

2015

Schnaps2gether - Produktmappe

MEHRSPIELER SCHNAPSEN ANDROID APP
EBERHARD, EDER, MAIER, PACHATZ, PLATTER

APPSOLUTE GAMES | EBERHARD, EDER, MAIER, PACHATZ, PLATTER

Inhalt

1.	Einleitung	3
1.1.	Zweck dieses Dokuments	3
1.2.	Projektrahmen	3
1.3.	Projektübersicht	3
2.	Allgemeine Produktbeschreibung	3
2.1.	Produktfunktionen	3
2.1.2.	Zusatzfeature 1 – Steuerung durch Bewegung	3
2.1.3.	Zusatzfeature 2 – Steuerung durch Sprache (ab Version 2.0)	4
2.1.4.	Zusatzfeature 3 – Statistik aller Spiele	4
2.1.5.	Information zu ähnlichen Systemen	4
2.1.6.	Zielgruppe	4
2.1.7.	Rahmenbedingungen	4
3.	Nicht funktionale Anforderungen	4
3.1.1.	Zuverlässigkeit	4
3.1.2.	Performanz	4
4.	Terminplan	5
4.1.1.	User Stories	5
5.	Vorläufiger Budgetplan	6
6.	Qualitätssicherung	6
6.1.1.	Verifikation der Anforderung	6
6.1.2.	Umfangsbestimmung von Quellcode	6
6.1.3.	Automatisierung	6
6.1.4.	Testfall	6
6.1.5.	Verwendete Testmethoden	6
6.1.6.	Metriken	7
6.1.11.	Weighted Methods per Class (Summe der Komplexität)	9
6.1.12.	Metriken zu Spiel3.cs	9
6.1.13.	McCabe Metrik zu Spiel3.cs	10
6.1.14.	Method Lines of Code Metrik zu Spiel3.cs	11
6.1.15.	Leistung CPU Metrik - Menu	11
6.1.16.	Leistung – Schnelles Spiel (1 Runde)	11
6.1.17.	Memory – Schnelles Spiel (1 Runde)	12
6.1.18.	Datenverbrauch	12
7.	Anhang	12

7.1.1.	Protokoll Kick-off Meeting	12
7.1.2.	Wichtigste Konzepte von Scrum	13
7.1.3.	Übersicht zur Android Programmierung	13
7.1.4.	Handhabung der Zeiterfassung.....	13
7.1.5.	Zeiterfassungseinträge von Appbsolute Games	14
7.1.6.	Testberichte	14
7.1.7.	Burndowncharts	14
7.1.8.	Sprintbacklogs und Product-Backlog.....	14

1. Einleitung

1.1. Zweck dieses Dokuments

Dieses Dokument dient dazu die Anforderungen und Pläne für das Projekt Schnaps2gether zu sammeln. Des Weiteren ist ein Kapitel über die Qualitätssicherung der App enthalten. Im Anhang finden Sie eine vollständige Stundenliste aller Team-Mitglieder.

1.2. Projektrahmen

1.2.1. Projektteam

Name	Matrikelnummer	Aufgaben
Eberhard Alice	1360554	Design
Eder Norbert	1360123	Testen / Qualitätssicherung / Metriken
Maier Kerstin	1360152	Implementierung
Pachatz Veronika	1360191	Implementierung
Platter Frederik	1360281	Dokumentation

1.3. Projektübersicht

Das Ziel dieses Projekts ist die Erstellung und Auslieferung eines Softwareprodukts. Dieses Produkt setzt sich aus der Android-Applikation Schnaps2gether, der Produktmappe und der zugehörigen Dokumentation (Wartungshandbuch, Benutzerhandbuch) zusammen.

2. Allgemeine Produktbeschreibung

2.1. Produktfunktionen

2.1.1. Basisfeatures

- Schnaps2gether ermöglicht 2, 3 oder 4 Spielern gegeneinander an einem Ort Schnapsen zu spielen.
- Ein Spieler kann ein neues Spiel starten oder einem bereits erstellten Spiel beitreten.
- Der Spieler kann ein Spiel für 2, 3 oder 4 Spieler anlegen.
- Jeder Spieler kann sich einen Nicknamen geben.
- Jedem Spieler werden die aktuellen Punktelisten des Spiels angezeigt.
- Spieler können im 2er Spiel Schummeln und beim Schummeln ertappt werden (durch Betätigung von einer Schaltfläche).
- Möglichkeiten zum Schummeln sind Karten austauschen und in die Karten von Gegnern schauen.
- Wenn ein Spieler das Spiel verlässt wird die Spielrunde automatisch beendet.
- Die Spielregeln sind als Dokumentation in der App aufrufbar.

2.1.2. Zusatzfeature 1 – Steuerung durch Bewegung

- Als Spieler kann ich durch eine Schüttelbewegung des Android-Geräts das Schummeln eines gegnerischen Spielers unterbinden.

2.1.3. Zusatzfeature 2 – Steuerung durch Sprache (ab Version 2.0)

- Spieler können 20er und 40er ansagen, wenn Sie mit dem nächsten Spielzug an der Reihe sind.
- Ein Spieler kann beim 3er und 4er Schnapsen ansagen, welche Spielart (Land, Schnapser, etc.) er spielen möchte und das Spiel eines anderen „fleckten“.

2.1.4. Zusatzfeature 3 – Statistik aller Spiele

- Jeder Spieler kann sich seine persönliche Statistik über all seine bisherigen Spiele anzeigen lassen. Die Statistik enthält:
 - die Anzahl der gewonnen und verlorenen Spiele
 - die Gesamtanzahl der erreichten Bummerl
 - die höchste erreichte Punkteanzahl in einem Spiel

2.1.5. Information zu ähnlichen Systemen

Kartenspielapplikationen sind keine Besonderheit, allerdings verwenden sie im Allgemeinen international bekannte Kartenspiele. Mehrspielerfähige Applikationen gibt es auch in großer Menge, jedoch verwenden diese fast immer eine Internetverbindung, sodass Spieler aus allen Winkeln der Welt miteinander spielen können. Mehrspieler-Kartenspielapplikationen sind meist simple Implementierungen ihrer realen Vorbilder und brillieren durch ein intuitives User Interface.

2.1.6. Zielgruppe

Als Zielgruppe werden Jugendliche im Alter von 14 bis 25 Jahren betrachtet, die in Cafés und Gasthäuser gehen und sich durch die Applikation Schnaps2gether zusammensetzen können, um Karten zu spielen.

2.1.7. Rahmenbedingungen

Die Obergrenze des Budgets des Kunden liegt bei 100.000 €. Acht Konkurrenten bewarben sich um die Ausschreibung und der Kunde legt bei der Wahl des Angebots vor allem Wert auf das Preis-Leistungs-Verhältnis. Das vollständige Produkt soll dem Kunden am 24.06.2015 präsentiert werden.

3. Nicht funktionale Anforderungen

3.1.1. Zuverlässigkeit

Die App soll zuverlässig genug sein um den Spielspaß nicht zu beeinträchtigen.

3.1.2. Performanz

Die App soll flüssig genug laufen um den Spielspaß nicht zu beeinträchtigen.

4. Terminplan

4.1.1. User Stories

Order	ID	Title	State	Effort	Description	Tags
1	27	Als Spieler kann ich zwischen drei Spielmodi entscheiden. (2er-,3er-, oder...	New	1	3	*
2	28	Als Spieler kann ich, mittels Sprachsteuerung, 20er und 40er ansagen	New	5	2	*
3	29	Als Spieler kann ich mir einen Nicknamen geben.	Approved	2	5	*
4	30	Als Spieler kann ich schummeln. (Karten austauschen, in Karten schauen,...	New	13	4	*
5	31	Als Spieler kann ich andere beim schummeln ertappen und abwenden	New	5	4	*
6	47	Als Spieler habe ich ein Kartendeck, dass Schnapsen ermöglicht	New	5	5	*
7	32	Als Spieler kann ich, mittels einer Geste, das Schummeln anderer Spieler ...	New	5	2	*
8	33	Als Spieler kann ich wählen, welches Spiel ich spielen will (Bauernschna...	New	5	3	3 4
9	34	Als Spieler kann ich, mittels Sprachsteuerung, ein Spiel ansagen.	New	5	1	3 4
10	35	Als Spieler kann ich, mittels Sprachsteuerung oder mittel Knopfdruck, das ...	New	5	3	3 4
11	36	Als Spieler kann ich eine Kurzbeschreibung der möglichen Spiele einsehen.	New	3	3	3 4
12	37	Als Spieler sehe ich immer den aktuellen Spielstand und die Anzahl der b...	New	5	2	*
13	38	Als Spieler kann ich meine Statistik anschauen (gewonnenen Spiele, Anz...	New	5	4	*
14	49	Als Spieler kann ich den Modi 4er Schnapsen spielen	New	13	4	4
15	48	Als Spieler kann ich den Modi 3er Schnapsen spielen	New	20	4	3
16	39	Als Spieler kann ich den Modi 2er Schnapsen spielen	New	20	5	2
17	40	Design des Startmenüs	Approved	5	5	*
18	54	Design Spielfeld (Elemente)	New	3	5	*
19	41	Design Spielfeld (Grundfeld)	Approved	5	5	3
20	42	Als Spieler seh ich meine Hand in sortierter Reihenfolge	New	3	2	*
21	43	Zu Beginn des Spieles, kann ich entweder ein neues Spiel beginnen, ode...	New	13	5	*
22	44	Als Spieler möchte ich eine Mindestanzahl von zwei Spielern	New		Constraint	
23	50	Als Spieler muss ich mich mit den Mitspielern in einem Raum befinden	New		Constraint	
24	51	Das ganze Spiel muss zuverlässig sein	New		Constraint	

ID	Title	State	Assigned To	Remaining Work
29	Als Spieler kann ich mir einen Nicknamen geben.	Approved		
52	Eingabefeld	To Do	Kerstin Maier	2.11
53	Überprüfung, ob Nickname bereits vergeben ist	To Do	Veronika Pachatz	2.11
40	Design des Startmenüs	Approved		
55	Menüpunkte	To Do	Alice Eberhard	6.33
56	Hintergrund	To Do	Alice Eberhard	4.22
41	Design Spielfeld (Grundfeld)	Approved		
57	Hintergrund	To Do	Alice Eberhard	10.55

5. Vorläufiger Budgetplan

Pos.	Beschreibung	EUR
1	Basisfeatures	72 345,00 €
2	Zusatzfeature 1 – Steuerung durch Bewegung	2 205,00 €
3	Zusatzfeature 2 – Steuerung durch Sprache	15 750,00 €
4	Zusatzfeature 3 – Statistik aller Spiele	4 410,00 €
	SUMME	94 710,00 €

6. Qualitätssicherung

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

6.1.1. Verifikation der Anforderung

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

6.1.2. Umfangsbestimmung von Quellcode

- Lines of Code
- Halstead
- McCabe

6.1.3. Automatisierung

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

6.1.4. Testfall

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

6.1.5. Verwendete Testmethoden

6.1.5.1. 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.

6.1.5.2. White-Box-Test

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

6.1.5.3. Reviews

Reviews wurden wöchentlich durchgeführt. Dadurch wurde die Qualität erhöht und Fehler frühzeitig erkannt.

6.1.6. Metriken

6.1.6.1. 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%

6.1.7. Projekt – Lines of Code and Command lines

- Lines of Code: 13.179 Anzahl alle Programmzeilen (Projektumfang)
- Source Code Lines: 10.650 Algorithmen-Implementierungsgröße
- Comment Lines: 544 Kommentaranteil des Projektes

6.1.8. 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

6.1.9. 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

6.1.10. Größe und Anzahl der Dateien

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	7kB	497	8	169	55
tif (TIF files)	1x	70kB	70kB	70kB	70kB	1391	1391	1391	1391
xml (XML configuration file)	92x	895kB	0kB	141kB	9kB	18784	1	984	204

6.1.11. Weighted Methods 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.Startmenuue		3,00	12
appsolutegamesgmbh.schnaps2gether.GUI.Statistik		1,25	5
Total			1797
Average		3,05	57,97

6.1.12. Metriken zu Spiel3.cs

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).

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

6.1.13. McCabe Metrik zu Spiel3.cs

M McCabe-Zahl >10: schlechte Wartungseigenschaften

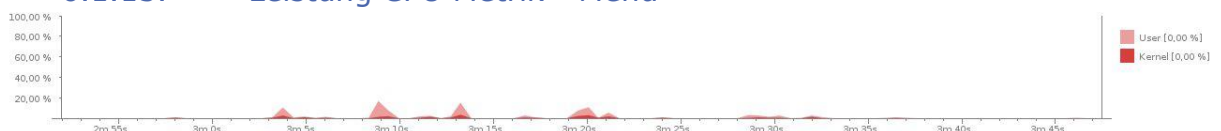
Metric	Total	Mean	Std. Dev.	Maximum
▼ M McCabe Cyclomatic Complexity (avg/max per method)		5,343	12,292	70
▼ Spiel3		5,343	12,292	70
istSpielzuEnde	70			
DarfKarteAuswaehlen	29			
ZugAuswerten	12			
hat20er	12			
DarfSpielAnsagen	8			
SpielAnsagen	8			
Auspielen	6			
kannFlecken	5			
Ansagen20er	4		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

6.1.14. Method Lines of Code Metrik zu Spiel3.cs

McCabe-Zahl >10: schlechte Wartungseigenschaften

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			

6.1.15. Leistung CPU Metrik - Menu

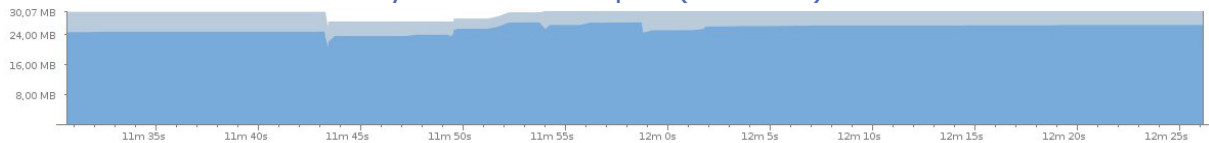


- Peak1: „Name ändern“
- Peak2: „Schnelles Spiel“
- Peak3: „Spielregeln“
- Peak4: „Spielen“
- Peak5: „Statistik“
- Peak6: „Beenden“

6.1.16. Leistung – Schnelles Spiel (1 Runde)



6.1.17. Memory – Schnelles Spiel (1 Runde)



6.1.18. Datenverbrauch

- Der Datenverbrauch pro Runde beträgt für 2 Spieler ca. 23 KB.

7. Anhang

7.1.1. Protokoll Kick-off Meeting

- Besprochene Punkte:
 - Aus welchem Grund wollen sie diese App erstellen?
 - Wie sieht die Zielgruppe aus?
 - Welches Spiel sollte genau implementiert werden?
 - Welche Spielregeln sollten dem Spiel zugrunde liegen?
 - Wie hoch sollte die maximale Spieleranzahl? Bei Schnapsen z.B. maximal 4 Spieler
 - In der Kurzbeschreibung wurde erwähnt, dass alle Spieler vor Ort sein sollen. Kann Implementierung via Bluetooth erfolgen.
 - Wie sollte es möglich sein zu Schummeln, Schummeln zu unterbinden und in die Karten eines anderen zu schauen?
 - Wie sollten Benutzer Gesten, Sprachsteuerungen, Bewegungen zur Spielbeeinflussung nutzen können?
 - Genügt die Unterstützung ab Android Version 4.0
 - Welche Produktfunktionen sind Musskriterien?
 - Sind Punktlisten ein Muss-/Wunschkriterium?
 - Gibt es einen Zeitrahmen / Budgetrahmen?
 - Sind Wartungstätigkeiten zu erwarten? Wenn ja, werden dafür nach Projektfertigstellung eigene Wartungsverträge vereinbart oder soll dies gleich in die Aufwandsschätzung mit einfließen?
 - Gibt es für Sie noch wesentliche Punkte die Musskriterien sind und in der bisherigen Diskussion nicht besprochen wurden?

7.1.2. Wichtigste Konzepte von Scrum

- **Daily Scrum Meeting:** Tägliche maximal 15 minütige Meetings des Entwicklungsteams und des Scrum Masters. Besprochen werden die zuletzt erledigten und die als nächstes anfallenden User Stories.
- **Product Owner:** Er repräsentiert den Kunden und vertritt die Interessen des Kunden.
- **Product-Backlog:** Geordnete Auflistung aller User Stories eines Softwareproduktes.
- **Scrum Master:** Person die das Scrum-Team vom Management abschirmt, aber nicht Teil des Scrum-Teams oder des Managements ist. Sorgt für einen fortlaufenden Entwicklungsprozess.
- **Scrum Team:** Entwicklungsteam ohne fixe Zuteilung von Aufgaben über alle Sprints hinweg.
- **Scrum:** agiles Softwareentwicklungs-Framework.
- **Sprint Review Meeting:** Scrum-Team präsentiert Ergebnisse des letzten Sprints.
- **Sprint:** kurze Entwicklungszyklen in der Länge von 2 bis 6 Wochen.
- **Sprint-Backlog:** Geordnete Auflistung aller User Stories die innerhalb eines Sprints abgearbeitet werden.
- **Story Point:** Einheit zur Größenschätzung einer User Story.
- **User Story:** Beschreibung eines konkreten Anwendungsfalls aus der Sicht des Nutzers.

7.1.3. Übersicht zur Android Programmierung

Die Entwicklung dieses Software-Produkts wird mit der Hilfe von Android Studio erfolgen. Dies ist ein IDE für das Java-Android-Framework und wurde von Google entwickelt. Grundsätzlich gibt es einen Source-Ordner mit ein oder mehreren *.xml-Dateien und darin befinden sich Informationen über die GUI-Elemente. Außerdem kommen auch verwendete Bilder in diesen Ordner. Des Weiteren gibt es eine Main-Datei, von der aus die App gestartet wird. Die wichtigsten Funktionen der App befinden sich in dieser Main-Datei. Beim Erstellen eines neuen Projektes kann die Android-Version ausgewählt werden und es wird automatisch angezeigt zu viel Prozent die App mit verschiedenen Android-Geräten kompatibel ist. Man kann innerhalb des Android Studios einen Emulator starten um die App zu testen. Leider kann mit dem Emulator keine Mehrspieler-App die via WLAN funktioniert getestet werden. Zu Beginn des Projektes hatte keines der Teammitglieder praktische Erfahrung mit der Android-App Programmierung.

Die Programmierung benötigt keine speziellen Kenntnisse. Grundlegende Java-Kenntnisse reichen aus um die Implementierung vorzunehmen. Beim Design ist darauf zu achten, dass alle GUI Elemente und Activities für unterschiedliche Bildschirmgrößen (Tablet, Smartphones) angepasst werden.

Die Mehrspielerverbindung wurde mit Hilfe von der Google Nearby Connection implementiert.

7.1.4. Handhabung der Zeiterfassung

Für die Zeiterfassung wurde die kostenlose Online-Plattform toggl verwendet. Die Beschreibung des Zeiterfassungseintrags hat folgende Syntax: „Tätigkeit – User Story Titel“. Tätigkeit kann Analyse, Implementierung, QA, Dokumentation oder

Meeting sein. Die Zeiterfassungseinträge werden maximal auf 15 Minuten genau verbucht.

7.1.5. Zeiterfassungseinträge von Appsolute Games

- Zeiterfassungen gesammelt: [Zeiterfassung\Zeiterfassung_Gesamtbericht.pdf](#)
- Zeiterfassung Eberhard: [Zeiterfassung\Zeiterfassung_Eberhard.pdf](#)
- Zeiterfassung Eder: [Zeiterfassung\Zeiterfassung_Eder.pdf](#)
- Zeiterfassung Maier: [Zeiterfassung\Zeiterfassung_Maier.pdf](#)
- Zeiterfassung Pachatz: [Zeiterfassung\Zeiterfassung_Pachatz.pdf](#)
- Zeiterfassung Platter: [Zeiterfassung\Zeiterfassung_Platter.pdf](#)

7.1.6. Testberichte

- Sammlung der Testberichte: [Testberichte](#)

7.1.7. Burndowncharts

- Sammlung der Burndowncharts: [Burndowncharts](#)

7.1.8. Sprintbacklogs und Product-Backlog

- Sammlung der Backlogs: [Backlogs](#)