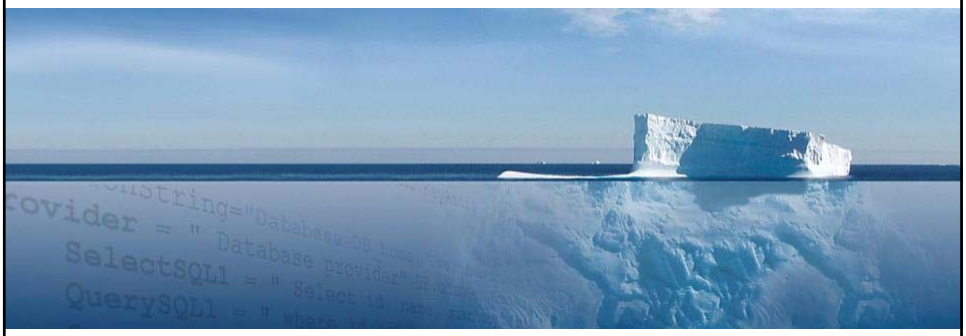
CAST
ACHIEVE INSIGHT. DELIVER EXCELLENCE.



9 Digit Stakes…

…and the Measurement Stack

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

Bill's December 2011 Trip

OWASP Top 10 – 2013 Edition

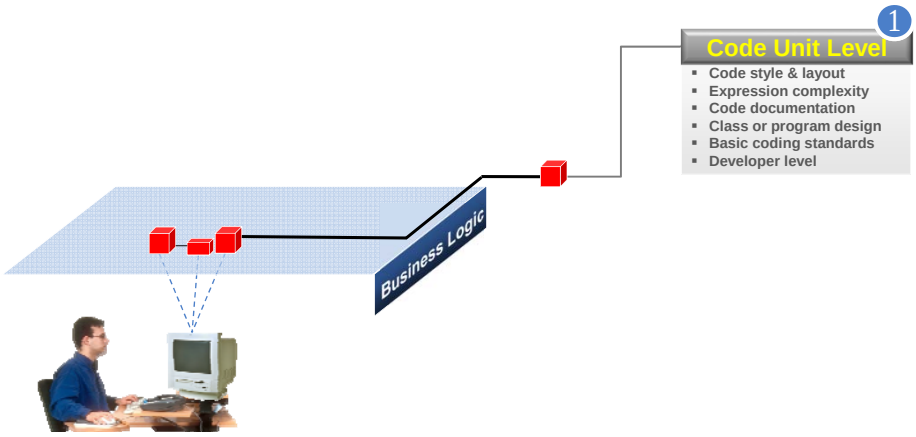
A1 Injection
A2 Broken Authentication and Session Management
A3 Cross-Site Scripting (XSS)
A4 Insecure Direct Object References
A5 Security Misconfiguration
A6 Sensitive Data Exposure
A7 Missing Function Level Access Control
A8 Cross-Site Request Forgery (CSRF)
A9 Using Known Vulnerable Components – New (prev
Misconfiguration")
A10 Unvalidated Redirects and Forwards

CAST

It's 10AM, Do You Know Where Your Money Is ?

No man's property is safe while Wall Street is in session !

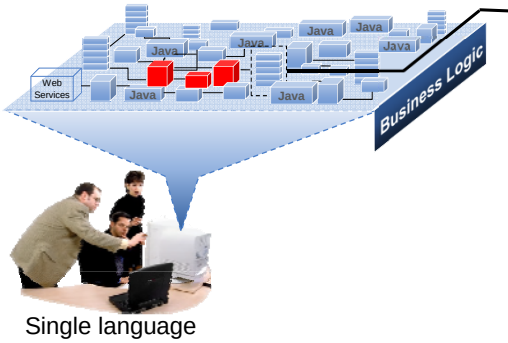
Code Unit Level — Pre-Build Analysis



IDE Unit Test & Static Analysis tools

Developer level code unit analysis

Technology Level — Post-Build Analysis



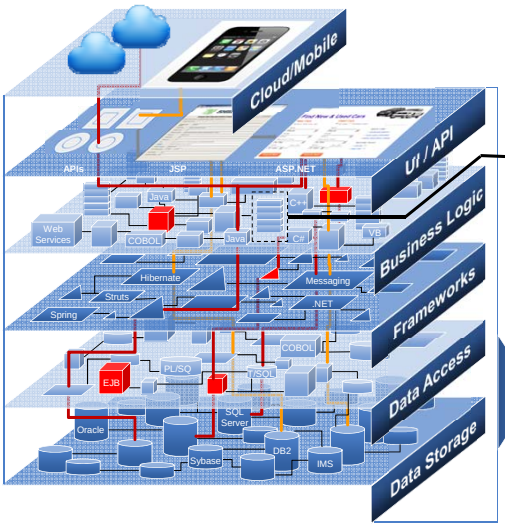
Single language static analysis tools

Quality Assurance

1 Code Unit Level
- Code style & layout
- Expression complexity
- Code documentation
- Class or program design
- Basic coding standards
- Developer level

2 Technology Level
- Single language/technology layer
- Intra-technology architecture
- Intra-layer dependencies
- Design & structure
- Inter-program invocation
- Security vulnerabilities
- Development team level

System Level — System Integration Analysis

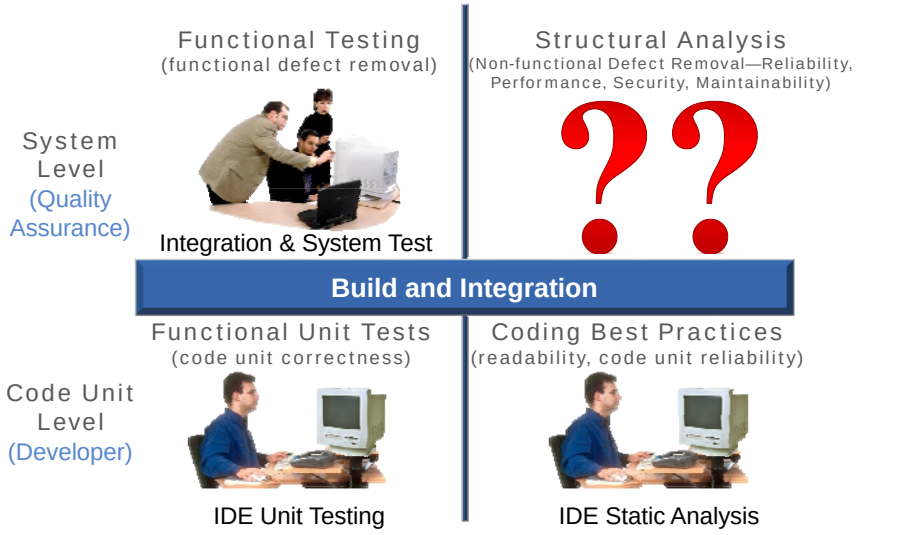


1 Code Unit Level
- Code style & layout
- Expression complexity
- Code documentation
- Class or program design
- Basic coding standards
- Developer level

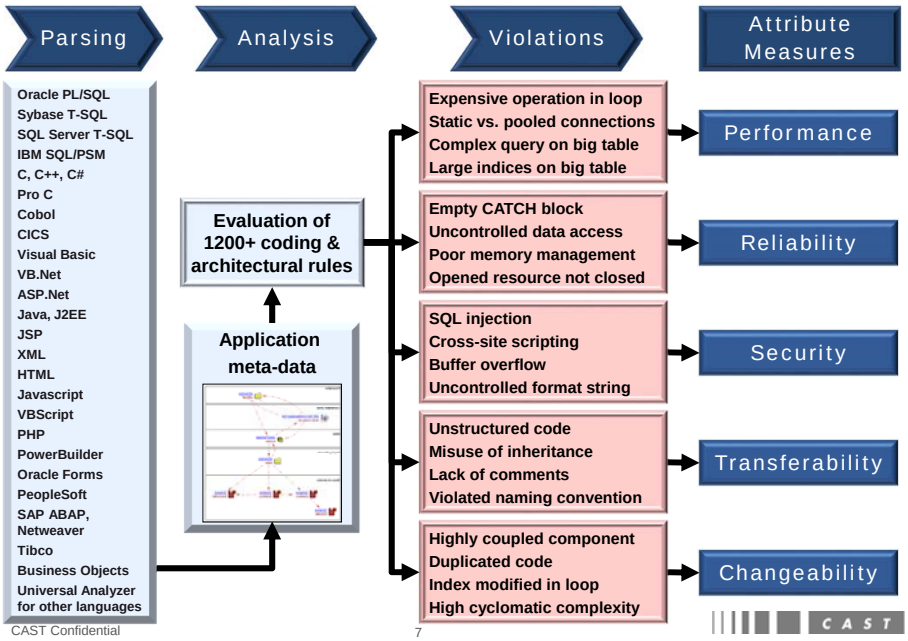
2 Technology Level
- Single language/technology layer
- Intra-technology architecture
- Intra-layer dependencies
- Design & structure
- Inter-program invocation
- Security vulnerabilities
- Development team level

3 Application Stack Level
- Integration quality
- Architectural compliance
- Risk propagation
- Application security
- Resiliency checks
- Transaction integrity
- Function point,
- Effort estimation
- Data access control
- SDK versioning
- Calibration across technologies
- IT organization level

The QA Gap



Analyzing System Level Structural Quality



Architecturally-Complex, Multi-Component Defects

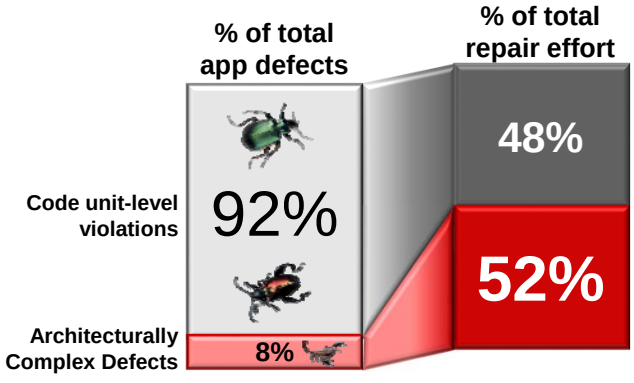
Study of defects across 1 open source, 2 large NASA applications

Observation	% of cases
Fixes mapping to ≥ 2 files	≈ 60%
Fixes mapping to ≥ 3 files	≈ 30-40%
Fixes mapping to ≥ 2 components	≈ 10-36%
Fixes mapping to ≥ 2 subsystems	≈ 10-20%
Spread of faults	80% of faults in 20% of files

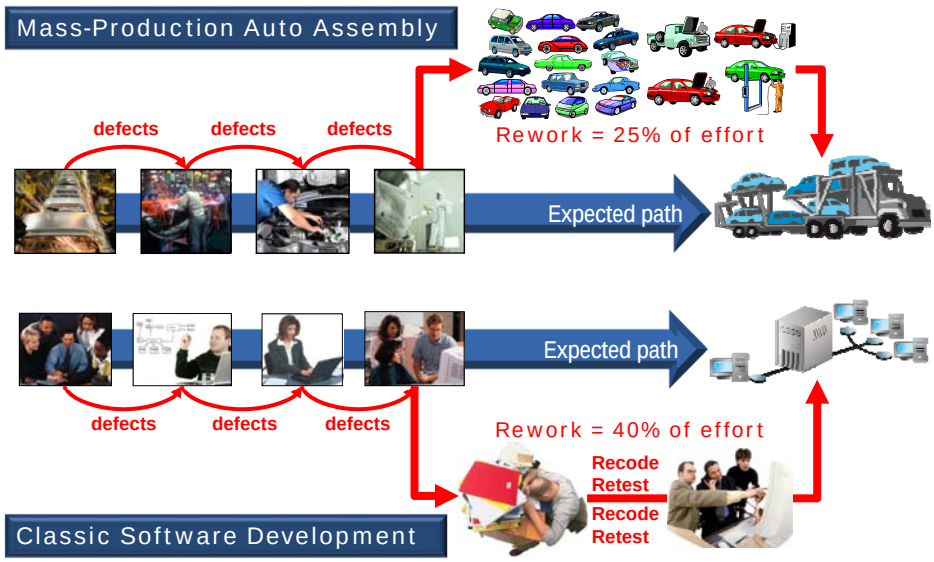
M. Hamill & K. Goseva-Popstojanova (2009). Common faults in software fault and failure data. *IEEE Transactions of Software Engineering*, 35 (4), 484-496.

2) Detect Architecturally Complex Defects

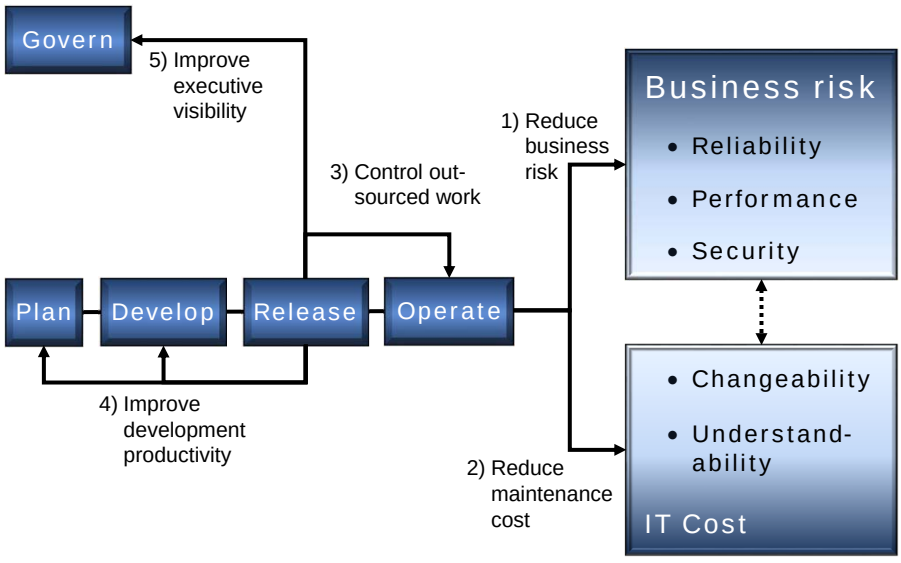
Architecturally Complex Defect: A structural flaw involving interactions among multiple components, often residing in different subsystems



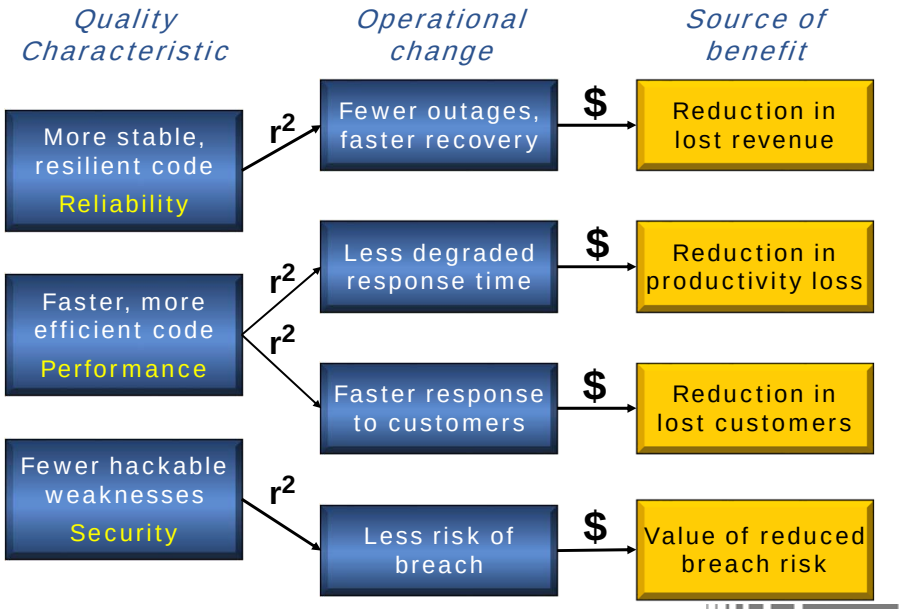
Productivity and Rework — Detroit Was Better



Five Purposes for Software Measurement



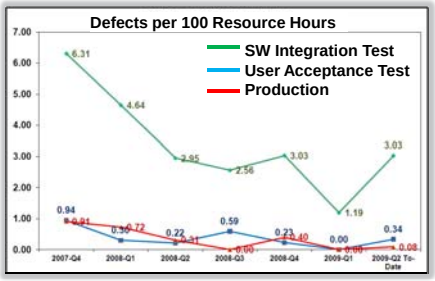
Structural Quality in Business Risk Terms

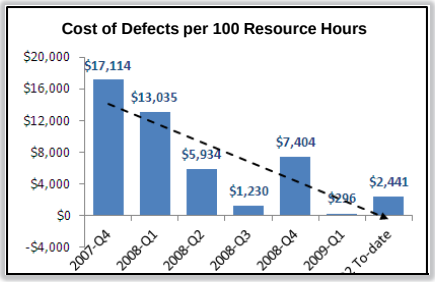


Case Study 1 — Major US Consumer Bank

Situation

- Retirement services, >$100B in assets
- 75 supported application
- Complex technology environment
- IT-intensive business process
- Initiated structural quality analysis 4Q07



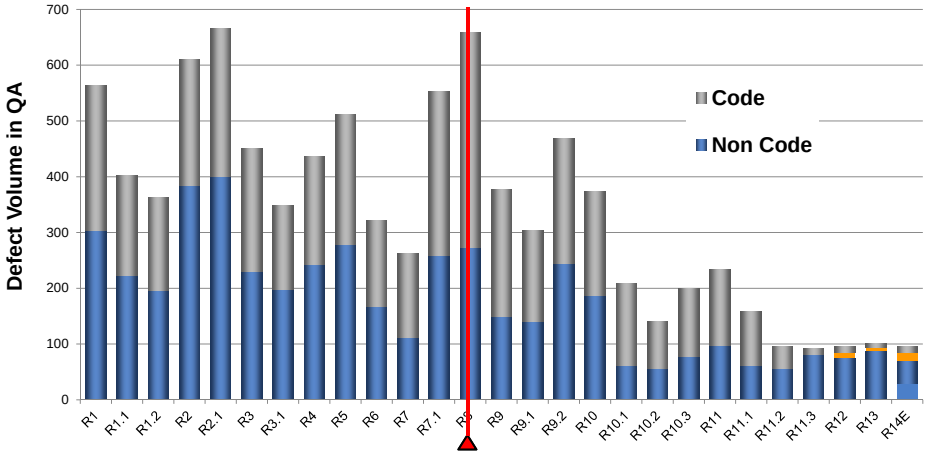


Result

- Sustained reduction in test and production defects
- 7X reduction in defect costs

Case Study 2 — Large Telco Reduces Defect Costs

- Order Management System (OMS)
- J2EE, VB, ASP, OMS Oracle, XML, Amdocs Enabler
- Multi-year development, >$100m per year, 6 releases PY, runaway costs,



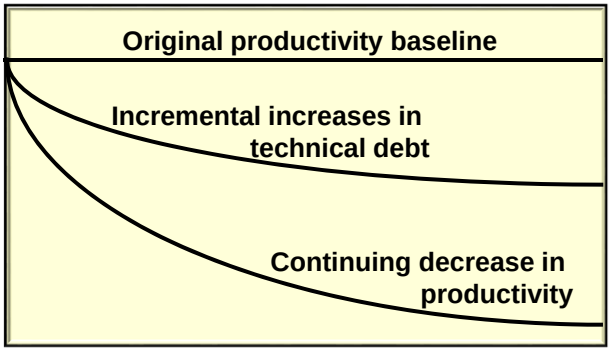
Rethinking Productivity Measurement

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

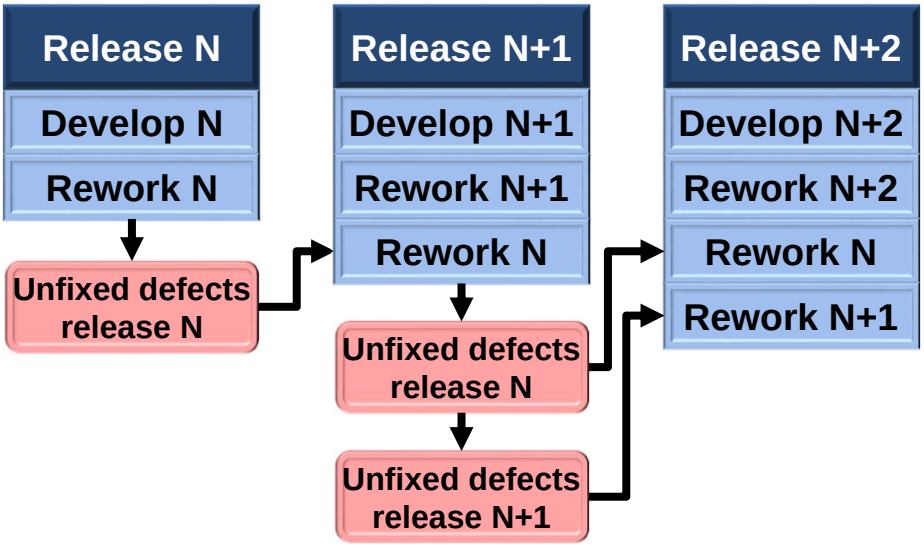
Productivity baseline —

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

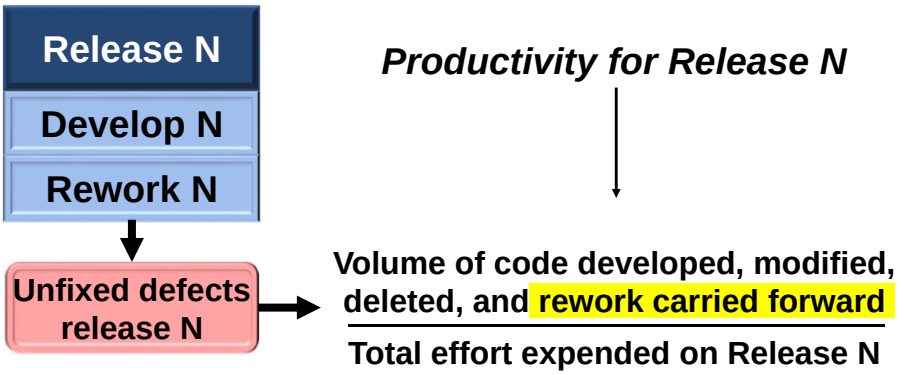
... unless you take action



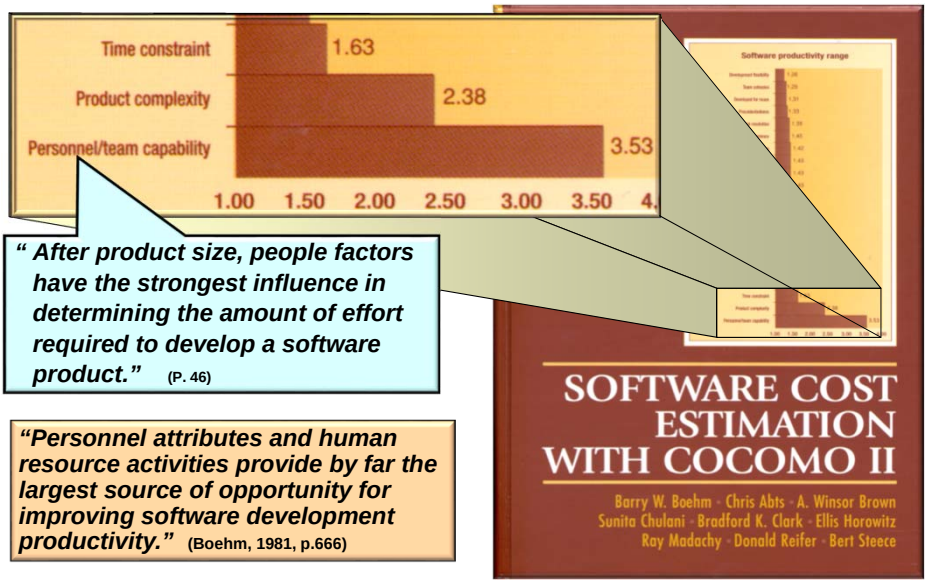
Technical Debt = Carry-forward Rework



Adjust Productivity for Technical Debt



What Predominates Software Variation

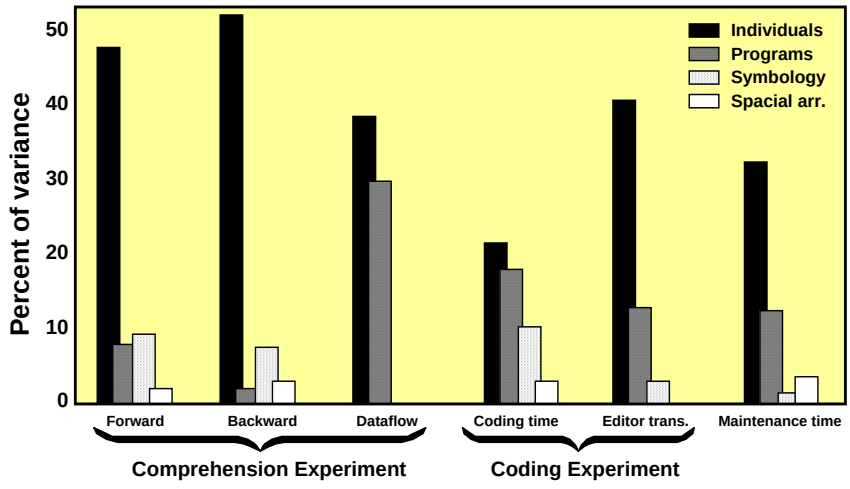


" After product size, people factors have the strongest influence in determining the amount of effort required to develop a software product." (P. 46)

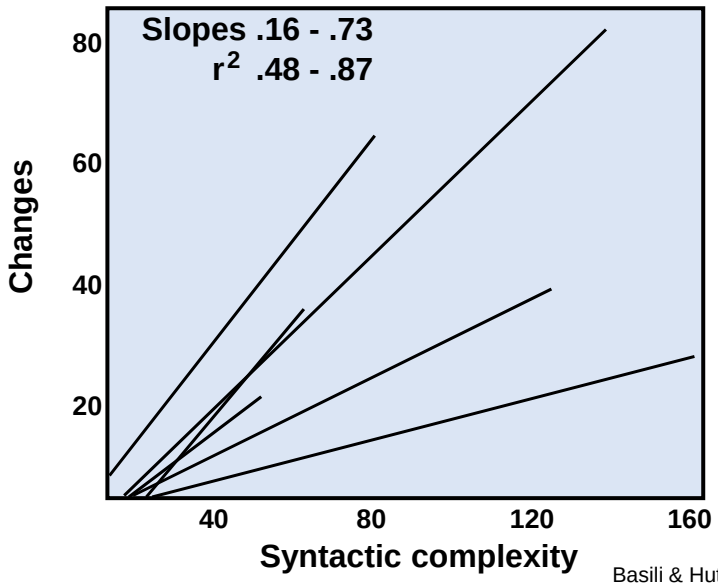
"Personnel attributes and human resource activities provide by far the largest source of opportunity for improving software development productivity." (Boehm, 1981, p.666)

Boehm, et. al (2000)

Programmer Variation Swamps Everything



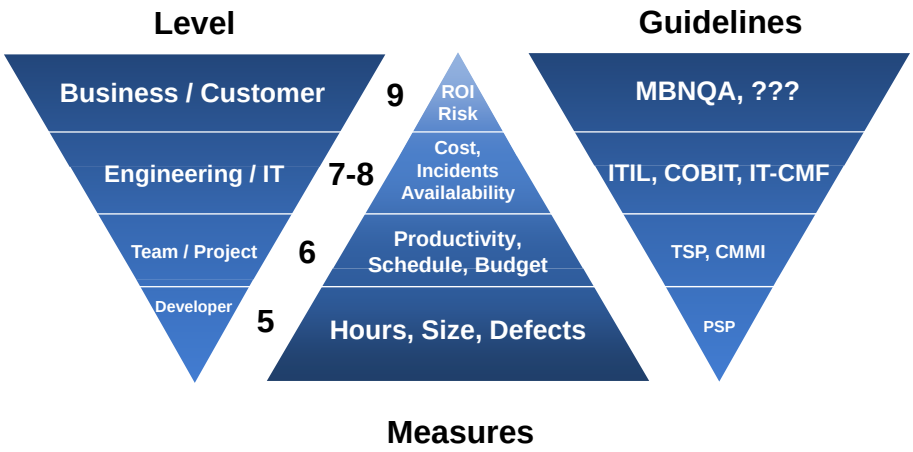
Complexity Profiles for Individual Developers



Basili & Hutchens (1983)

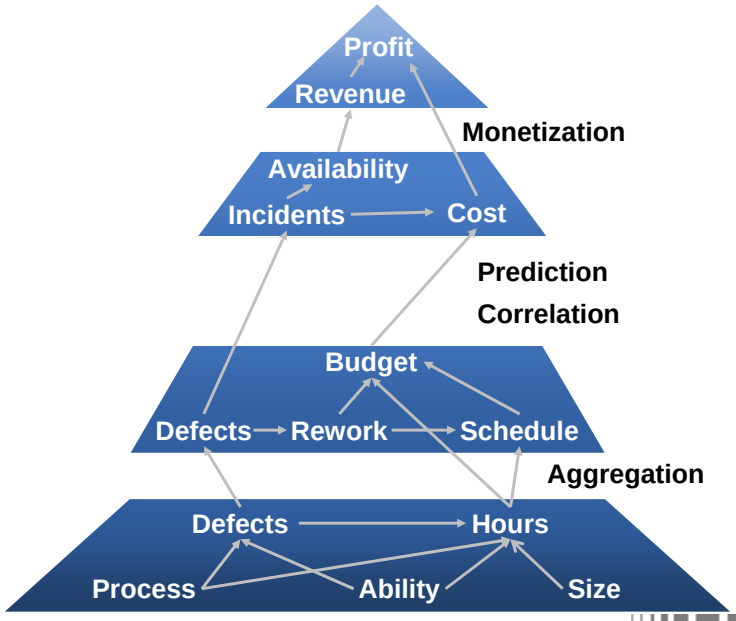
CAST

The Measurement Stack



CAST

Value Transitions in the Measurement Stack



Consortium for IT Software Quality

- Co-sponsored by SEI and the Object Management Group (OMG)
- 24 original member companies
- Objective to standardize code level measurement of software attributes
- Automated Function Points now a supported specification of OMG

www.it-cisq.org — Membership Is Free