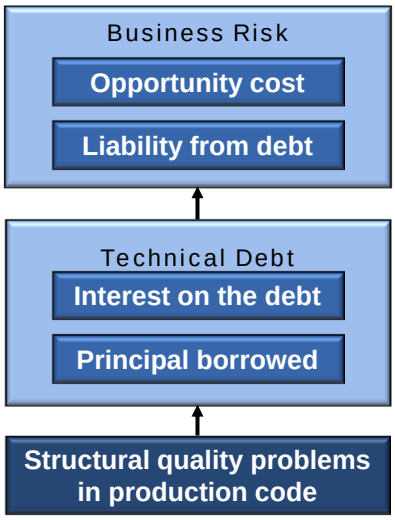
Measuring and Managing Technical Debt

Dr. Bill Curtis
SVP & Chief Scientist, CAST Research Labs
Director, Consortium for IT Software Quality

The Technical Debt Metaphor

Technical Debt — **the future cost of defects remaining in code at release, a component of the cost of ownership**



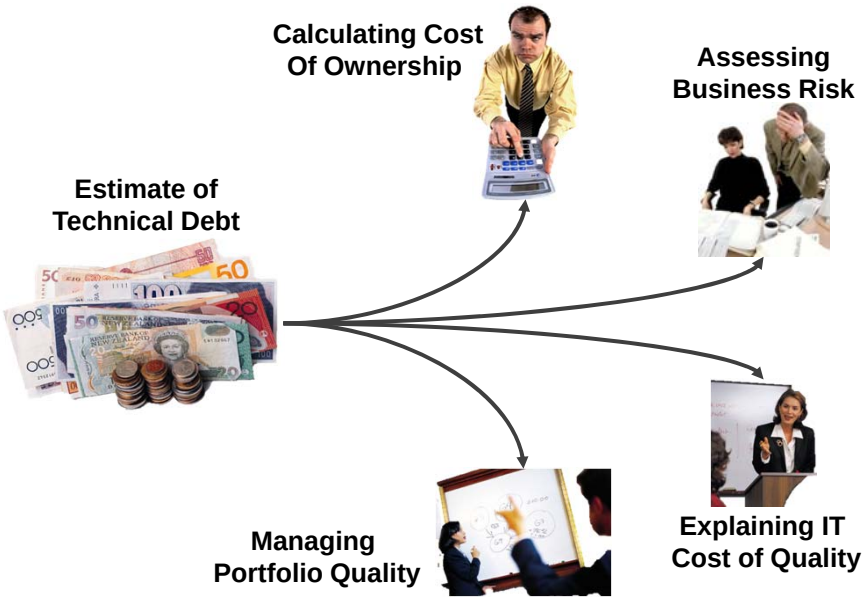
Opportunity cost—**benefits that could have been achieved had resources been put on new capability rather than retiring technical debt**

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

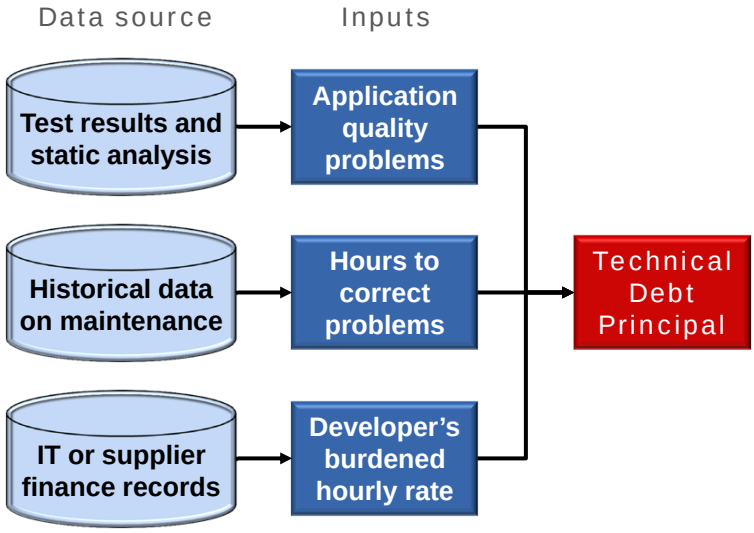
Interest—**continuing IT costs attributable to the violations causing technical debt, i.e, higher maintenance costs, greater resource usage, etc.**

Principal—**cost of fixing problems remaining in the code after release that must be remediated**

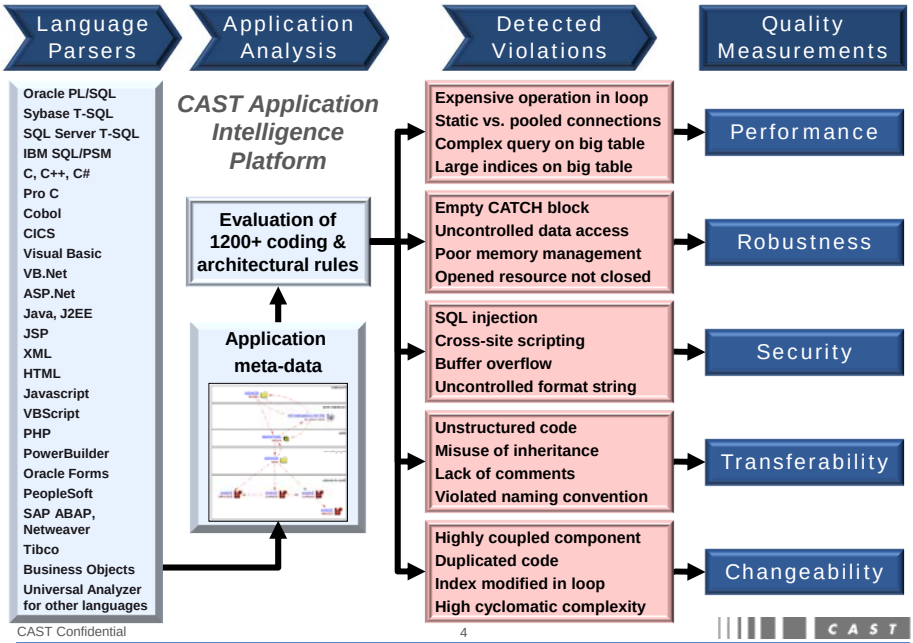
Uses of Technical Debt Metaphor



Inputs for Estimating Principal

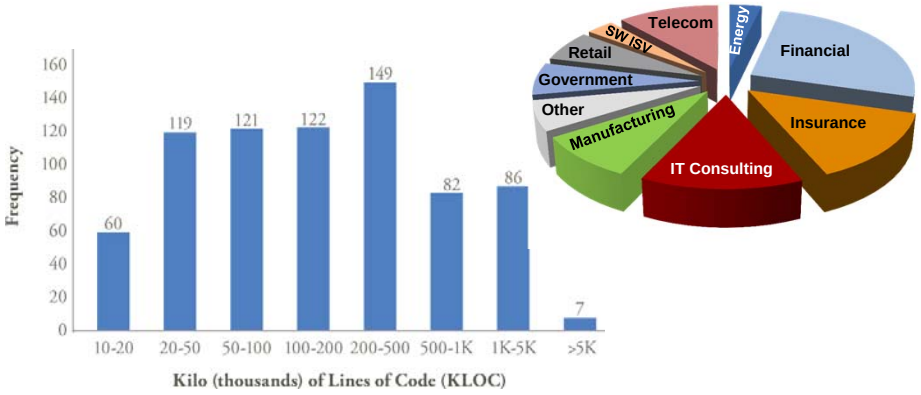


Analyzing Structural Quality at System Level

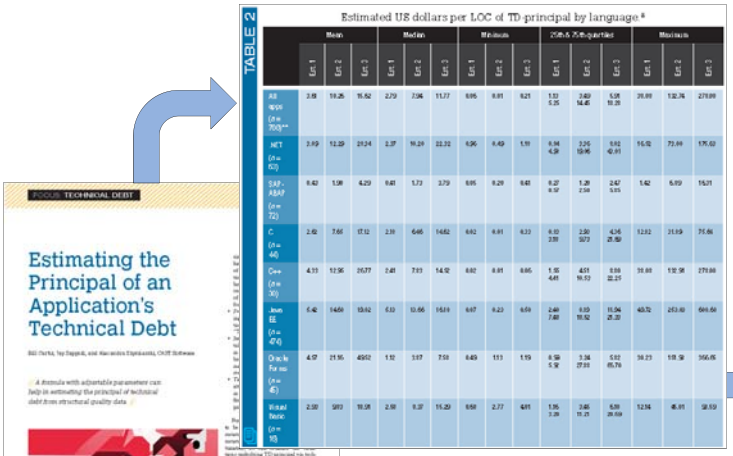


Appmarq Repository

- Industry-leading repository on structural quality
  - **745 Applications**
  - **160 Companies, 14 Countries**
  - **321,259,160 Lines of Code; 59,511,706 Violations**



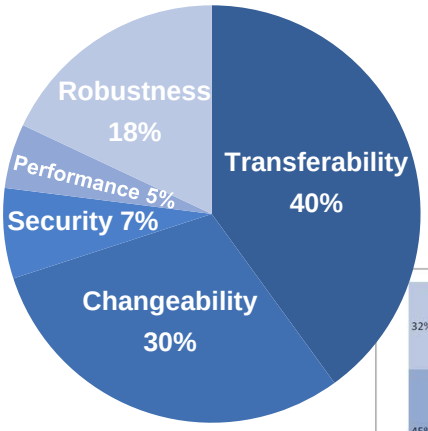
Estimating Technical Debt



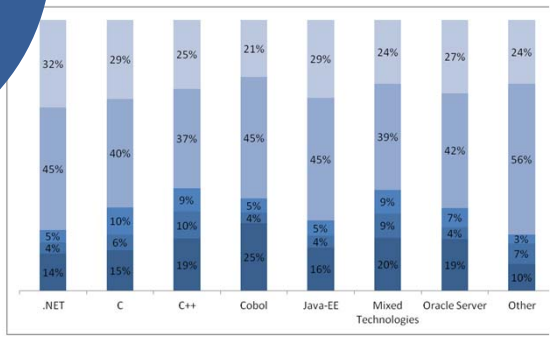
Conservative estimate: $3.61 per LOC

"Even when measured with a conservative formula, the amount of technical debt in most business applications is formidable… estimates of [technical debt] can be a powerful tool to aid management in understanding and controlling IT costs and risks."

Technical Debt by Software Quality Attribute



- **70% of Technical Debt is in IT Cost** (Transferability, Changeability)
- **30% of Technical Debt is in Business Risk** (Robustness, Performance, Security)
- **Proportions are generally consistent across technologies**



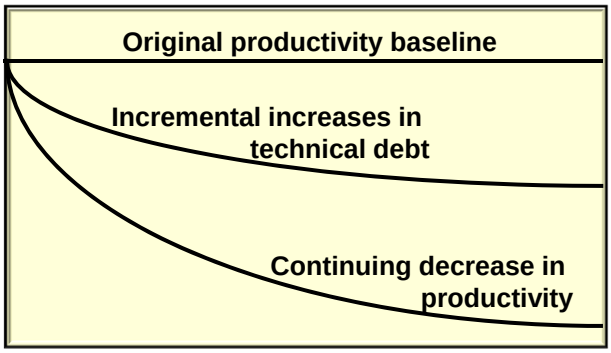
Rethinking Productivity Measurement

$$\text{Release Productivity} = \frac{\text{Volume of code developed, modified, or deleted}}{\text{Total effort expended on the release}}$$

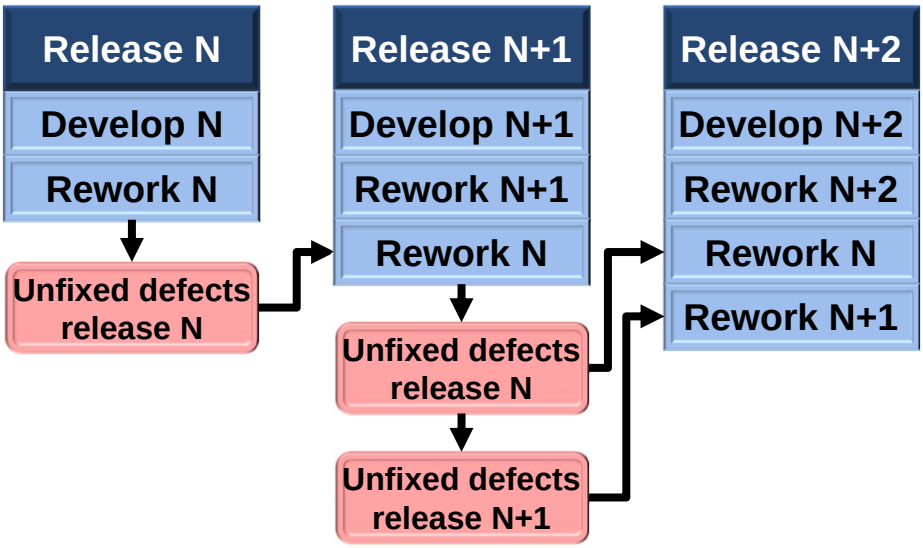
Productivity baseline —

a value in a monotonically declining function that compares the amount of product produced to the effort required to produce it

… unless you take action



Technical Debt = Carry-forward Rework



Quality-Adjusted Productivity

Quality-Adjusted Productivity = **Release productivity ⊗ f(Technical debt)**

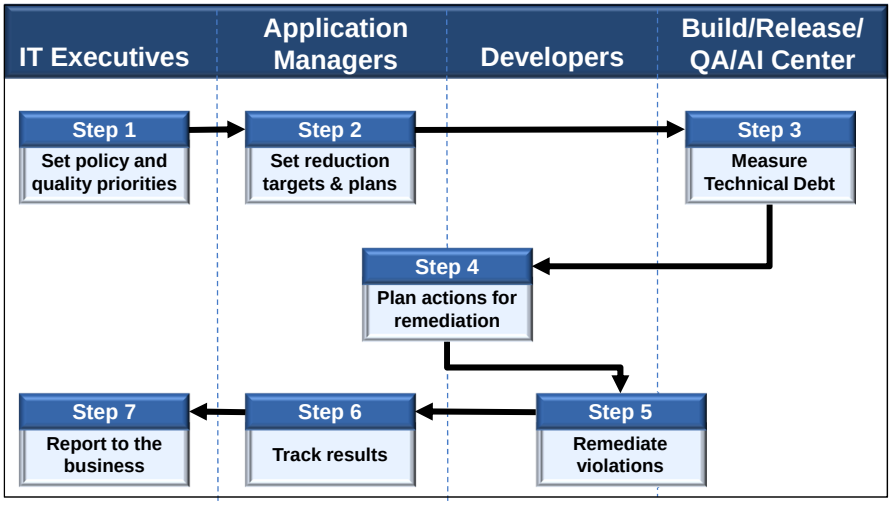
Release Productivity should be adjusted for:

1. **Effort shifted forward for fixing functional defects added in this release**
2. **Effort shifted forward for fixing structural defects added in this release**
3. **Future effort caused by maintainability problems added in this release**

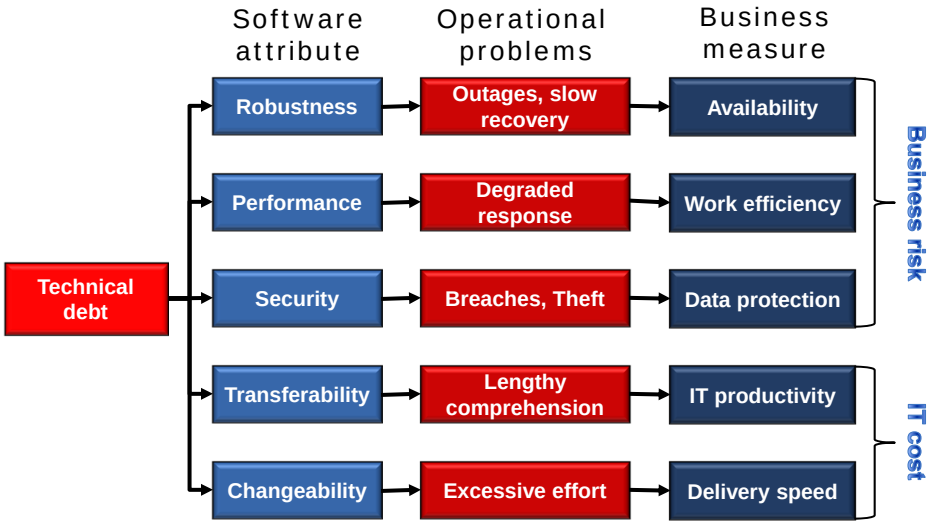
Stronger relationship to:

- **Architectural integrity**
- **Total Cost of Ownership**
- **Business risk**

Manage Technical Debt to Manage Productivity



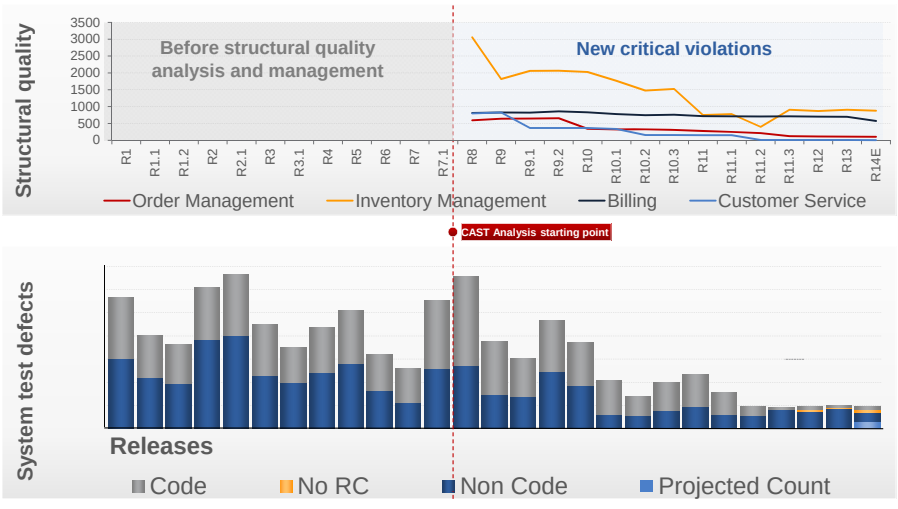
Translating Tech Debt to Business Measures



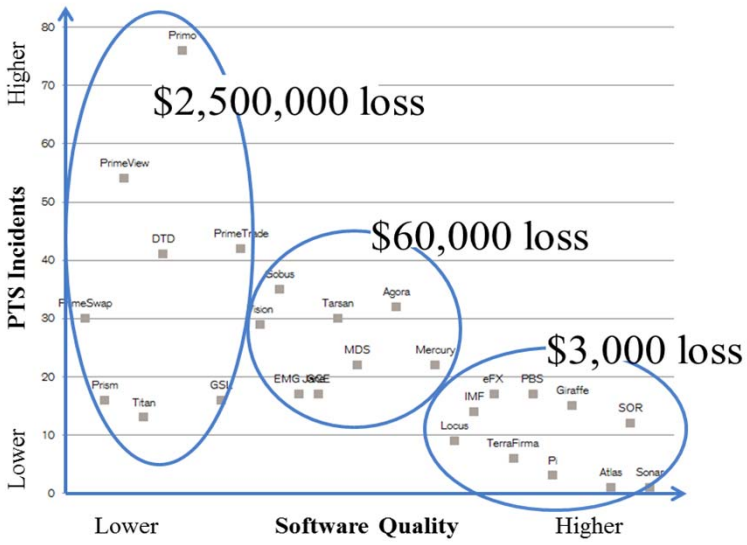
Managing Structural Quality in Telecom

Measured impact in a complex enhancement-heavy environment

CLIENT STUDY OVER 24 MONTHS

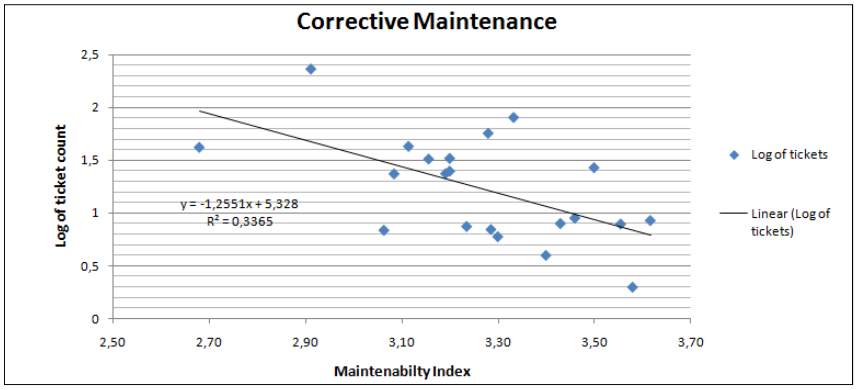


Benefit of Tech Debt Reduction in Banking



Tech Debt Reduction and Incident Rate

Correlation of maintenance effort with incident tickets across 20 customers of a global system integrator



Increase of TQI by 0.24 = decrease in maintenance activity by 50%