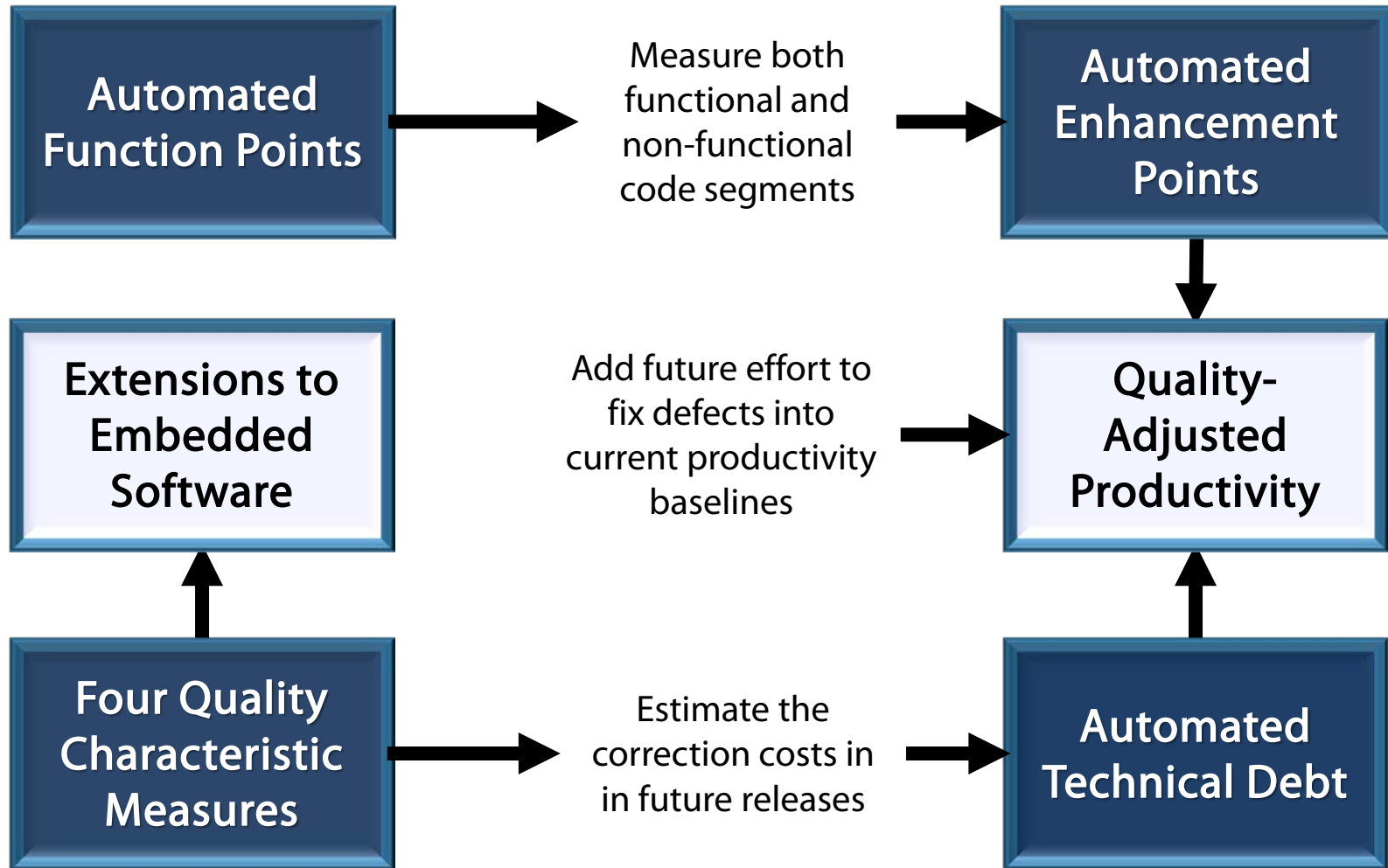- Expensive loop operation
- Un-indexed data access
- Unreleased memory

Maintainability	**20 weaknesses**

- Excessive coupling
- Dead code
- Hard-coded literals

CISQ
Consortium for IT Software Quality

CISQ Technical Debt Calculation

Application Technical Debt ← **Sum total hours of all four quality characteristics**

Quality charac. Technical Debt ← **Sum total hours of weaknesses in a quality characteristic**

Specific weakness Technical Debt ← **Sum adjusted values for the instances of the weakness**

Default hours to fix a single instance of a specific weakness

Adjustment factors

- Structural **complexity**
- Call graph **exposure**
- Language **diversity**
- Weakness **concentration**
- Threshold **gap**

CISQ Measurement Roadmap

Automated Function Points → Measure both functional and non-functional code segments → **Automated Enhancement Points**

Extensions to Embedded Software

Add future effort to fix defects into current productivity baselines → **Quality-Adjusted Productivity**

Four Quality Characteristic Measures → Estimate the correction costs in in future releases → **Automated Technical Debt**

Join CISQ !   www.it-cisq.org