Measuring Software Testability Modulo Test Quality

Valerio Terragni

Pasquale Salza

Mauro Pezzè

Università della Svizzera italiana

University of Zurich[UZH]

Università della Svizzera italiana

Schaffhausen Institute of Technology

ICPC 2020

International Conference on Program Comprehension 2020

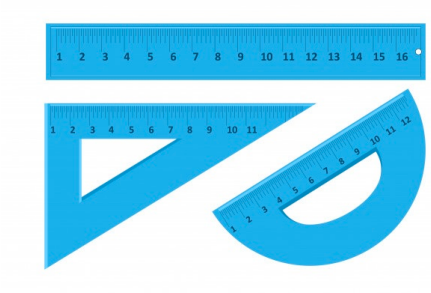
Software Testability

the degree to which a software system or component under test supports its own testing

IEEE 610.12-1990 and ISO/IEC 9126 standards

Measure Software Testability

Software Metrics



LOCs
methods
OO metrics
….

correlation

Test Effort

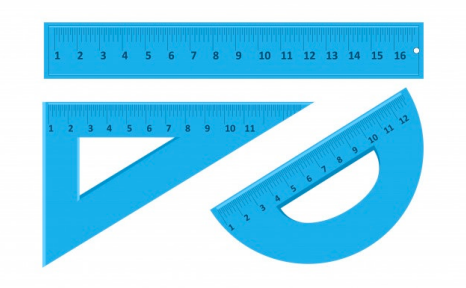
test cases
test assertions
LOC test cases
…

Magiel Bruntink and Arie van Deursen. 2006. An Empirical Study Into Class Testability. Journal of Systems and Software 79, 9 (2006), 1219–1232

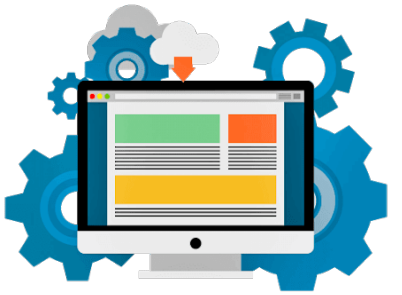
Mourad Badri and Fadel Toure. 2012. Empirical Analysis of Object-Oriented Design Metrics for Predicting Unit Testing Effort of Classes. Journal of Software Engineering and Applications 5, 7 (2012), 513

Limitations of the State of the Art

Ignore the test quality



limited number of software metrics



small number of software projects (at most 8)

The Importance of Test Quality

Class A

LOC 1K

tests 10 (test effort)

Class B

LOC 1K

tests 30 (test effort)

Which class has a higher degree of testability?

line coverage **5%**

line coverage **90%**

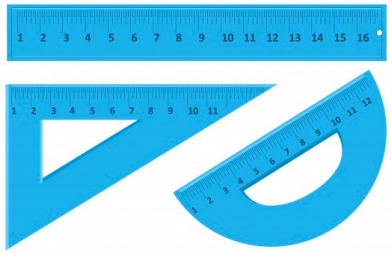
Ignoring the quality of the test suite produces imprecise correlation results

Measure Software Testability

Bruntink and van Deursen JSS 2006

5 Java projects with **similar code coverage**

Software Metrics



LOCs
methods
OO metrics
….

correlation

Test Effort

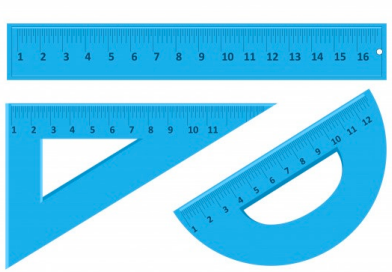
test cases
test assertions
LOC test cases
…

Measure Software Testability

Software Metrics

In this paper

1,186 Java Project **with different test quality**



LOCs
methods
OO metrics
….

correlation

Test Effort

test cases
test assertions
LOC test cases
…

In this paper

Normalize test effort with respect test quality

Normalization

Class A

LOC 1K

tests 10 (test effort)

line coverage **5%**

Class B

LOC 1K

tests 30 (test effort)

line coverage **90%**

$$\frac{\text{normalized test-effort value}}{\text{target test-quality value}} = \frac{\text{actual test-effort value}}{\text{actual test-quality value}}$$

Normalization

Class A

LOC 1K

tests 10 (test effort)

line coverage 5%

Class B

LOC 1K

tests 30 (test effort)

line coverage 90%

$$X = \frac{\text{normalized test-effort value}}{\text{target test-quality value}} = \frac{\text{actual test-effort value}}{\text{actual test-quality value}}$$

$$X = 100\% = 200 \text{ \# tests (test effort)}$$

Normalization

Class A

LOC 1K

tests 200 (test effort)

line coverage **5%**

Class B

LOC 1K

tests 30 (test effort)

line coverage 90%

$$X \cdot \frac{\text{normalized test-effort value}}{\text{target test-quality value}} = \frac{\text{actual test-effort value}}{\text{actual test-quality value}}$$

100%

X = 33 # tests (test effort)

Normalization

Class A

LOC 1K

tests **200** (test effort)

line coverage **5%**

Class B

LOC 1K

tests **33** (test effort)

line coverage **90%**

$$\frac{\text{normalized test-effort value}}{\text{target test-quality value}} = \frac{\text{actual test-effort value}}{\text{actual test-quality value}}$$

Step 1 – Subjects Collection

GitHub

Java open-source projects

> 50 stars

> 1 fork

clone

OR

Gradle

Apache maven

AND

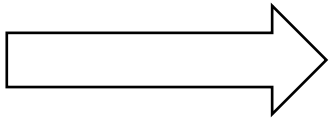
JUnit

build

1,186 Java projects

Step 2 – Metric Computation

1,186
Java projects

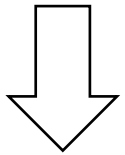


identify pairs of
Class and Test Class

Connector

ConnectorTest

naming conventions



9,861 pairs Class and Test Class

28 class metrics

6 test effort metrics
and
3 test quality metrics

28 class metrics

Size	Lines of Code (LOC)
	Number of Bytecode Instructions (NBI)
	Lines of Comment (LOCCOM)
	Number of Public Methods (NPM)
	Number of STAtic Method (NSTAM)
	Number of Fields (NOF)
	Number of STAtic Fields (NSTAF)
	Number of Method Calls (NMC)
	Number of Method Calls Internal (NMCI)
	Number of Method Calls External (NMCE)
Complexity	Weighted Methods per Class (WMC)
	Average Method Complexity (AMC)
	Response For a Class (RFC)
Inheritance	Depth of Inheritance Tree (DIT)
	Number of Children (NOC)
	Measure of Functional Abstraction (MFA)
Coupling	Coupling Between Object classes (CBO)
	Inheritance Coupling (IC)
	Coupling Between Methods (CBM)
	Afferent Coupling (Ca)
	Efferent Coupling (Ce)
Cohesion	Lack of Cohesion in Methods (LCOM)
	Lack of Cohesion Of Methods (LCOM3)
	Cohesion Among Methods in class (CAM)
Encapsulation	Data Access Metrics (DAM)
	Number of PRIvate Fields (NPRIF)
	Number of PRIvate Methods (NPRIM)
	Number of PROtected Methods (NPROM)

6 test effort metrics

TEST – Lines Of Code (T-LOC)
TEST – Number Of Tests (T-NOT)
TEST – Number Of Assertions (T-NOA)
TEST – Number of Method Calls (T-NMC)
TEST – Weighted Methods per Class (T-WMC)
TEST – Average Method Complexity (T-AMC)

3 test quality metrics

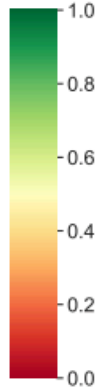
Line coverage
Branch coverage
Mutation score

Step 3 – Spearman Correlation

168 pairs of class and test effort metrics

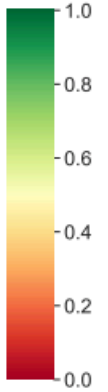
before normalization

	LOC	NBI	LOCCOM	NPM	NSTAM	NOF	NSTAF	NMC	NMCI	NMCE	WMC	AMC	RFC	DIT	NOC	MFA	CBO	IC	CBM	CA	CE	LCOM	LCOM3	CAM	DAM	NPRIF	NPRIM	NPROM
T-LOC	0.53	0.45	0.49	0.25	0.16	0.25	0.16	0.44	0.43	0.34	0.39	0.27	0.45	0.0088	0.027	0.022	0.36	0.04	0.048	0.066	0.35	0.25	0.026	0.32	0.1	0.23	0.27	0.12
T-NOT	0.35	0.33	0.29	0.26	0.18	0.15	0.13	0.32	0.29	0.26	0.35	0.14	0.33	0.078	0.025	0.082	0.17	0.0012	0.0022	0.11	0.14	0.25	0.038	0.28	0.0098	0.13	0.23	0.034
T-NOA	0.3	0.3	0.26	0.26	0.11	0.16	0.13	0.24	0.23	0.19	0.28	0.15	0.24	0.01	0.038	0.014	0.041	0.01	0.018	0.089	0.014	0.18	0.052	0.21	0.017	0.088	0.086	0.022
T-NMC	0.47	0.46	0.41	0.28	0.17	0.26	0.15	0.46	0.44	0.36	0.42	0.25	0.48	0.014	0.013	0.029	0.35	0.036	0.044	0.069	0.33	0.26	0.029	0.36	0.1	0.24	0.26	0.13
T-WMC	0.37	0.35	0.31	0.23	0.18	0.17	0.14	0.34	0.32	0.28	0.35	0.17	0.35	0.072	0.023	0.075	0.23	0.014	0.0088	0.089	0.21	0.25	0.023	0.29	0.028	0.15	0.26	0.055
T-AMC	0.33	0.34	0.31	0.13	0.012	0.19	0.048	0.31	0.33	0.22	0.22	0.29	0.31	0.09	0.00029	0.071	0.25	0.078	0.086	0.0034	0.26	0.099	0.059	0.18	0.094	0.17	0.11	0.15



after normalization (mutation score)

	LOC	NBI	LOCCOM	NPM	NSTAM	NOF	NSTAF	NMC	NMCI	NMCE	WMC	AMC	RFC	DIT	NOC	MFA	CBO	IC	CBM	CA	CE	LCOM	LCOM3	CAM	DAM	NPRIF	NPRIM	NPROM
T-LOC	0.72	0.67	0.65	0.47	0.26	0.38	0.26	0.63	0.55	0.53	0.61	0.35	0.68	0.031	0.024	0.053	0.45	0.11	0.12	0.062	0.44	0.39	0.0073	0.54	0.11	0.33	0.32	0.12
T-NOT	0.58	0.58	0.49	0.47	0.29	0.29	0.24	0.54	0.45	0.47	0.57	0.25	0.58	0.088	0.023	0.1	0.31	0.08	0.083	0.1	0.28	0.39	0.045	0.51	0.038	0.24	0.31	0.061
T-NOA	0.39	0.4	0.35	0.34	0.16	0.21	0.17	0.33	0.3	0.27	0.37	0.19	0.34	0.012	0.031	0.02	0.097	0.036	0.045	0.084	0.07	0.25	0.062	0.3	0.017	0.12	0.12	0.04
T-NMC	0.67	0.67	0.59	0.47	0.27	0.38	0.24	0.65	0.57	0.54	0.62	0.33	0.69	0.029	0.014	0.052	0.45	0.1	0.11	0.063	0.44	0.39	0.014	0.55	0.12	0.33	0.33	0.14
T-WMC	0.6	0.6	0.52	0.46	0.29	0.32	0.25	0.57	0.48	0.5	0.59	0.27	0.62	0.084	0.02	0.099	0.36	0.079	0.083	0.08	0.34	0.4	0.031	0.53	0.057	0.27	0.33	0.074
T-AMC	0.53	0.56	0.48	0.35	0.12	0.31	0.15	0.5	0.45	0.4	0.44	0.35	0.53	0.061	0.0068	0.035	0.34	0.15	0.15	0.0023	0.35	0.25	0.046	0.4	0.12	0.27	0.19	0.16



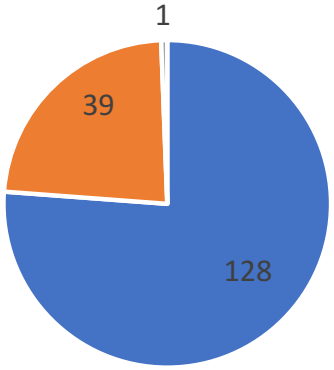
Summary of Findings

Normalization increases the correlation

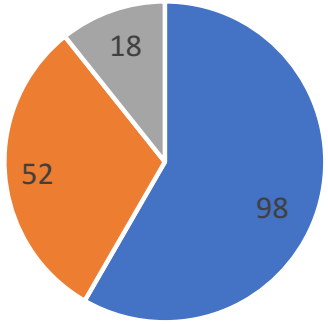
168 pairs of class and test effort metrics

weak corr <= 0.3 0.3 < **medium corr** < 0.5 **strong corr** >= 0.5

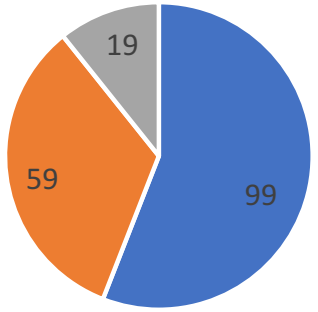
no normalization



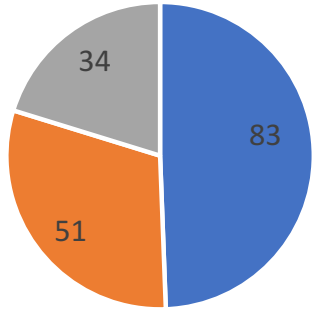
normalization by line coverage



normalization by branch coverage



normalization by mutation score



new result

Size, Complexity, Coupling and Cohesion are the OO design properties that most correlates with test effort

Software Testability

the degree to which a software system or component under test supports its own testing

IEEE 610.12-1990 and ISO/IEC 9126 standards

Step 1 – Subjects Collection

GitHub

Java open-source projects

> 50 stars ★
> 1 fork

clone

OR Gradle / Apache Maven

AND JUnit

build ✓

1,186 Java projects

Normalization

Class A

LOC 1K

tests **200** (test effort)

line coverage **5%**

Class B

LOC 1K

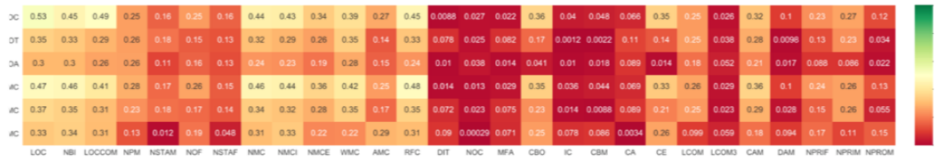
tests **33** (test effort)

line coverage **90%**

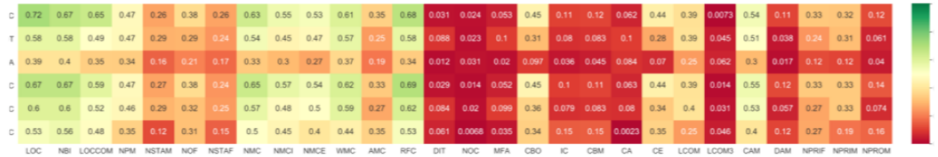
$$\frac{\text{normalized test-effort value}}{\text{target test-quality value}} = \frac{\text{actual test-effort value}}{\text{actual test-quality value}}$$

Step 3 – Spearman Correlation

before normalization



after normalization (mutation score)



Replication Package

https://doi.org/10.5281/zenodo.3740499