

Assessing Incremental Testing Practices and Their Impact on Project Outcomes

Ayaan M. Kazerouni, Clifford A. Shaffer, Stephen H. Edwards, Francisco Servant

Department of Computer Science, Virginia Tech

SIGCSE 2019, Minneapolis, MN Thursday, February 28, 2018



Study Context

- Third year (post-CS2) Data Structures & Algorithms course
- 157 students
- 4 assignments
 - Median 1.4 kLOC
 - 3-4 weeks long
- 415 implementations (unbalanced)



Contributions

- Family of metrics to assess incremental testing
- Empirical study
 - 1. How does the **balance** of testing effort relate to project outcomes?
 - 2. How does the **sequence** of testing effort relate to project outcomes?



Better Feedback on Process



[1] Quantifying Incremental Development Practices and Their Relationship to Procrastination. Ayaan M. Kazerouni, Stephen H. Edwards, and Clifford A. 4



Assessing Incremental Testing



Synthetic example: sequence of developer activity







Solution code	Test code	
		Method A
		Method B
		Method C
		Any method



















Motivating Example from Fall 2016



Work Session #



Empirical Study



Data Collection

Edit Event

Type: Edit

Snapshot Id: 23479b3

400+ project implementations

Time: 1477672862

😑 😑 📓 eclipse-workspa	- org.webcat.eclipse.projectlink/src/org/webcat/eclipse/deveventtracker/sensorbase/SensorBaseClient.java - Eclipse	
😁 • 🗟 🕼 🔹 🔌 🂁 🕸 • 💽 • 💁 🗑 • 🖉	🖨 🖋 • : 🍄 🅖 🕼 📓 🗊 : 🖗 • 👌 • 🗇 • 🗇 • 🗇 •	4
😫 Package Explorer 🕴 隆 Project Explorer 📟	🖸 SensorBaseClient.java 🛛 🖉 EclipseSensor.java 📝 PostToServerTimerTask.java 🗍 EclipseSensorConstants.java	, 6
E 2 2	317 */	
rg.webcat.eclipse.cs2114	3180 public synchronized void putSensorData(SensorData data)	
irg.webcat.eclipse.projectlink [eclipse-plugins-importer-exporte	319 throws SensorBaseClientException	
🙀 Referenced Libraries	200 1 if (anthony Transmott)	
JRE System Library [J2SE-1.5]	321 tr (gerbanoseiver(j) 327 f	
🙀 Plug-in Dependencies	323 // Retrieve the stored user UUD from preferences, or from the	
👼 src	324 // server if	
Image: Barbar Bar Barbar Barbar Ba	325 // not present.	
# # org.webcat.eclipse.deveventtracker	326 String userUuid = retrieveUser(getEmail()).toString();	
BuffTransTimertask.java	327	
EclipseSensor.java	338 String studentProjectula = PetrievestudentProject(
EclipseSensorConstants.java	325 data.getrojectar().tostring();	
PostToServerTimerTask.java	331 String requestString = "postSensorDate7studentProject[aide"	
SensorShellWrapper.java	332 + studentProjectUuid + "&userUuid + "&time-"	
StateChangeTimerTask.java	333 + data.timestomp + "&runtime=" + data.runtime + "&tool="	
Image: Barrier Barr	334 + data.tool + "&sensorDataType=" + data.sensorDataType	
eta org.webcat.eclipse.deveventtracker.event	335 + "&uri=" + data.uri;	
# B org.webcat.eclipse.deveventtracker.sensorbase	356 int counter = 1;	
Properties.java	222 http://www.f	
Property.java	330 requestString += "Ange" + counter + "a" + URLEncoder_encode(p.getKev(), "UTE-8"):	
SensorBaseClient.java	340 requestString += "åvalue" + counter + "=" + URLEncoder.encode(p.getValue(), "UTF-8");	
SensorBaseClientException.java	341 counter++;	
SensorData.java	342 } catch (UnsupportedEncodingException e) {	
SensorDatas.java	343 Activator.getDefault().log(e);	
# org.webcat.eclipse.deveventtracker.sensorshell	344 J	
Image: Book and Bo	245 J 245 Persenere serenere – mekePersuerk/Makhad (CT, sesuerkStation, pull))	
Image: Barbard Barb	37 if (Inesonase and Status() (Success())	
Image: Barrier Barr	348	
#gorg.webcat.eclipse.projectlink.exporter	349 throw new SensorBaseClientException(response.getStatus());	
Image: Barrier Barr	350 }	
#g org.webcat.eclipse.projectlink.importer	351 3	
Image:	352 }	
#g org.webcat.eclipse.projectlink.preferences		
Image: Barrier Barr	🕐 Problems 🕴 🖉 Javadoc 😥 Declaration 🔗 Search 🖵 Console 🔅 🦻 🌚 🖤	° C
tions :	Dierrors, 22 warnings, Diothers	
ib lb	Description A Besource Path Location Type	
META-INF	▶ (Warnings (22 Items)	
build.properties		
plugin.xml		
epodriller		
ensordata-repo-mining		

Туре	Size	Time
Change in method insertFront	+5	12:41:02
Change in method getSize	+1	12:41:02
Change in test for insertFront	+3	12:41:02

Ц



Study Design

- Fixed effects: 5 measures of testing effort
- Random effects: students, assignments
- Outcome variables:
 - Correctness, measured by the percentage of reference tests passed
 - Code coverage achieved by the student's own test suite

Mixed effects model: repeated measures for each student, and for each assignment.



Results



Project-wide Overall Testing Effort

Solution Code Effort

Test Code Effort

Expectation: Positive relationship with correctness and code coverage.

Correctness		Code Coverage		
Regression estimate	p	Regression estimate	p	
0.30	< 0.001 *	0.23	< 0.001 *	

- Implementations with a higher project-wide testing effort achieved:
 - Higher semantic correctness
 - Higher code coverage



Project-wide per-Session Testing Effort



Expectation: Positive relationship with correctness and code coverage.

Correctness		Code Coverage		
Regression estimate	p	Regression estimate	p	
0.30	0.005 *	0.12	0.008 *	

- Implementations with higher testing effort within each work session achieved
 - Higher semantic correctness
 - Higher code coverage



Motivating Example (Reprise)



Work Session #



Method-specific Sequence of Testing Effort



Expectation: Positive or no relationship with project outcomes.

Correctness		Code Coverage		
Regression estimate	р	Regression estimate	р	
	0.10	-0.06	< 0.001 *	

- Implementations where a higher proportion of testing for a method was done before the method was finalised, achieved:
 - No significant change in correctness
 - Lower code coverage



Putting It All Together





Closing Remarks



Summary

- Quantified test-writing practices
- Empirical study
 - Higher testing effort is good (whole project and per-method)
 - Higher testing effort per work session is good
 - No such relationship on a per-method basis
 - Higher testing effort before finalizing relevant solution code
 - Does not lead to improved correctness
 - Negative relationship with code coverage
- Next step: Design and deploy automated interventions for continuous feedback

ayaan@vt.edu https://github.com/ayaankazerouni/incremental-testing



Bonus Material



Per-session Testing Effort: Distribution



Project-wide per-Session Balance of Test Effort (PSB)



Method-specific Sequence of Testing Effort





Method-specific per-Session Testing Effort



Expectation: Positive relationship with both project outcomes.

Correctness		Condition Coverage		
Regression estimate	p	Regression estimate	p	
	0.10	0.23	< 0.001 *	

- Higher testing effort per-method, per-session achieved:
 - Higher condition coverage
 - No significant change in



Mixed effects model (Process)

	Correctness		Code Coverage	
Metric	Regression estimate	р	Regression estimate	р
Testing per-Session	0.30	0.005 *	0.12	0.008 *
Testing per-Session per-Method		0.10	0.09	0.002 *
Sequence of testing		0.62	-0.06	0.02 *



Mixed effects model (Overall)

	Correctness		Code Coverage	
Metric	Regression estimate	р	Regression estimate	Р
Testing	0.30	< 0.001 *	0.23	< 0.001 *
Testing per-Method		0.12		0.41
Testing per-Session		0.83		0.97 *
Testing per-Session, per-Method		0.97	0.08	0.01 *
Sequence of testing		0.74	-0.06	0.03 *

Fixed effects $R^2 = 5\%$

Fixed effects $R^2 = 10\%$