



Durchführung einer Entwicklerstudie zum Ermitteln von Quality Smells und deren Beseitigung auf Android-Systemen

Verteidigung Großer Beleg

Martin Brylski

Dresden, 27.3.2014

Quality Smells

“ Quality Smells? Zu meinen Studienzeiten nannte man das noch „nichtfunktionale Anforderungen“. Tsss, tsss...(-: ”

Andreas Schildbach (Öffi)

Quality Smells

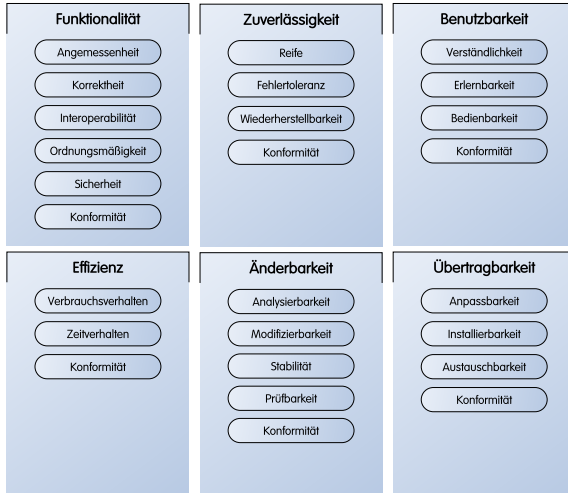
“ Quality Smells? Zu meinen Studienzeiten nannte man das noch „nichtfunktionale Anforderungen“. Tsss, tsss...(-: ”

Andreas Schildbach (Öffi)

Definition

Quality Smells = Code Smells
+ Refactoring [Fow99, ASH13, Par94]
+ Softwarequalität

Softwarequalitäten



Messen von Softwarequalitäten

AUTOMATISIERBARKEIT		OBJEKTIV	SUBJEKTIV
Schwierigkeit ↑		Energieeffizienz	Verständlichkeit
		Speichereffizienz	Erlernbarkeit
		Recheneffizienz	Bedienbarkeit
			Sicherheit
		Antwortzeit	Analysierbarkeit
		Korrektheit	Komplexität
		Startzeit	Benutzerkonformität

Momentaner Stand

“ In our experience no set of metrics rivals informed human intuition. ”
Fowler

Organisation

- AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis von [Bro98]

Anforderungsanalyse

- Use cases: patterns and blueprints von [OP05]

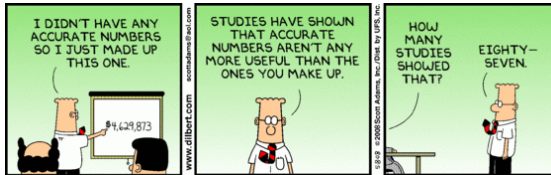
Softwarearchitektur

- Patterns of Enterprise Application Architecture von [Fow99]
- Design Patterns: Elements of Reusable Object-Oriented Software von [GHJV94]
- Identifying Architectural Bad Smells von [GPEM09]

Implementierung

- Refactoring: Improving the Design of Existing Code von [FBB⁺12]
- New Software Performance AntiPatterns: More Ways to Shoot Yourself in the Foot von [SW03]
- Implementation Pattern von [Bec07]

Studie

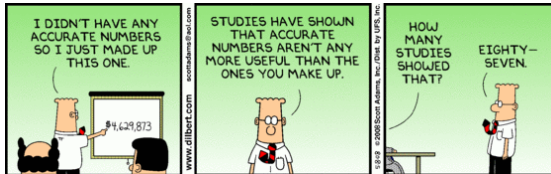


- Observation bei Entwicklung
- Interviews & Umfragen
- Aggregation von Dokumenten

Quellcode, Dokumentation, Präsentationen, Bug-Tracker, Foren, Blogs

- Stackoverflow
- Programmers Stackexchange
- Android Stackexchange
- Android Issues
- Android Developer in Google Groups
- Android Entwicklerdokumentation
- Android Developers Blog
- Android Design Patterns
- Google IO Vorträge
- Pro Android Apps Performance Optimization von [Gui12]

Studie

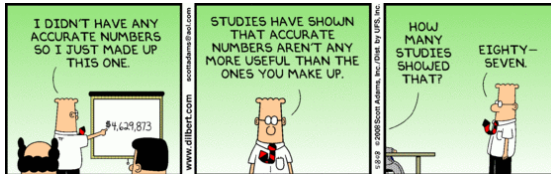


- Observation bei Entwicklung
- Interviews & Umfragen
- Aggregation von Dokumenten

Quellcode, Dokumentation, Präsentationen, Bug-Tracker, Foren, Blogs

- Stackoverflow
- Programmers Stackexchange
- Android Stackexchange
- Android Issues
- Android Developer in Google Groups
- Android Entwicklerdokumentation
- Android Developers Blog
- Android Design Patterns
- Google IO Vorträge
- Pro Android Apps Performance Optimization von [Gui12]

Studie

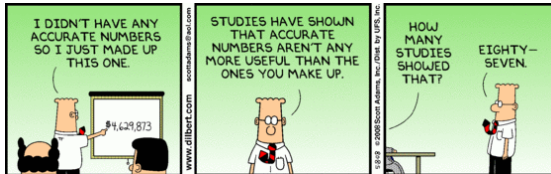


- Observation bei Entwicklung
- Interviews & Umfragen
- Aggregation von Dokumenten

Quellcode, Dokumentation, Präsentationen, Bug-Tracker, Foren, Blogs

- Stackoverflow
- Programmers Stackexchange
- Android Stackexchange
- Android Issues
- Android Developer in Google Groups
- Android Entwicklerdokumentation
- Android Developers Blog
- Android Design Patterns
- Google IO Vorträge
- Pro Android Apps Performance Optimization von [Gui12]

Studie

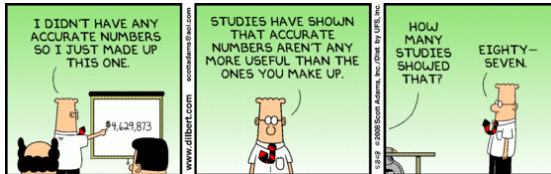


- Observation bei Entwicklung
- Interviews & Umfragen
- Aggregation von Dokumenten

Quellcode, Dokumentation, Präsentationen, Bug-Tracker, Foren, Blogs

- Stackoverflow
- Programmers Stackexchange
- Android Stackexchange
- Android Issues
- Android Developer in Google Groups
- Android Entwicklerdokumentation
- Android Developers Blog
- Android Design Patterns
- Google IO Vorträge
- Pro Android Apps Performance Optimization von [Gui12]

Studie

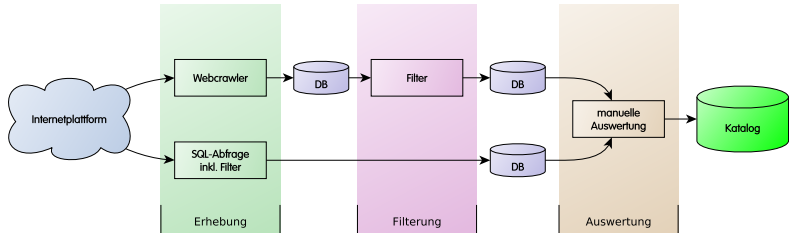


- Observation bei Entwicklung
- Interviews & Umfragen
- Aggregation von Dokumenten

Quellcode, Dokumentation, Präsentationen, Bug-Tracker, Foren, Blogs

- Stackoverflow
- Programmers Stackexchange
- Android Stackexchange
- Android Issues
- Android Developer in Google Groups
- Android Entwicklerdokumentation
- Android Developers Blog
- Android Design Patterns
- Google IO Vorträge
- Pro Android Apps Performance Optimization von [Gui12]

Studie



Webcrawler Scrapy

Filterwörter energy, power, drain, efficient, inefficient, inefficiency, efficiency, energy consumption, power consumption, memory consumption, energy bug, power bug, energy drain, performance, io access, speed up, code quality, clean code, response time, faster, memory, overhead, code review, wake lock, refactoring, refactor, blocking operation, best practice, code smell, bad smell, access time, traffic, reduce, power-saving, power saving, energy-saving, energy saving, time-saving, time saving, low battery, economic, battery, periodic, minimize, consumption, high usage, slow, leak, power management, power save, discharge, uncharge

Ergebnis

QUELLE	NACH FILTERUNG	GEFUNDENE SMELLS
Stackoverflow	1116	3
Programmers Stackexchange	28	1
Android Stackexchange	500	0
Android Issues	466	0
Android Developer in Google Groups	556	1
Android Entwicklerdokumentation	*	2
Android Developers Blog	*	5
Android Design Patterns	*	2
Google IO Vorträge	*	8
Pro Android Apps Performance Optimization	*	5
Sonstige	*	3

* nur manuell durchsucht

Rigid AlarmManager

Name	Rigid AlarmManager
Affected Qualities	efficiency, energy consumption
Context	implementation
Tags	energy, alarm, efficiency
Description	With the use of AlarmManager it is possible that operations can be executed at a sepcific moment in the future. It is possible that several operations got executed. Every AlarmManager-triggered operation wakes up the phone, so the overall use of energy and CPU might be higher, then if bundled together.

```
AlarmManager am = (AlarmManager) ctx.getSystemService(Context.ALARM_SERVICE);

Intent intent = new Intent(context, MyService.class);
PendingIntent pendingIntent = PendingIntent.getService(ctx, 0, intent, 0);

long interval = DateUtils.MINUTE_IN_MILLIS * 30;
long firstWake = System.currentTimeMillis() + interval;

am.setRepeating(AlarmManager.RTC_WAKEUP, firstWake, interval, pendingIntent);
```

Refactoring

Name	Set Inexact Alarmmanager
Resolved Qualities	efficiency, energy consumption
Affected Qualities	–
Description	To ensure that the system is able to bundle several updates together, it is recommended to use:

```
AlarmManager.setInexactRepeating(int type, long triggerAtMillis, long  
intervalMillis, PendingIntent operation)
```

References	http://www.google.com/events/io/2009/sessions/CodingLifeBatteryLife.html http://developer.android.com/reference/android/app/AlarmManager.html
Related Smells	Bulk data transfer on slow network, Data Transmission Without Compression, Durable WakeLock, Early Resource Binding, Illicit data transfer, Prohibited data transfer

Katalog

Android SmellsCatalogueAbout

SearchSubmit

Listing of smells (30)

- Bulk Data Transfer On Slow Network
- Data Transmission Without Compression
- Debuggable Release
- Dropped Data
- Durable WakeLock
- Early Resource Binding
- Inefficient Data Structure
- Inefficient SQL Query
- Inefficient Data Format And Parser
- Internal Getter/Setter
- Interrupting From Background
- Leaking Inner Class
- Leaking Thread
- Member-Ignoring Method
- Nested Layout

- Network & IO Operations In Main Thread
- No Low Memory Resolver
- Not Descriptive UI
- Overdrawn Pixel
- Prohibited Data Transfer
- Public Data
- Rigid AlarmManager
- Set Config Changes
- Slow Loop
- Tracking Hardware Id
- Uncached Views
- Unclosed Closable
- Uncontrolled Focus Order
- Unnecessary Permission
- Untouchable

Table of smells and affected qualities

Smell	Accessibility	Efficiency	Energy Efficiency	Memory Efficiency	Security	Stability	Startup Time	User Conformity	User Experience
Bulk Data Transfer On Slow Network			✗					✗	
Data Transmission Without Compression			✗						

http://www.modelrefactoring.org/smell_catalog/

Tool-Support

Refactory [Rei10, RA13]

modellgestütztes Werkzeug zum Erkennen von Smells und Spezifizieren von Refactorings

```
1 package org.emftext.refactoring.languages.java.smell.android.test;
2
3 import android.app.Activity;
10
11 public class RigidAlarmManagerTest extends Activity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16
17         AlarmManager am = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
18
19         Intent intent = new Intent(this, InterruptingFromBackgroundServiceTest.class);
20         PendingIntent pendingIntent = PendingIntent.getService(this, 0, intent, 0);
21
22         long interval = DateUtils.MINUTE_IN_MILLIS * 30;
23         long firstWake = System.currentTimeMillis() + interval;
24
25         am.setRepeating(AlarmManager.RTC_WAKEUP, firstWake, interval, pendingIntent);
26     }
27 }
28
```

🔔 The inexact alarm manager should be used if exact execution is not needed.
1 quick fix available:
[Replace exact with inexact AlarmManager](#)
Press 'F2' for focus

Auswertung & Ausblick

- **Sammeln und Generalisieren von QS**
- Ontologie (für Design Patterns [Kam07])
- alternatives Qualitätsmodell (aktivitätenbasiertes SQM [De09])
- vereint verschiedene Sichten auf Softwarequalitäten
- Quality Smell mit Messung verknüpfen

Auswertung & Ausblick

- Sammeln und Generalisieren von QS
- Ontologie (für Design Patterns [Kam07])
- alternatives Qualitätsmodell (aktivitätenbasiertes SQM [De09])
- vereint verschiedene Sichten auf Softwarequalitäten
- Quality Smell mit Messung verknüpfen

Auswertung & Ausblick

- Sammeln und Generalisieren von QS
- Ontologie (für Design Patterns [Kam07])
- alternatives Qualitätsmodell (aktivitätenbasiertes SQM [De09])
- vereint verschiedene Sichten auf Softwarequalitäten

				Wartbarkeit			
				Analyse		Implementierung	
				Concept Location	Impact Analyse	Erstellung	Änderung
Produkt	Software	Klasse	Strukturiertheit	+	+		+
		Methode	Strukturiertheit		+		+
			Redundanz		-		-
	Dokumentation	Glossar	Vollständigkeit	+		+	
		Kommentar	Prägnanz	+		+	+

- Quality Smell mit Messung verknüpfen

Auswertung & Ausblick

- Sammeln und Generalisieren von QS
- Ontologie (für Design Patterns [Kam07])
- alternatives Qualitätsmodell (aktivitätenbasiertes SQM [De09])
- vereint verschiedene Sichten auf Softwarequalitäten

				Wartbarkeit			
				Analyse		Implementierung	
				Concept Location	Impact Analyse	Erstellung	Änderung
Produkt	Software	Klasse	Strukturiertheit	+	+		+
		Methode	Strukturiertheit		+		+
			Redundanz		-		-
	Dokumentation	Glossar	Vollständigkeit	+		+	
		Kommentar	Prägnanz	+		+	+

- Quality Smell mit Messung verknüpfen

Literaturverzeichnis I



Paris Avgeriou, Michael Stal, and Rich Hilliard.
Architecture Sustainability.
IEEE Software, 30(6):40–44, November 2013.



Kent Beck.
Implementation Pattern.
Pearson Education, 2007.



William J. Brown.
AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis.
Wiley, 1998.



Florian Deiß enböck.
Kontinuierliches Qualitäts-Controlling langlebiger Softwaresysteme.
Ausgezeichnete Informatikdissertationen, 2009.



Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts.
Refactoring: Improving the Design of Existing Code.
Addison-Wesley, 2012.



Martin Fowler.
Refactoring: Improving the Design of Existing Code.
Addison-Wesley, 1999.

Literaturverzeichnis II



Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.
Design Patterns: Elements of Reusable Object-Oriented Software.
Pearson Education, 1994.



J. Garcia, D. Popescu, G. Edwards, and N. Medvidovic.
Identifying Architectural Bad Smells.
In 13th European Conference on Software Maintenance and Reengineering, 2009.



Hervé Guihot.
Pro Android Apps Performance Optimization.
Apress, 2012.



Holger Kampffmeyer.
Formalization of Design Patterns by Means of Ontologies.
2007.



Gunnar Övergaard and Karin Palmkvist.
Use cases: patterns and blueprints.
Addison-Wesley, 2005.



David Lorge Parnas.
Software aging.
pages 279–287, May 1994.

Literaturverzeichnis III



Jan Reimann and Uwe Aßmann.

Quality-Aware Refactoring For Early Detection And Resolution Of Energy Deficiencies.
In Proceedings of 4th International Workshop on Green and Cloud Computing Management, 2013.



Jan Reimann.

Generisches Modellrefactoring für EMFText.
Master's thesis, Technische Universität Dresden, 2010.



Connie U. Smith and Lloyd G. Williams.

More New Software Performance Antipatterns: Even More Ways to Shoot Yourself in the Foot.
In Computer Measurement Group Conference 2003, pages 717–725. Computer Measurement Group, 2003.