

Qualitätssicherung

Die Qualität des Produkts wurde durch eine genaue Anforderungsanalyse, Beschreibung einzelner Tasks, Durchführung von Reviews und Produkttests gewährleistet.

Verifikation der Anforderung

Durch das Framework Scrum wurden die Anforderungen an die App in User-Stories zerlegt und mit den Teammitgliedern besprochen.

Umfangsbestimmung von Quellcode

- Lines of Code
- Halstead
- McCabe

Automatisierung

- Regressionstests

Für Continuous Integration wurde Jenkins verwendet. Damit konnten täglich Builds erstellt werden; zusätzlich wurden Junit-Tests ausgeführt.

Testfall

Test-Spezifikationen dienten als Basis bei der Erstellung von Testfällen.

Verwendete Testmethoden

Black-Box-Test

Das Produkt wurde durch spezifizierte Use-Cases getestet und der Input sowie der Output wurden genau dokumentiert. Fehler wurden behoben, bis das Ergebnis so ausfiel, wie erwartet.

White-Box-Test

Testfälle wurden direkt auf Basis des Codes erzeugt. Für diese Tests wurde Junit verwendet.

Reviews

Reviews wurden wöchentlich durchgeführt.

Dadurch wurde die Qualität erhöht und Fehler frühzeitig erkannt.

Lines of Code

Source File	Total Lines	Source Code Lines	Source Code Lines %	Comment Lines	Comment Lines %	Blank Lines	Blank Lines %
Application Test	13	8	62%	3	23%	2	15%
Bummerl2	59	46	78%	0	0%	13	22%
Bummerl3	67	55	82%	0	0%	12	18%
Bummerl4	70	56	80%	0	0%	14	22%
ChangeNickname	59	43	73%	9	15%	7	12%
GameEnd	71	50	70%	14	20%	7	10%
Karte	194	163	84%	0	0%	31	16%
KartenKomparator	33	29	88%	0	0%	4	12%
Lobby	503	358	71%	63	13%	82	16%
NeuesSpiel	76	57	75%	5	7%	14	18%
Rufspiel	43	35	81%	3	7%	5	12%
Spiel2	1083	793	73%	90	8%	200	18%
Spiel3	1041	858	82%	32	3%	151	15%
Spiel4	846	695	82%	44	5%	107	15%
Spieler	83	64	77%	0	0%	19	23%
Spielfeld2	803	639	80%	26	3%	138	17%
Spielfeld2Client	1094	895	82%	26	2%	173	16%
Spielfeld2Host	1242	1000	81%	39	3%	203	16%
Spielfeld3Client	1518	1267	83%	28	2%	223	15%
Spielfeld3Host	1916	1575	82%	44	2%	297	16%
Spielfeld4Client	793	660	83%	40	5%	93	12%
Spielfeld4Host	913	768	84%	54	6%	91	10%
Spielfeld4Logik	402	348	87%	6	1%	48	12%
Spielregeln	54	38	70%	5	9%	11	20%
Startmenue	118	91	77%	5	4%	22	19%
Statistik	75	53	71%	5	7%	17	23%
WrongGameException	10	6	60%	3	30%	1	10%
Total:	13179	10650	81%	544	4%	1985	15%

Projekt

Lines of Code: 13.179

Source Code Lines: 10.650

Comment Lines: 544

Anzahl alle Programmzeilen (Projektumfang)

Algorithmenimplementierungsgröße

Kommentaranteil des Projektes

Halstead Metrik

Klasse Spieler.java

Operatoren n1: 19

Operanden n2: 41

Operatoren N1: 200

Operanden N2: 296

Programmlänge: $N1+N2 = 60$

Vokabulargröße: $n1+n2 = 496$

Schwierigkeitsgrad (D): 13,85

Programmniveau (L): 4909,10

Testabdeckung

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
WrongGameException		0%		n/a	1	1	2	2	1	1	1	1
Bummer13		37%		0%	6	15	11	29	3	12	0	1
Spiel3		44%		33%	136	187	311	588	16	35	0	1
Bummer12		49%		75%	3	12	7	24	2	10	0	1
Karte		52%		25%	31	41	68	110	12	19	0	1
Bummer14		57%		0%	4	15	4	26	2	13	0	1
Spiel4		71%		59%	95	191	140	501	4	22	0	1
Spiel2		73%		63%	104	241	118	521	6	33	0	1
Rufspiel		95%		94%	1	11	1	22	0	3	0	1
KartenKomparator		100%		94%	1	10	0	16	0	2	0	1
Spieler		100%		n/a	0	15	0	29	0	15	0	1
Total	4.322 of 11.180	61%	540 of 1.140	53%	382	739	662	1.868	46	165	1	11

Größe und Anzahl der Dateien im Projekt

Extension ▲	Count	Size	Size MIN	Size MAX	Size AVG	Lines	Lines MIN	Lines MAX	Lines AVG
bat (BAT files)	1x	2kB	2kB	2kB	2kB	90	90	90	90
bin (BIN files)	5x	9.567kB	0kB	7.955kB	1.913kB	9586	2	7590	1917
config (CONFIG files)	1x	0kB	0kB	0kB	0kB	0	0	0	0
css (CSS files)	4x	7kB	0kB	4kB	1kB	501	13	243	125
docx (DOCX files)	9x	3.347kB	15kB	2.624kB	371kB	26232	95	21284	2914
docx# (DOCX# files)	2x	0kB	0kB	0kB	0kB	2	1	1	1
gitignore (GITIGNORE files)	2x	0kB	0kB	0kB	0kB	8	1	7	4
gradle (GRADLE files)	2x	0kB	0kB	0kB	0kB	20	1	19	10
htm (HTM files)	1x	14kB	14kB	14kB	14kB	310	310	310	310
html (HTML files)	126x	757kB	1kB	62kB	6kB	4274	1	378	33
idx (IDX files)	3x	117kB	4kB	105kB	39kB	927	34	837	309
java (Java classes)	27x	501kB	0kB	77kB	18kB	13179	10	1916	488
js (JS files)	3x	65kB	2kB	59kB	21kB	1758	101	1510	586
lock (LOCK files)	1x	0kB	0kB	0kB	0kB	1	1	1	1
name (NAME files)	3x	0kB	0kB	0kB	0kB	3	1	1	1
ods (ODS files)	5x	352kB	59kB	87kB	70kB	2708	467	668	541
pack (PACK files)	3x	51.133kB	109kB	46.559kB	17.044kB	399458	840	363754	133152
pdf (PDF files)	16x	2.818kB	19kB	699kB	176kB	67705	195	35069	4231
pro (PRO files)	1x	0kB	0kB	0kB	0kB	17	17	17	17
project (PROJECT files)	1x	0kB	0kB	0kB	0kB	11	11	11	11
properties (Java properties files)	4x	1kB	0kB	0kB	0kB	36	1	18	9
sample (SAMPLE files)	9x	14kB	0kB	4kB	1kB	497	8	169	55
ttf (TTF files)	1x	70kB	70kB	70kB	70kB	1391	1391	1391	1391
xml (XML configuration file)	92x	895kB	0kB	141kB	9kB	18784	1	984	204

Weighted Method Per Class

(Summe der Komplexität)

Rote/Hohe WMC Werte lassen auf einen höheren Wartungsaufwand schließen und die Klassen sind schwerer wiederverwendbar.

Method metrics	Class metrics		
	▼ class	OCavg	WMC
appsolutegamesgmbh.schnaps2gether.DataStructure.Bummerl2		1,10	11
appsolutegamesgmbh.schnaps2gether.DataStructure.Bummerl3		1,08	13
appsolutegamesgmbh.schnaps2gether.DataStructure.Bummerl4		1,08	14
appsolutegamesgmbh.schnaps2gether.DataStructure.Karte		1,68	32
appsolutegamesgmbh.schnaps2gether.DataStructure.KartenKomparator		8,00	8
appsolutegamesgmbh.schnaps2gether.DataStructure.Rufspiel		3,67	11
appsolutegamesgmbh.schnaps2gether.DataStructure.Spiel2		5,27	174
appsolutegamesgmbh.schnaps2gether.DataStructure.Spiel3		4,11	144
appsolutegamesgmbh.schnaps2gether.DataStructure.Spiel4		5,09	112
appsolutegamesgmbh.schnaps2gether.DataStructure.Spieler		1,00	15
appsolutegamesgmbh.schnaps2gether.DataStructure.Spielfeld4Logik		1,88	75
appsolutegamesgmbh.schnaps2gether.DataStructure.WrongGameException		1,00	1
appsolutegamesgmbh.schnaps2gether.GUI.ChangeNickname		1,00	4
appsolutegamesgmbh.schnaps2gether.GUI.GameEnd		1,25	5
appsolutegamesgmbh.schnaps2gether.GUI.Lobby		2,32	65
appsolutegamesgmbh.schnaps2gether.GUI.NeuesSpiel		1,14	8
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2		2,32	79
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2.Zugende		14,00	14
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2Client		2,29	117
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2Client.Zugende		20,00	20
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2Host		2,60	143
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld2Host.Zugende		4,00	4
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld3Client		4,10	197
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld3Host		4,03	246
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld4Client		3,19	115
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld4Client.Zugende		1,00	1
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld4Host		3,60	144
appsolutegamesgmbh.schnaps2gether.GUI.Spielfeld4Host.Zugende		3,00	3
appsolutegamesgmbh.schnaps2gether.GUI.Spielregeln		1,25	5
appsolutegamesgmbh.schnaps2gether.GUI.Startmenue		3,00	12
appsolutegamesgmbh.schnaps2gether.GUI.Statistik		1,25	5
Total			1797
Average		3,05	57,97

Metriken zur Klasse Spiel3

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
▶ Number of Parameters (avg/max per method)		0,886	0,887	3	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Number of Static Attributes (avg/max per type)	0	0	0	0	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Specialization Index (avg/max per type)		0	0	0	/Schnapsen/src/Schnaps2gether/Spiel3.java
Number of Classes	1				
▶ Number of Attributes (avg/max per type)	14	14	0	14	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Number of Static Methods (avg/max per type)	0	0	0	0	/Schnapsen/src/Schnaps2gether/Spiel3.java
Number of Interfaces	0				
Total Lines of Code	858				
▶ Weighted methods per Class (avg/max per type)	187	187	0	187	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Number of Methods (avg/max per type)	35	35	0	35	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Depth of Inheritance Tree (avg/max per type)		1	0	1	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ McCabe Cyclomatic Complexity (avg/max per method)		5,343	12,292	70	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Nested Block Depth (avg/max per method)		1,6	1,101	6	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Lack of Cohesion of Methods (avg/max per type)		0,864	0	0,864	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Method Lines of Code (avg/max per method)	754	21,543	55,92	325	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Number of Overridden Methods (avg/max per type)	0	0	0	0	/Schnapsen/src/Schnaps2gether/Spiel3.java
▶ Number of Children (avg/max per type)	0	0	0	0	/Schnapsen/src/Schnaps2gether/Spiel3.java

Die Klasse Spiel3.java enthält im Durchschnitt 0,886 Parameter für die 35 Methoden. Die Methoden „ZugAuswerten“ und „TalonAustauschen“ haben 3 Parameter. Es sind keine statischen Attribute enthalten, aber 14 mit private gekennzeichnete. SLOC beträgt 858. Je höher die Verschachtelungstiefe desto schwerer ist der Code zu verstehen (Verschachtelungstiefe >5 = erschwertes Programmverständnis).

Klasse Spiel3 – McCabe

Metric	Total	Mean	Std. Dev.	Maximum
▼ McCabe Cyclomatic Complexity (avg/max per method)		5,343	12,292	70
▼ Spiel3		5,343	12,292	70
istSpielzuEnde	70			
DarfKarteAuswaehlen	29			
ZugAuswerten	12			
hat20er	12			
DarfSpielAnsagen	8			
SpielAnsagen	8			
Auspielen	6			
kannFlecken	5			
Ansagen20er	4			
Spiel3	3			
Anfangsdeck	3			
Trumpfansagen	3			
KartenMischen	2			
getTalon	1			
getS1	1			
getS2	1			
getS3	1			
getTrumpf	1			
getSpiel	1			
setSpiel	1			
getSpieler	1			

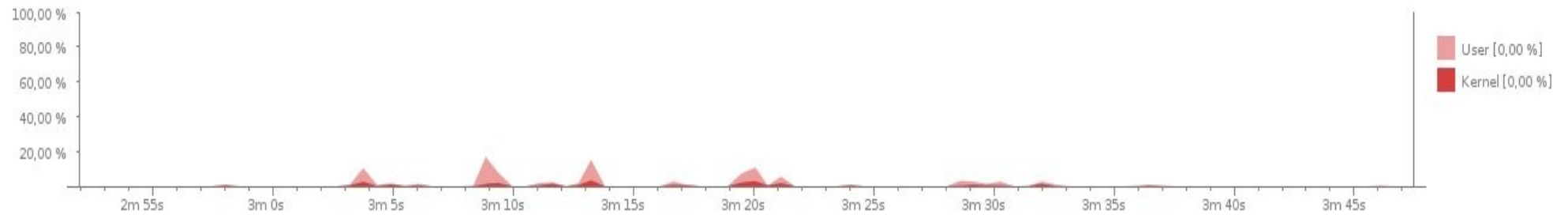
M McCabe-Zahl >10: schlechte Wartungseigenschaften

Klasse Spiel3 – Method Lines of Code

Metric	Total	Mean	Std. Dev.	Maximum
▼ Method Lines of Code (avg/max per method)	754	21,543	55,92	325
▼ Spiel3	754	21,543	55,92	325
istSpielzuEnde	325			
Anfangsdeck	75			
Trumpfansagen	66			
DarfKarteAuswaehlen	53			
SpielAnsagen	52			
ZugAuswerten	50			
Spiel3	29			
Auspielen	21			
Ansagen20er	16			
DarfSpielAnsagen	15			
hat20er	12			
KartenMischen	10			
kannFlecken	5			
TalonAustauschen	4			
getTalon	1			
getS1	1			
getS2	1			
getS3	1			
getTrumpf	1			
getSpiel	1			
setSpiel	1			

Leistung: CPU – Menu

(Testgerät: Samsung GT-I9300 – Android 4.3)



Peak1: „Name ändern“

Peak2: „Schnelles Spiel“

Peak3: „Spielregeln“

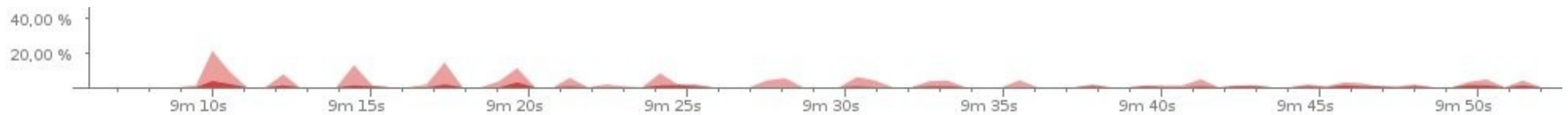
Peak4: „Spielen“

Peak5: „Statistik“

Peak5: „Beenden“

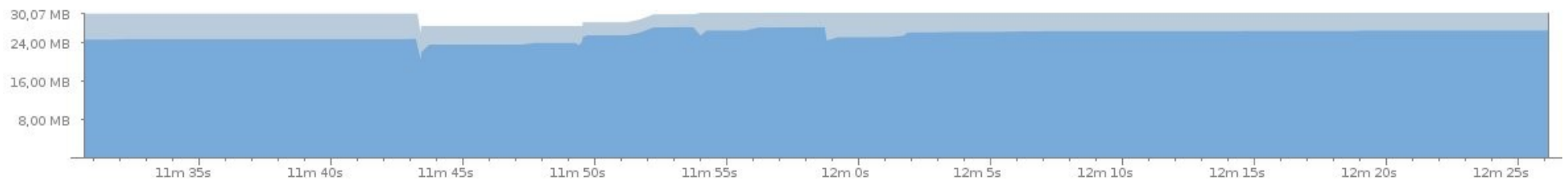
Leistung: Schnelles Spiel

1 Runde



Memory: Schnelles Spiel

1 Runde



Datenverbrauch (2 Spieler): ca. 23kb je Runde