Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Tržaška 25, Ljubljana, Slovenija

Seminarska naloga

ZANESLJIVOSTNA ANALIZA HTC T-MOBILE G1

pri predmetu

RAČUNALNIŠKA ZANESLJIVOST IN DIAGONSTIKA

AVTORJI :

Anže Leban

Tom Vodopivec

Davor Sluga

Matjaž Okretič

Miha Gorše

Luka Finžgar

Miha Nagelj

Ivo Križman

Blaž Primc

Ljubljana, dne 23.5.2009

KAZALO

Uvod

Pred vami je seminarska naloga, ki obravnava zanesljivostno analizo naprave HTC G1 z naloženim sistemom Android. Če na hitro preletimo kazalo, lahko vidimo, da je seminarska naloga sestavljena in treh sklopov zanesljivostne analize. Najprej je obravnavana statična analiza zanesljivosti programske opreme. Tu so podani rezultati analize izvorne kode. Obravnavanih je več različnih metrik kot so Halsteadovo število predvidenih hroščev, ciklomatična kompleksnost, indeks vzdrževanja, … Drugi sklop predstavlja FTA *(Fault Tree Analysis)* analiza naprave HTC G1. Tu so pomembni FTD-ji *(Fault Tree Diagram),* ki prikazujejo stanje sistema v odvisnosti stanj njegovih komponent. FTA analiza zajema vrsto mogočih dogotkov kot so izpad oskrbe z električno energijo, okvara LCD zaslona, … Zadnji sklop te seminarske naloge se nanaša na FMEA analizo *(Failure Mode and Effects Analysis).* To je skupek metodologij, ki so namenjene predvidevanju možnih načinov okvar proizvoda ali procesa. V našem primeru je analiza seveda namenjena napravi HTC G1. Po FMEA analizi imamo na voljo ocene tveganj okvar naprave, potem je tu še pregled akcij za zmanjševanje tveganj in vnovične ocene tveganja za sprejete akcije.

Pri nekaterih poglavjih so podani tudi opisi programske opreme, ki je bila uporabljena pri analizi zmogljivosti. S tem imamo na voljo pregled orodij, ki so primerna za posamezne analize in s tem priporočljiva v primeru, če se tudi bralec loti podobne analize.

Pred poglavji o sami analizi je tu še kratek pregled samega sistema na katerem se je analiza izvajala. Predstavljene so tako specifikacije naprave HTC G1, kot tudi podatki o programski opremi, ki je naložena na dotično napravo.

1. HTC G1 in Android

1.1 HTC G1

Slika naprave:

Slika 1.1: HTC G1

Specifikacije:

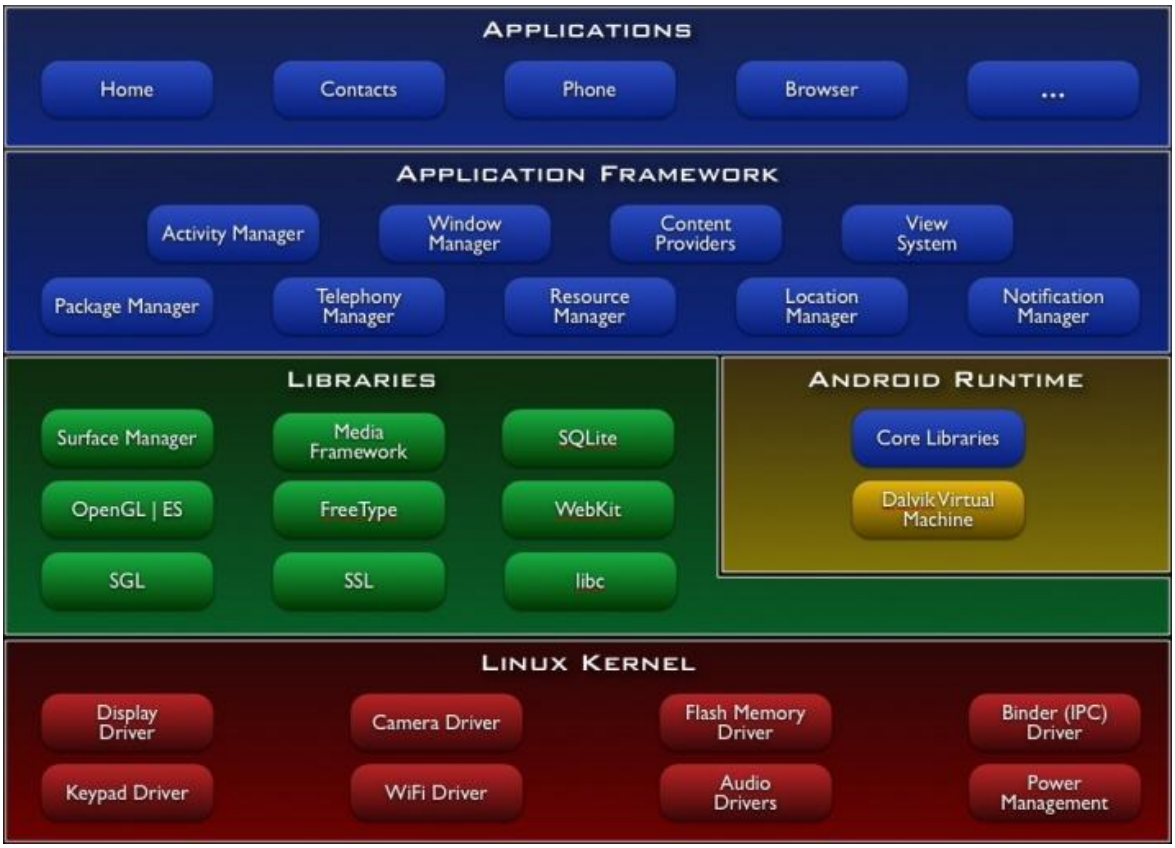
Procesor	Qualcomm® MSM7201A™, 528 MHz
Operacijski sistem	Android™
Spomin	ROM: 256, RAM:192
Dimenzije (VxDxŠ)	117.7 mm x 55.7 mm x 17.1 mm
Teža	158 gramov z baterijo
Zaslon	3.2-inčni TFT-LCD, plosk, občutljiv na dotik, resolucija: 320 x 480 (HVGA)
Omrežja	HSPA/WCDMA: 2100 MHz, Quad-band GSM/GPRS/EDGE: 850/900/1800/1900 MHz
Tipkovnica	Zdrsljiva 5-vrstična QWERTY tipkovnica
GPS	GPS navigacija kompatibilna z Google Maps™
Povezljivost	Bluetooth® 2.0, Wi-Fi®: IEEE 802.11b/g, HTC ExtUSB™
Kamera	3.2 megapikslov, barvna kamera z avtofokusom
Avdio	Vgrajen mikrofon in zvočnik, podprti formati: AAC/+, AMR-NB, SP/MIDI, MP3, WMA, WMV
Baterija	Litij-ionska, kapaciteta: 1150 mAh
Razširitvena reža	microSD™ spominska kartica (SD 2.0 kompatibilna)
AC Adapter	Obseg/frekvenca napetosti: 100 ~ 240V AC, 50/60 Hz DC izvor: 5V, 1A
Posebnosti	Digitalni kompas, senzor gibanja

Tabela 1.1: HTC G1 specifikacije

1.2 Google Android

Android je programska platforma in operacijski sistem za pametne mobilne telefone, ki temelji na Linux jedru. Razvija ga Google v sodelovanju s podjetji združenja Open Handset Alliance (OHA)[10]. Android je v prvi vrsti namenjen mobilnim telefonom, v prihodnosti pa naj bi se z njim opremilo tudi nekatere druge naprave (prenosni računalniki in podobne naprave). Večina kode je izdane kot odprtokodne pod Apache 2.0 licenco.

Glavne komponente operacijskega sistema so prikazane na spodnji sliki:



Slika 1.2 : Glavne komponente OS Android

Za razvoj programov platforme Android nam je na voljo Android SDK, ki vsebuje potrebna orodja za emulacijo, razhroščevanje ter programske vmesnike(API). Razvoj poteka v programskem jeziku Java. Datoteke vrste *class* se pretvori v format *dex* (Dalvik izvršljiva koda) z orodjem dx. Vsaka Android aplikacija nato teče v svojem Dalvik virtualnem stroju, ki je bil razvit z namenom uporabe na računalnikih, ki so pomnilniško in/ali procesno omejeni.

Android temelji na Linux jedru verzije 2.6, ki ga uporablja za varnost, upravljanje s pomnilnikom, razvrščevanje, mrežni sklad ter gonilnike. Za Android so na voljo tudi C/C++ knjižnice, ki jih potrebujejo razni deli sistema. Razvijalec lahko do njih dostopa preko programskega vmesnika.

2. Statična analiza programske opreme

Naša naloga je bila preveriti zanesljivost programske opreme Google Android, namenjene mobilnim napravam. Zaradi mladosti Google Android platforme bo zelo pomembno, kako se bo pri tako velikem projektu pazilo na to, da ga bo lahko vzdrževati ter da bo čim manj napak. Naš test zajema izvorno kodo Androidovega ogrodja in izvorno kodo nekaterih pogostih aplikacij. Za tesiranje Linux jedra se nismo odločili, saj je za slednje bilo narejenih že veliko testiranj in analiz.

2.1 Metode

Pri našem delu smo uporabili program za statično testiranje **CMTJava** podjetja TestWell, poleg tega pa smo imeli zaradi odprtokodne narave Androida tudi vpogled v izvorno kodo, t.i. white box testing. Datotek z izvorno kodo v jeziku Java je pri Androidu preko 10 000, zato pregledovanje vsake datoteke ni prišlo v poštev. Morali smo poiskali programe, ki to avtomatizirajo in delujejo po metodah, ki smo jih na predavanjih tudi spoznali. Program CMTJava analizira datoteke z izvorno kodo in pregleda njihovo kompleksnost ter predvidi, kako težko bo projekt vzdrževati. Izhodne podatke s poročilom lahko dobimo v obliki txt, html, xls ali xml datoteke. Program uporablja dve zelo znani meri znotraj deterministčnega modela testiranja programske zanesljivosti. To sta McCabejeva ter Halsteadova mera.

McCabejeva nam nudi izračun zgornje meje preostalih napak v programu, Halsteadova pa oceno števila napak v programu. Metodi bomo natančneje opisali v nadaljevanju.

CMTJava nam vrne naslednje metrike:

a) Število vrstic izvorne kode (LOC)

Pri tem nam izpiše:

- število vseh vrstic (LOCphy)
- število vrstic s programsko kodo(LOCpro)
- število vrstic s komentarji(LOCcom), število praznih vrstic(LOCbl)
- število podpičij
- število bločnih komentarjev

b) Ciklomatično kompleksnost (McCabejevo mero)

McCabejevo ciklomatično število ***v(G)*** nam pokaže, kako kompleksne so vejitve v programu. Podaja nam število neodvisnih poti v programu, poleg tega pa nam poda tudi zgornjo mejo števila potrebnih testov programa, ki jih moramo izvesti, če hočemo, da se vsak programski stavek izvede vsaj enkrat. Večje kot je to število,več možnih poti je skozi program in težje ga je razumeti. ***v(G)*** je neodvisno od kompleksnosti podatkovnih struktur,vmesnikov...

v(G) metode naj bi bil manj kot 10, saj je po izkušnjah več kot 10 poti težko dobro stestirati.

Povsem sekvenčen program ima ***v(G)=1***, za 1 pa ga povečajo:

- ***if*** stavek
- ***for*** in ***while*** zanke
- ***case*** deli stavka ***switch*** (razen ***default*** dela)
- ***catch()*** del pri lovljenju napak (razen stavka ***finally***)
- Pogojni izrazi tipa ***expr1 ? expr2 : expr3***
- Sestavljeni pogoji z znaki ***&&***, || in drugi

Primer za McCabejevo mero s stavkom switch:

```
switch (...) {
     ...
     case 5: foo1(); break;   //v(G) se poveča za 1
     case 8: foo2(); break;   //v(G) se poveča za 1
     default: foo3();         //v(G) se ne poveča
}
```

c) Halsteadovo mero

Tu se prešteje število različnih operatorjev ***n1***, število različnih operandov ***n2***, število vseh operatorjev ***N1*** in število vseh operandov ***N2***.

Pod operande štejemo:

- preddefinirane ključne besede, ki definirajo podatkovne tipe (***boolean***, ***byte***, ***char***, ***double***, ***float***, ***int***, ***long***, ***short***)
- ključne besede ***false***, ***null***, ***super***, ***this***, ***void***
- vse oznake, ki niso ključne besede
- numerični literali in literali iz nizov

Vse ostale "žetone" v izvorni kodi štejemo med operatorje, pri čemer se znak '***:***' v stavku ***case:*** obravnava kot del stavka ***case***, zato se štejeta skupaj, ***(...)*** v stavkih ***for(...)***, ***if(...)***, ***while(...)*** itd. se štejejo skupaj. Pari oklepajev ***[....]***, ***{...}***, ***(...)*** se obravnavajo kot en operator.

Vse ostale Halsteadove mere program izračuna iz ***n1***, ***n2***, ***N1***, ***N2*** po naslednjih formulah:

- ***Dolžina programa (N)*** je vsota vseh operatorjev in operandov; ***N=N1+N2***
- ***Velikost slovarja (V)*** je vsota razlčnih operatorjev in različnih operandov; ***n=n1+n2***
- ***Volumen programa*** pove, koliko informacije (v bitih) vsebuje program; ***V=N*log2(n)***
- ***Nivo težavnosti(D)*** je proporcionalen razmerju med številom operandov in številom različnih operandov(če se isti operand večkrat uporabi v programu je s tem več možnosti za napake); ***D=(n1/2)*(N2/n2)***
- ***Nivo programa(L)*** je inverz nivoja težavnosti, program z nizkim nivojem je bolj občutljiv na napake kot program z višjim; ***L=1/D***

- ***Težavnost implementacije(E)*** je proporcionalna volumnu ter nivoju težavnosti; ***E=V*D***
- ***Čas za implementacijo ali čas za razumevanje programa(T)*** je proporcionalen težavnosti implementacije. Halstead je empirično ugotovil, da deljenje ***E*** z 18 daje zadostno aproksimacijo za čas v sekundah; ***T=E/18***
- ***Predvideno število programskih hroščev(B)*** je povezan s kompleksnostjo programa; ***B=(E^(2/3))/3000***, ta vrednost se potem pomnoži s korelacijskim faktorjev, ki je določen v konfiguracijski datoteki.

d) <u>Indeks možnosti vzdrževanja (MI = Maintainability Index)</u> je število, ki nam pomaga oceniti možnost vzdrževanja naše izvorne kode, poleg tega pa nam pokaže tudi kakšno je tveganje, da se bo program preveč razrasel in tako postal težko obvladljiv. ***MI*** se ponavadi spremlja z namenom, da se prepreči ali zmanjša entropijo programske opreme in s tem poveča integriteto. Vsaka programska oprema, ki se ji s časom dodajajo nove funkcionalnosti, postaja vedno bolj kompleksna in vedno slabše organizirana, pri tem pa se zgublja tudi struktura, ki je bila zamišljena ob načrtovanju. ***MI*** nam lahko pove tudi, kdaj bi bilo ceneje ali manj tvegano kodo napisati na novo, namesto da staro ves čas spreminjamo.

V našem programu se izračunata dve števili MI: eno vsebuje kometarje (***MI***), druga pa komentarjev ne upošteva (***Miwoc***).
Pomen vrednosti ***MI*** (s komentarji):

- >=85 lahko vzdrževanje
- 65-85 srednja težavnost vzdrževanja
- < 65 težavno vzdrževanje

Pri velikih izvornih datotekah, ki so nekometirane in slabo strukturirane, je lahko vrednost ***MI*** celo negativna.

Izračun vrednosti MI

MI se izračuna za vsako funkcijo/razred/strukturo, za vsako datoteko ter za vse datoteke skupaj.
Dejansko se izračunajo tri vrednosti:

1. ***Miwoc*** – to je ***MI*** brez komentarjev
2. ***Micw*** – utež za kometarje pri ***MI***
3. ***MI*** – indeks možnosti vzdrževanja **= *MIwoc* + *MIcw***

Računa se po naslednjih formulah:
MIwoc = 171 - 5.2 * ln(aveV) -0.23 * aveG -16.2 * ln(aveLOC)
MIcw = 50 * sin(sqrt(2.4 * perCM))
MI = MIwoc + MIcw

Kjer so:
aveV = velikost slovarja(V) na razred
aveG = povprečna ciklomatična kompleksnost v(G) na razred
aveLOC = povprečno število vrstic (LOCphy) na razred
perCM = povprečen procent vrstic s komentarj na razred

2.2 Rezultati

V nadaljevanju bodo predstavljeni rezultati statične analize izvorne kode sistema Android. Kot smo že omenili, v testiranje nismo vključili raznoraznih dodatnih programskih paketov, ki so na voljo kot del izvorne kode Androida. Za to potezo smo se odločili, ker je že obseg osnovne izvorne kode 1086 java datotek. Če bi upoštevali vse dodatne Androidove pakete, pa bi število naraslo krepko preko 10000 datotek izvorne kode.

Najprej bomo predstavili rezultate, ki se nanašajo na vse datoteke izvorne kode, ki pripadajo sistemu Android. To je nekakšen povzetek in analiza rezultatov, ki so bili pridobljeni s pregledovanjem celotne izvorne kode. Nato bomo predstavili še nekaj primerov analize izvorne kode na dejanskih primerih datotek izvorne kode (izmed 1086 datotek smo izbrali nekaj takih, ki so primerne za razlago rezultatov). Celotna analiza bo na voljo tudi kot priloga.

2.2.1 Rezultati celotne izvorne kode sistema Android

Naslednji rezultati nam povedo povprečne vrednosti metrik, ki jih lahko izmerimo s pomočjo programa CMTJava, prav tako pa dobimo podatke različnih lastnostih same izvorne kode (lastnosti posameznih javanskih datotek, metod, razredov, vmesnikov in podobno).

Za večjo preglednost najprej definirajmo nekaj okrajšav in razlag:

- LOCall: Število vrstic kode (Lines Of Code)
- LOCpro: Število vrstic programske kode
- LOCcom: Število vrstic komentarjev
- LOCbl: Število praznih vrstic
- Extends: Določa razred ali vmesnik, ki je razširljiv
- Implements: Določa implementirane vmesnike
- Class: Razred
- Interface: Vmesnik

Iz tabele 1 lahko razberemo podatke o številu datotek z izvorno kodo, številu razredov, metod in vmesnikov. Prav tako je za vsako datoteko podano LOCall, LOCpro, LOCcom in LOCbl. Vidimo tudi povprečno število podpičij v datoteki. Še en zanimiv podatek je število blokov javanskih komentarjev. To je število komentarjev, ki so združeni z /* … */ ali /** … **/.

Podobno lahko tudi za metode vidimo število le teh. Zraven pa imamo še dodaten podatek o povprečni ciklomatični kompleksnosti in povprečna števila LOC. Metode, ki so vključene v rezultate, so vzete le iz prvih dveh najvišjih nivojev izvorne kode.

Za razrede in vmesnike velja podobna razdelitev. Za oboje lahko vidimo celotno njihovo število v izvorni kodi in pa njihovo število le na najvišjem nivoju. Rezultati nam povedo tudi število posameznih razredov ali vmesnikov, ki so razširljivi. Za razrede imamo še podatek o številu razredov z implementacijo drugih razredov ali vmesnikov.

Podan imamo še podatek o direktorijih, v katerih je izvorna koda, in sicer skupno število direktorijev. Na koncu imamo še informacije o celotnem sistemu. Podan imamo podatek o ciklomatični kompleksnosti in indekse možnosti vzdrževanja.

Datoteke	
1086	Skupaj datotek
358186	LOCall
191563	LOCpro
125678	LOCcom
42239	LOCbl
98329	Število podpičij - ";"
329	Povprečno LOCall
35%	Povprečen odstotek LOCcom
14794	Število javanskih blokov komentarjev
Metode	
17400	Skupaj metod
8	Povprečno LOCpro
30	Povprečen odstotek LOCcom
2	Povprečen v(G)
Razredi	
1711	Skupno razredov
987	Število razredov na najvišjem nivoju
918	Extends (razširljiv)
472	Implements (implementira)
Vmesniki	
334	Skupno vmesnikov
122	Število vmesnikov na najvišjem nivoju
59	Extends (razširljiv)
Direktoriji	
69	Skupaj direktorijev
Celotni Sistem	
25101	Ciklomatična kompleksnost celotne izvorne kode
100	Indeks možnosti vzdrževanja - samo programska koda (brez komentarjev)
38	Indeks možnosti vzdrževanja - samo komentarji
138	Indeks možnosti vzdrževanja – koda + komentarji

Tabela 2.1: Lastnosti izvorne kode

V naslednji tabeli so prikazani rezultati osnovnih metrik. Tabela 2 prav tako vsebuje rezultate za celotno izvorno kodo. Če primerjamo originalne rezultate s temi iz tabele 2, je opaziti, da se število merjenih objektov ne ujema s seštevkom razredov, metod in vmesnikov iz tabele 1. To je posledica nekaterih napak pri izvajanju programa CMDJava. Vzrokov za napake na žalost nismo uspeli odpraviti, ampak je razlika v rezultatih kljub napakam skoraj neopazna.

Tabela 2 prikazuje število alarmov za posamezno metriko. Alarm se sproži, če je vrednost metrike izven vnaprej določenih mejnih vrednosti. Mejne vrednosti lahko spreminjamo tudi sami. V našem primeru se slednjega nismo posluževali, saj smo ocenili, da so privzete mejne vrednosti v skladu z optimalnimi mejnimi vrednostmi.

Preprost primer, ko se sproži alarm (ali ne sproži), je primer komentarjev v programski kodi. Obstajata dva robna primera. Prvi je, ko v izvorni kodi sploh ni komentarjev in drugi ko je komentarjev veliko več kot same kode. Obeh primerov se najraje izogibamo. Tako se odločimo za maksimalni in minimalni odstotek komentarjev v programski kodi. Če je odstotek komentarjev v testirani izvorni kodi izven določenih meja, se sproži alarm, drugače pa ne.

Metrika	**Št. Merjenih objektov**	**Število alarmov**	**Delež alarmov**	**Meje za razrede**	**Meje za metode**
Ciklomatična kompleksnost	19100	481	0.02	1 – 100	1 – 10
LOCpro	19100	682	0.03	4 – 400	1 – 40
Odstotek komentarjev	19100	3262	0.17	20 – 60	20 – 60
Halsteadov volumen	19100	1352	0.07	100 – 8000	4 – 1000
Halsteadovo število predvidenih hroščev	19100	275	0.01	0 – 2	Ni mej
Indeks možnosti vzdrževanja	19100	333	0.02	80 -	80 -
Skupaj	114600	6385	0.05		

Tabela 2.2: Metrike in alarmi

2.2.2 Primeri rezultatov za posamezne razrede, datoteke in metode

Kot smo že omenili, so rezultati statične analize izvorne kode sistema Android izjemno obširni. Zato smo se odločili, da rezultate podamo le za nekaj primerov, ki služijo kot nekakšna navodila za uporabo rezultatov.

Naši rezultati so del kratkega poročila programa CTMJava. Kratko poročilo smo izbrali zato, ker je dolgo preobsežno in je za naše potrebe dovolj pregled kratkega poročila rezultatov. Dolgo naročilo je namreč na voljo le v XML datoteki, ki v našem primeru vsebuje krepko preko 1090000 vrstic XML kode. Za predstavitev kratkega poročila smo izkoristili možnost programa CTMJava, da nam rezultate pretvori v HTML obliko. Pregledovanje rezultatov je tako dokaj preprosto in intuitivno. Rezultati so nam na voljo v poddirektoriju CTM2JAVA, direktorija v katerem smo izvajali meritev.

Za pregled rezultatov se najprej postavimo v njihov direktorij in v enem od brskalnikov poženemo datoteko index.html. Pred nami je povzetek rezultatov statičnega testiranja za izbrano izvorno kodo. Na koncu povzetka imamo podatke, ki smo jih predstavili že v poglavju 3.1. Drugi del povzetka pa predstavlja pregled alarmov za vse razrede iz izvorne kode sistema Android.

Na sliki 3.1 so rezultati za prvih 30 razredov iz izvorne kode. Na skrajni levi strani vidimo indikator, ki nam pove, kolikšna je stopnja alarma v procentih za izbrani razred. Nato vidimo ime razreda, sledijo pa še alarmi po metrikah. Kot lahko vidimo si sledijo najprej metrika ciklomatične kompleksnosti, število vrstic izvorne datoteke, število vrstic programske kode v izvorni datoteki, metrika, ki pove, če je odstotek komentarjev v določenih mejah. Potem so tu še halsteadov volumen, predvideno halsteadovo število hroščev in na koncu indeks možnosti vzdrževanja.

Če je vrednost, ki pripada metriki rdeče barve, pomeni, da je izven določenih mej (meje lahko spreminjamo v konfiguracijski datoteki). Za primer vzemimo razred BatteryStats. Kot vidimo so izven določenih meja vrednosti ciklomatične kompleksnosti, število vrstic programske kode, halsteadov volumen in vrednost za predvideno halsteadovo število hroščev. V primeru, da je katerakoli od matrik izven podanih meja, pa se v rdečo barvo obarva tudi ime razreda, za katerega je bila omenjena metrika izmerjena.

Alarms-%	Measured item	v(G)	LOCphy	LOCpro	c%	V	B	MI
	Bundle	86	1423	725-		26965-	8.21-	138
	TokenWatcher							
	Death	2	29	26	-	417	0.12	108
	TokenWatcher	16	175	122		3249	1.07	124
	Vibrator	5	68	34		544	0.13	142
	BatteryStats							
	Counter	1	18	4		104	0.02	162
	Timer	1	28	5		158	0.03	162
	Uid	2	162	47		2074	0.37	162
	BatteryStats	140-	1295	799-		49772-	10.83-	109
	MemoryFile							
	MemoryInputStream	6	59	51	-	1168	0.39	110
	MemoryOutputStream	2	19	16	-	424	0.12	106
	MemoryFile	24	233	141		4691	1.68	135
	Broadcaster							
	Registration	1	9	8		78-	0.01	113
	Broadcaster	27	194	160	-	5289	2.37-	99
	CountDownTimer	5	118	43		1219	0.39	135
	BadParcelableException	1	12	8		96-	0.02	161
	HandlerStateMachine							
	SmHandler	11	61	37		1053	0.24	120
	HandlerStateMachine	19	269	116		3762	0.83	135
	FileUtils							
	FileStatus	1	19	15		259	0.04	127
	FileUtils	20	169	112		4976	1.32	124
	Looper							
	HandlerException	2	14	12	-	241	0.07	114
	Looper	13	201	106		3662	0.81	138
	IServiceManager	1	52	16		611	0.14	154
	SystemClock	4	135	26	-	568	0.16	146
	BatteryManager	1	28	15		519	0.08	124
	RegistrantList	7	104	82	-	1924	0.64	133
	NetStat	15	223	123		4386	1.26	139
	ServiceManagerNative	9	85	67	-	1661	0.44	114
	ServiceManagerProxy	1	69	62	-	2104	0.53	104
	MailboxNotAvailableException	1	19	12		69-	0.02	152
	ParcelFormatException	1	13	8		80-	0.02	159
	AsyncResult	1	48	29	-	526	0.16	127
	Binder	17	271	120		4093	1.05	147
	BinderProxy	3	55	47	-	1403	0.32	122
	FileObserver							
	ObserverThread	5	59	47	-	1547	0.47	128
	FileObserver	7	120	96	-	3499	0.95	115
	RemoteMailException	1	12	10	-	69-	0.02	145

Slika 2.1: Rezultati po razredih

Če želimo, lahko pregledamo rezultate za točno določen razred oziroma datoteko. Metrike v podrobnejšem pogledu so iste kot v povzetku. Na voljo imamo dva načina dostopa do natančnejših podatkov. Prvi način je, da izberemo že na začetku pregledovanja podrobnejši pogled. S tem imamo pred seboj rezultate vseh razredov in njihovih metod. Drugi način podrobnega pogleda pa omogočimo s klikom na izbrani razred. Tako se nam odpre okno, v katerem je pogled postavljen na razred, katerega rezultati nas zanimajo. Na sliki 3.2 je primer podrobnega pogleda rezultatov za razred Process, ki se nahaja v datoteki Process.java.

File : ./os/Process.java
Package: android.os

Line	Measured item	v(G)	LOCphy	LOCpro	c%	V	B	MI
29	class ZygoteStartFailedEx							
31	ZygoteStartFailedEx()	1	5	1		5	0.00	183
36	ZygoteStartFailedEx()	1	1	1		25	0.00	154
37	ZygoteStartFailedEx()	1	1	1		25	0.00	154
38	Summary: ZygoteStartFailedEx	1	10	5		97-	0.02	175
40	class Process							
192	start()	4	65	30		679	0.17	113
258	start()	1	9	5		183	0.05	151
268	invokeStaticMain()	2	24	14		427	0.10	123
296	openZygoteSocketIfNeeded()	10	75	49-	-	1112-	0.27	93
372	zygoteSendArgsAndGetPid()	7	65	35		939	0.24	103
438	startViaZygote()	12-	81	50-		1564-	0.51	93
520	getElapsedCpuTime()	1	5	1		24	0.01	177
526	myPid()	1	5	1		24	0.01	177
532	myTid()	1	5	1		24	0.01	177
538	myUid()	1	4	1		24	0.01	180
543	getUidForName()	1	6	1		33	0.01	173
550	getGidForName()	1	6	1		33	0.01	173
557	setThreadPriority()	1	15	2		66	0.02	155
573	setThreadPriority()	1	17	2		53	0.01	154
591	getThreadPriority()	1	14	2		42	0.01	158
606	supportsProcesses()	1	7	1		24	0.01	172
614	setOomAdj()	1	12	1		47	0.01	160
627	setArgV0()	1	9	1		33	0.01	167
637	killProcess()	1	14	3		56	0.01	156
652	setUid()	1	2	1		32	0.01	186
655	setGid()	1	2	1		32	0.01	186
658	sendSignal()	1	7	1		47	0.01	169
666	getFreeMemory()	1	2	1		24	0.01	187
669	readProcLines()	1	3	2		70	0.02	170
673	getPids()	1	2	1		56	0.01	183
693	readProcFile()	1	3	2		110	0.02	168
697	getPss()	1	9	1		33	0.01	167
706	Summary: Process	31	667	248		10880-	2.24-	132

Summary: v(G): 31 LOCphy: 706 LOCbl: 100 LOCpro: 263 LOCcom: 345 ';': 145

Slika 2.2: Pregled podrobnih rezultatov

Na voljo imamo tudi pogled izvorne kode vseh metod, ki so vključene v rezultatih. Kodo prikličemo s klikom na izbrano metodo.

2.2.3 Rezultati za nekaj najbolj uporabljanih programov

Preverili smo zanesljivost naslednjih aplikacij, ki so že prednaložene na telefon HTC G1: budilka, brskalnik, kalkulator, koledar, kontakti, google iskalnik za namizje, email, mms, program za neposredno sporočanje, predvajalnik glasbe, apkbuilder, nastavitve in snemalnik zvokov. S tem izborom smo skušali izbrati različne aplikacije, ki so pogosto v vsakdanji uporabi.

1. Budilka

Metode (zgornja dva nivoja)		Razredi		Vmesniki	
169	Skupaj	29	Skupaj	4	Skupaj
9	Povpr. LOCpro	17	na najvišjem nivoju	0	na najvišjem nivoju
14	Povpr. LOCcom/LOCphy %	17	Razširi	0	Razširi
2	Povpr. v(G)	8	Implementira	0	Implementira

Direktoriji		Datoteke	
1	Skupaj	17	Skupaj
		3206	LOCphy
		2081	LOCpro
System		663	LOCcom
217	v(G)	463	LOCbl
100	MI brez komentarjev	1158	';'
28	MI teža komentarjev	188	Povpr. LOCphy
127	MI	20	Povpr. LOCcom/LOCphy %
		72	Bločni komentarji Jave

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	201	2	0	1-100	1-100	1-10
LOCpro	201	6	2	4-400	4-400	1-40
Komentar %	201	57	28	20-60	20-60	20-60
V	201	22	10	100-8000	100-8000	4-1000
B	201	3	3	0-2	0-2	n/a
MI	201	2	0	80-	80-	80-
Skupaj	1206	92	7			

2. Brskalnik

Metode (zgornja dva nivoja)		Razredi		Vmesniki	
545	Skupaj	63	Skupaj	3	Skupaj
12	Povpr. LOCpro	33	na najvišjem nivoju	0	na najvišjem nivoju
18	Povpr. LOCcom/LOCphy %	40	Razširi	0	Razširi
2	Povpr. v(G)	25	Implementira	0	Implementira

Direktoriji		Datoteke	
1	Skupaj	33	Skupaj
		13032	LOCphy
		9055	LOCpro
System		2713	LOCcom
1177	v(G)	1347	LOCbl
94	MI brez komentarjev	4658	';'
31	MI teža komentarjev	394	Povpr. LOCphy
125	MI	20	Povpr. LOCcom/LOCphy %
		204	Bločni komentarji Jave

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	610	18	2	1-100	1-100	1-10
LOCpro	610	36	5	4-400	4-400	1-40
Komentar %	610	177	29	20-60	20-60	20-60
V	610	69	11	100-8000	100-8000	4-1000
B	610	10	2	0-2	0-2	n/a
MI	610	22	3	80-	80-	80-
Skupaj	3660	332	9			

3. Kalkulator

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ====================   ====================
   106 Skupaj                        15 Skupaj                0 Skupaj
     7 Povpr.  LOCpro                14 na najvišjem nivoju   0 na najvišjem nivoju
     7 Povpr.  LOCcom/LOCphy %       10 Razširi               0 Razširi
     2 Povpr.  v(G)                   2 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ============================================
     2 Skupaj                        14 Skupaj
                                   1640 LOCphy
                                   1091 LOCpro
System                              298 LOCcom
==============================      256 LOCbl
   113 v(G)                         626 ';'
   107 MI brez komentarjev          117 Povpr.  LOCphy
    20 MI teža komentarjev           18 Povpr.  LOCcom/LOCphy %
   127 MI                             7 Bločni komentarji Jave

                                          Meje             Meje          Meje
Metrika    Izmerjenih  Alarmi  %     razred 1. nivo   razred 2. nivo     metod
================================================================================
v(G)              121       2   1       1-100            1-100            1-10
LOCpro            121       0   0       4-400            4-400            1-40
Komentar %        121      40  33      20-60            20-60            20-60
V                 121       1   0     100-8000         100-8000           4-1000
B                 121       0   0       0-2              0-2              n/a
MI                121       0   0      80-              80-               80-
================================================================================
Skupaj            726      43   5
```

4. Koledar

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ====================   ====================
   418 Skupaj                        67 Skupaj                2 Skupaj
    17 Povpr.  LOCpro                36 na najvišjem nivoju   1 na najvišjem nivoju
    14 Povpr.  LOCcom/LOCphy %       37 Razširi               0 Razširi
     3 Povpr.  v(G)                  28 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ============================================
     2 Skupaj                        37 Skupaj
                                  13154 LOCphy
                                   9405 LOCpro
System                             2295 LOCcom
==============================     1660 LOCbl
  1058 v(G)                        5790 ';'
    86 MI brez komentarjev          355 Povpr.  LOCphy
    28 MI teža komentarjev           17 Povpr.  LOCcom/LOCphy %
   114 MI                            83 Bločni komentarji Jave

                                          Meje             Meje          Meje
Metrika    Izmerjenih  Alarmi  %     razred 1. nivo   razred 2. nivo     metod
================================================================================
v(G)              487      29   5       1-100            1-100            1-10
LOCpro            487      44   9       4-400            4-400            1-40
Komentar %        487     196  40      20-60            20-60            20-60
V                 487      93  19     100-8000         100-8000           4-1000
B                 487      11   4       0-2              0-2              n/a
MI                487      36   7      80-              80-               80-
================================================================================
Skupaj           2922     409  13
```

5. Kontakti

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ====================   ====================
   322 Skupaj                        43 Skupaj                0 Skupaj
    17 Povpr.  LOCpro                16 na najvišjem nivoju   0 na najvišjem nivoju
    14 Povpr.  LOCcom/LOCphy %       32 Razširi               0 Razširi
     3 Povpr.  v(G)                  16 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ============================================
     2 Skupaj                        16 Skupaj
                                   9318 LOCphy
                                   6761 LOCpro
System                             1591 LOCcom
==============================     1100 LOCbl
   851 v(G)                        3703 ';'
    87 MI brez komentarjev          582 Povpr.  LOCphy
    28 MI teža komentarjev           17 Povpr.  LOCcom/LOCphy %
   115 MI                           163 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	365	22	6	1-100	1-100	1-10
LOCpro	365	34	9	4-400	4-400	1-40
Komentar %	365	159	43	20-60	20-60	20-60
V	365	58	15	100-8000	100-8000	4-1000
B	365	10	3	0-2	0-2	n/a
MI	365	17	4	80-	80-	80-
Skupaj	2190	300	13			

6. Google iskalnik za namizje

```
 Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ====================   ====================
    18 Skupaj                         3 Skupaj                0 Skupaj
     7 Povpr.  LOCpro                 2 na najvišjem nivoju   0 na najvišjem nivoju
    15 Povpr.  LOCcom/LOCphy %        3 Razširi               0 Razširi
     1 Povpr.  v(G)                   0 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ============================================
     1 Skupaj                        2 Skupaj
                                   302 LOCphy
                                   193 LOCpro
System                              72 LOCcom
==============================      39 LOCbl
    18 v(G)                         94 ';'
   107 MI brez komentarjev         151 Povpr.  LOCphy
    29 MI teža komentarjev          24 Povpr.  LOCcom/LOCphy %
   136 MI                            4 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	21	0	0	1-100	1-100	1-10
LOCpro	21	0	0	4-400	4-400	1-40
Komentar %	21	5	23	20-60	20-60	20-60
V	21	1	4	100-8000	100-8000	4-1000
B	21	0	0	0-2	0-2	n/a
MI	21	0	0	80-	80-	80-
Skupaj	126	6	4			

7. Email

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ======================   ======================
  1140 Skupaj                       136 Skupaj                 5 Skupaj
    10 Povpr.  LOCpro                84 na najvišjem nivoju   5 na najvišjem nivoju
    20 Povpr.  LOCcom/LOCphy %       84 Razširi               0 Razširi
     2 Povpr.  v(G)                  34 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ==============================================
    10 Skupaj                        90 Skupaj
                                  23082 LOCphy
                                  14769 LOCpro
System                             5660 LOCcom
==============================     2708 LOCbl
  1754 v(G)                        7648 ';'
    97 MI brez komentarjev          256 Povpr.  LOCphy
    32 MI teža komentarjev           24 Povpr.  LOCcom/LOCphy %
   129 MI                           526 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	1281	47	3	1-100	1-100	1-10
LOCpro	1281	61	4	4-400	4-400	1-40
Komentar %	1281	275	21	20-60	20-60	20-60
V	1281	119	9	100-8000	100-8000	4-1000
B	1281	23	2	0-2	0-2	n/a
MI	1281	39	3	80-	80-	80-
Skupaj	7686	564	7			

8. MMS

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ======================   ======================
  1716 Skupaj                       161 Skupaj                15 Skupaj
     9 Povpr.  LOCpro               132 na najvišjem nivoju   8 na najvišjem nivoju
    12 Povpr.  LOCcom/LOCphy %       94 Razširi               2 Razširi
     2 Povpr.  v(G)                  62 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ==============================================
    12 Skupaj                       140 Skupaj
                                  29493 LOCphy
                                  20125 LOCpro
System                             5373 LOCcom
==============================     4166 LOCbl
  2384 v(G)                       10172 ';'
   102 MI brez komentarjev          210 Povpr.  LOCphy
    26 MI teža komentarjev           18 Povpr.  LOCcom/LOCphy %
   128 MI                           298 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	1892	36	1	1-100	1-100	1-10
LOCpro	1892	62	3	4-400	4-400	1-40
Komentar %	1892	519	27	20-60	20-60	20-60
V	1892	127	6	100-8000	100-8000	4-1000
B	1892	23	2	0-2	0-2	n/a
MI	1892	42	2	80-	80-	80-
Skupaj	11352	809	7			

9. IM (neposredno sporočanje)

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ======================   ======================
  1678 Skupaj                       190 Skupaj               26 Skupaj
     9 Povpr.  LOCpro               128 na najvišjem nivoju  20 na najvišjem nivoju
    13 Povpr.  LOCcom/LOCphy %      104 Razširi               5 Razširi
     2 Povpr.  v(G)                  49 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ==============================================
     8 Skupaj                    |  148 Skupaj
                                 |29149 LOCphy
                                 |19570 LOCpro
System                           | 5658 LOCcom
==============================   | 3980 LOCbl
  2170 v(G)                      |10127 ';'
   102 MI brez komentarjev       |  196 Povpr.  LOCphy
    26 MI teža komentarjev       |   19 Povpr.  LOCcom/LOCphy %
   128 MI                        |  488 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	1894	24	1	1-100	1-100	1-10
LOCpro	1894	57	3	4-400	4-400	1-40
Komentar %	1894	613	32	20-60	20-60	20-60
V	1894	121	6	100-8000	100-8000	4-1000
B	1894	24	2	0-2	0-2	n/a
MI	1894	61	3	80-	80-	80-
Skupaj	11364	900	7			

10. Predvajalnik glasbe

```
Metode (zgornja dva nivoja)      Razredi                  Vmesniki
==============================   ======================   ======================
   467 Skupaj                        57 Skupaj                5 Skupaj
    15 Povpr.  LOCpro                32 na najvišjem nivoju   0 na najvišjem nivoju
     9 Povpr.  LOCcom/LOCphy %       47 Razširi               0 Razširi
     3 Povpr.  v(G)                  17 Implementira          0 Implementira

Direktoriji                      Datoteke
==============================   ==============================================
     4 Skupaj                    |   32 Skupaj
                                 |12219 LOCphy
                                 | 9501 LOCpro
System                           | 1523 LOCcom
==============================   | 1232 LOCbl
  1222 v(G)                      | 5267 ';'
    91 MI brez komentarjev       |  381 Povpr.  LOCphy
    23 MI teža komentarjev       |   12 Povpr.  LOCcom/LOCphy %
   114 MI                        |  105 Bločni komentarji Jave
```

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	528	32	6	1-100	1-100	1-10
LOCpro	528	51	9	4-400	4-400	1-40
Komentar %	528	249	47	20-60	20-60	20-60
V	528	90	17	100-8000	100-8000	4-1000
B	528	12	3	0-2	0-2	n/a
MI	528	29	5	80-	80-	80-
Skupaj	3168	463	14			

11. ApkBuilder

Metode (zgornja dva nivoja)		Razredi		Vmesniki	
14	Skupaj	2	Skupaj	0	Skupaj
20	Povpr. LOCpro	1	na najvišjem nivoju	0	na najvišjem nivoju
16	Povpr. LOCcom/LOCphy %	0	Razširi	0	Razširi
5	Povpr. v(G)	0	Implementira	0	Implementira

Direktoriji		Datoteke	
1	Skupaj	1	Skupaj
		468	LOCphy
		323	LOCpro
System		88	LOCcom
		64	LOCbl
62	v(G)	151	';'
82	MI brez komentarjev	468	Povpr. LOCphy
30	MI teža komentarjev	18	Povpr. LOCcom/LOCphy %
112	MI	8	Bločni komentarji Jave

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	16	2	12	1-100	1-100	1-10
LOCpro	16	2	12	4-400	4-400	1-40
Komentar %	16	6	37	20-60	20-60	20-60
V	16	4	25	100-8000	100-8000	4-1000
B	16	1	6	0-2	0-2	n/a
MI	16	1	6	80-	80-	80-
Skupaj	96	16	16			

12. Nastavitve

Metode (zgornja dva nivoja)		Razredi		Vmesniki	
1036	Skupaj	125	Skupaj	5	Skupaj
10	Povpr. LOCpro	83	na najvišjem nivoju	0	na najvišjem nivoju
10	Povpr. LOCcom/LOCphy %	108	Razširi	0	Razširi
2	Povpr. v(G)	48	Implementira	0	Implementira

Direktoriji		Datoteke	
7	Skupaj	83	Skupaj
		21897	LOCphy
		15180	LOCpro
System		3419	LOCcom
		3338	LOCbl
1924	v(G)	8513	';'
98	MI brez komentarjev	263	Povpr. LOCphy
24	MI teža komentarjev	15	Povpr. LOCcom/LOCphy %
122	MI	251	Bločni komentarji Jave

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	1166	35	3	1-100	1-100	1-10
LOCpro	1166	40	3	4-400	4-400	1-40
Komentar %	1166	444	38	20-60	20-60	20-60
V	1166	103	8	100-8000	100-8000	4-1000
B	1166	19	2	0-2	0-2	n/a
MI	1166	29	2	80-	80-	80-
Skupaj	6996	670	9			

13. Snemalnik zvokov

Metode (zgornja dva nivoja)		Razredi		Vmesniki	
57	Skupaj	4	Skupaj	1	Skupaj
12	Povpr. LOCpro	4	na najvišjem nivoju	0	na najvišjem nivoju
12	Povpr. LOCcom/LOCphy %	2	Razširi	0	Razširi
2	Povpr. v(G)	2	Implementira	0	Implementira

Direktoriji		Datoteke	
1	Skupaj	3	Skupaj
		1178	LOCphy
		860	LOCpro
System		137	LOCcom
		191	LOCbl
94	v(G)	557	';'
94	MI brez komentarjev	392	Povpr. LOCphy
26	MI teža komentarjev	11	Povpr. LOCcom/LOCphy %
120	MI	11	Bločni komentarji Jave

Metrika	Izmerjenih	Alarmi	%	Meje razred 1. nivo	Meje razred 2. nivo	Meje metod
v(G)	62	1	1	1-100	1-100	1-10
LOCpro	62	5	8	4-400	4-400	1-40
Komentar %	62	21	33	20-60	20-60	20-60
V	62	8	12	100-8000	100-8000	4-1000
B	62	1	3	0-2	0-2	n/a
MI	62	3	4	80-	80-	80-
Skupaj	372	39	10			

2.3 Interpretacija rezultatov

Program je več kot 1000 datotek z izvorno kodo pregledal v le nekaj minutah. Dobljeni rezultati za celoten sistem so dobri, saj program ne javi veliko alarmov. Pa še tisti, ki jih javi, niso tako kritični.

V tabeli 2 po deležu alarmov najbolj izstopata metriki odstotek komentarjev ter indeks možnosti vzdrževanja. Vendar noben delež ni tako kritičen. Skupni delež vseh alarmov je samo 5%, za kar smo mnenja, da je dober rezultat. Seveda bi se rezultate dalo prirediti, če bi sami spreminjali mejne vrednosti. Glede na to, kako kompleksen je operacijski sistem sam, je indeks možnosti vzdrževanja še v mejah normale. Nato po deležu odstotkov sledi Halsteadovo število predvidenih hroščev.

Vsekakor pa nam tako orodje, kot je CMTJava, še kako prav pride, da nas že prej opozori, čemu moramo posvetiti več pozornosti. Če tako zahteven projekt sproti testiramo, bo končni rezultat pravilnejše delovanje in bolj pregledna koda. Sploh že zaradi odprtokodne narave sistema Android.

Povprečno ciklomatično število ***v(G)*** za metode je 2, kar je krepko pod mejno vrednostjo 10. Indeks možnosti vzdrževanja (koda in komentarji skupaj) je 138, kar je nad vrednostjo 85. To kaže na lahko vzdrževanje kode v prihodnosti. Koda je dobro komentirana, saj kar 35% vseh vrstic predstavljajo komentarji. Seveda pa komentarjev ni preveč, kar bi lahko javil alarm.

Pri testiranju posameznih, bolj uporabnih aplikacij, so bili rezultati podobno dobri. Nekaj več alarmov je bilo zaradi manj komentirane kode in velikosti slovarja (vsota različnih operatorjev in različnih operandov) pri budilki, brskalniku, koledarju, kontaktih, odjemalcu elektronske pošte ter predvajalniku glasbe. Povprečni ***v(G)*** pri metodah je pri vseh programih znoraj mej, prav tako ***MI*** povsod kaže na lahko vzdreževanje kode v prihodnje. Največ predvidenih napak je pričakovanih pri programu Koledar; teh je 11. Izmed testiranih aplikacij je imel največ vrstic s programsko kodo (***LOCpro***) program za pošiljanje MMSov (20125).

Pri gradnji manjših aplikacij se ponavadi zadovoljimo že s tem, da aplikacija deluje (če ne dodajamo kakšne nove funkcionalnosti ponavadi izvorne kode nikoli več ne pogledamo). Pri večjih projektih pa je potrebno misliti tudi na prihodnost in nadgradnje. Treba je preveriti, kako težko bo vzdrževati kodo, kakšna je berljivost kode, kako dobro je komentirana, koliko napak se lahko pričakuje in podobno. Ker je ročno pregledovanje kode zelo zamudno,je za take namene najbolje uporabiti programe tipa CMTJava in analizo avtomatizirati.

Zanimivo bi bilo videti kakšno interno analizo podjetja Google in preveriti, kakšnih metod se pri tovrstnih testih poslužujejo oni in katera orodja uporabljajo.

Za dejansko predstavo, kakšni so rezultati pri posameznih metrikah, bi morali imeti rezultate kakšnega podobnega sistema.

Rezultati so nas prijetno presenetili, saj smo pričakovali več možnosti napak in večji delež alarmov, vsaj glede na število vrstic kode, ki jih ima sistem Android. Bi bilo pa vsekakor zanimivo spremljati razvoj, saj je zunaj takorekoč šele prva različica. In v tej fazi še lahko upoštevamo alarme o deležu komentarjev in indeksu možnosti vzdrževanja, sploh glede na to, da je projekt odprtokoden.

Ker se s tovrstno analizo ukvarjajo le velika podjetja, je na voljo zelo malo brezplačne programske opreme za take teste. Zato gre na tem mestu posebna zahvala podjetju Testwell, ker so nam omogočili dostop do programa v izobraževalne namene.

3. FTA

FTA (Fault Tree Analysis) je metoda za analizo zanesljivosti in varnosti, ki z uporabo Boolove logike kombinira osnovne (nižje-nivojske) dogodke v višje-nivojske dogodke. Metoda identificira, modelira in oceni medsebojna razmerja dogodkov, ki privedejo do odpovedi ali neželenih dogodkov/stanj.

Metoda je bila sprva razvita za projekte kjer so napake oz. odpovedi nesprejemljive. Leta 1962 je bila prvič uporabljena s strani ameriške vojske pri razvoju bombnikov ter medcelinskih balističnih raket. Kmalu za tem so jo začeli obširno uporabljati pri jedrskih elektrarnah v ZDA ter pri letalskemu proizvajalcu Boeing. Danes se uporablja pri vseh glavnih vejah inženiringa, predvsem pa na področju varnostnega inženiringa (safety engineering) za kvantitativno določanje verjetnosti določenega nezaželenega stanja ali nevarnosti.

FTD (Fault tree diagram) so logični dogodkovni diagrami, ki prikazujejo stanje sistema (korenski dogodek) glede na stanja njegovih komponent (osnovnih dogodkov). Diagram se gradi od korenskega dogodka navzdol (top-down pristop), tako da se za vsak dogodek definira vse možne načine kako se je lahko zgodil – pri tem je ključnega pomena izredno dobro poznavanje sistema. Dogodke se običajno povezuje s konvencionalnimi logičnimi vezniki (IN, ALI, ekskluzivni ALI, …). Kako globoko se drevo razvija ni definirano, odvisno je od specifičnega primera – običajno se dodaja dogodke, dokler se ne doseže neke (skoraj) neposredno obvladljive ravni, mnogokrat se gre tudi do nivoja integriranega vezja, skoraj nikoli pa v interno arhitekturo čipov samih.

Vsakemu dogodku v FTD se določi verjetnost nastanka. Ker je testiranje kompleksnih sistemov oz. dogodkov predrago ali pa sploh ni mogoče (recimo vesoljskih raketoplanov ne moremo izstreliti 1000 in statistično ugotovit verjetnost odpovedi, lahko pa to storimo za čipe, tranzistorje, razne materialov, itd…), se določi verjetnosti le za osnovne dogodke (liste) ter se preostale, vključno s korenskim, izračuna po pravilih verjetnostne analize. Torej drevo mora biti razvito do te mere, da lahko dobro določimo verjetnosti vseh osnovnih dogodkov.

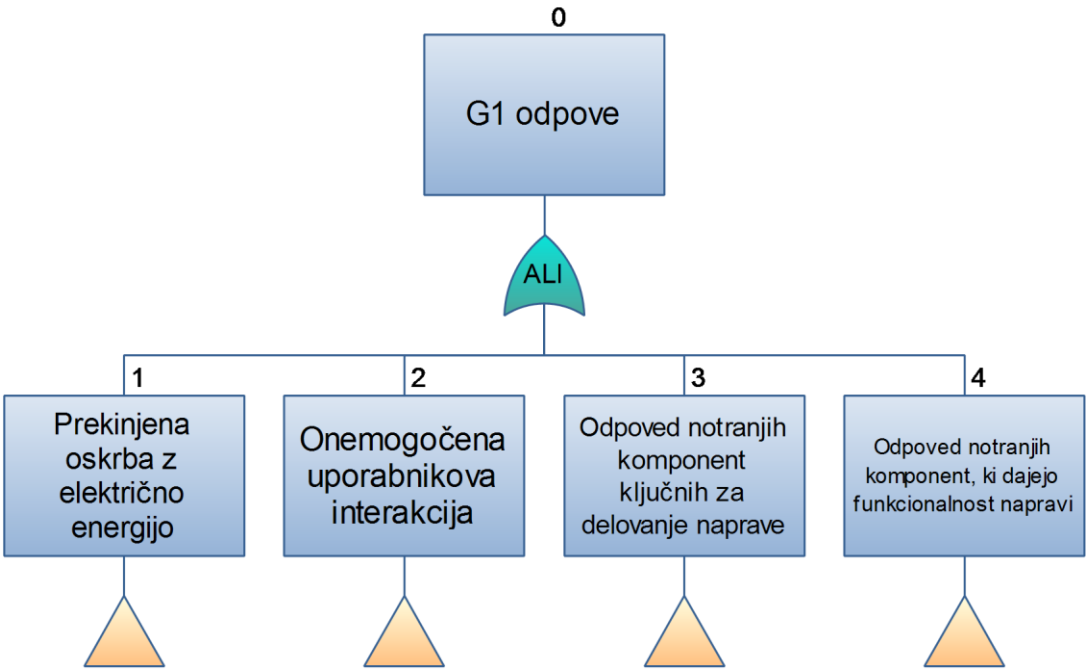
Pri ugotavljanju kateri vsi osnovni dogodki se morajo zgoditi istočasno, da se zgodi korenski dogodek si pomagamo s tako-imenovanimi cut-seti. To je vsaka množica osnovnih dogodkov, ki ob odpovedi vseh vsebovanih dogodkov povzroči odpoved korenskega dogodka. Minimalni cut-set je najmanjša množica osnovnih dogodkov, ki še povzroči odpoved korenskega – teh je običajno več in dejansko predstavljajo vse možne načine kako lahko sistema odpove (vse možne kombinacije osnovnih dogodkov, ki povzročijo odpoved).

Glavni koraki FTA analize:

1. Definirati sistem, korenski dogodek (potencialne napake / odpovedi) ter robne pogoje
2. Zgraditi FTD (fault tree diagram)
3. Poiskati minimalne cut-sete
4. Kvalitativno analizirati drevo (izboljšati / popraviti drevo)
5. Kvantitativno analizirati – izračunati verjetnosti vseh dogodkov ter minimalnih cut-setov
6. Poročati rezultate oz. jih uporabiti za boljšo kontrolo nad ugotovljenimi nevarnostmi v sistemu

3.1 Opredelitev vzrokov in načinov odpovedi

FTD 0 – G1 odpove

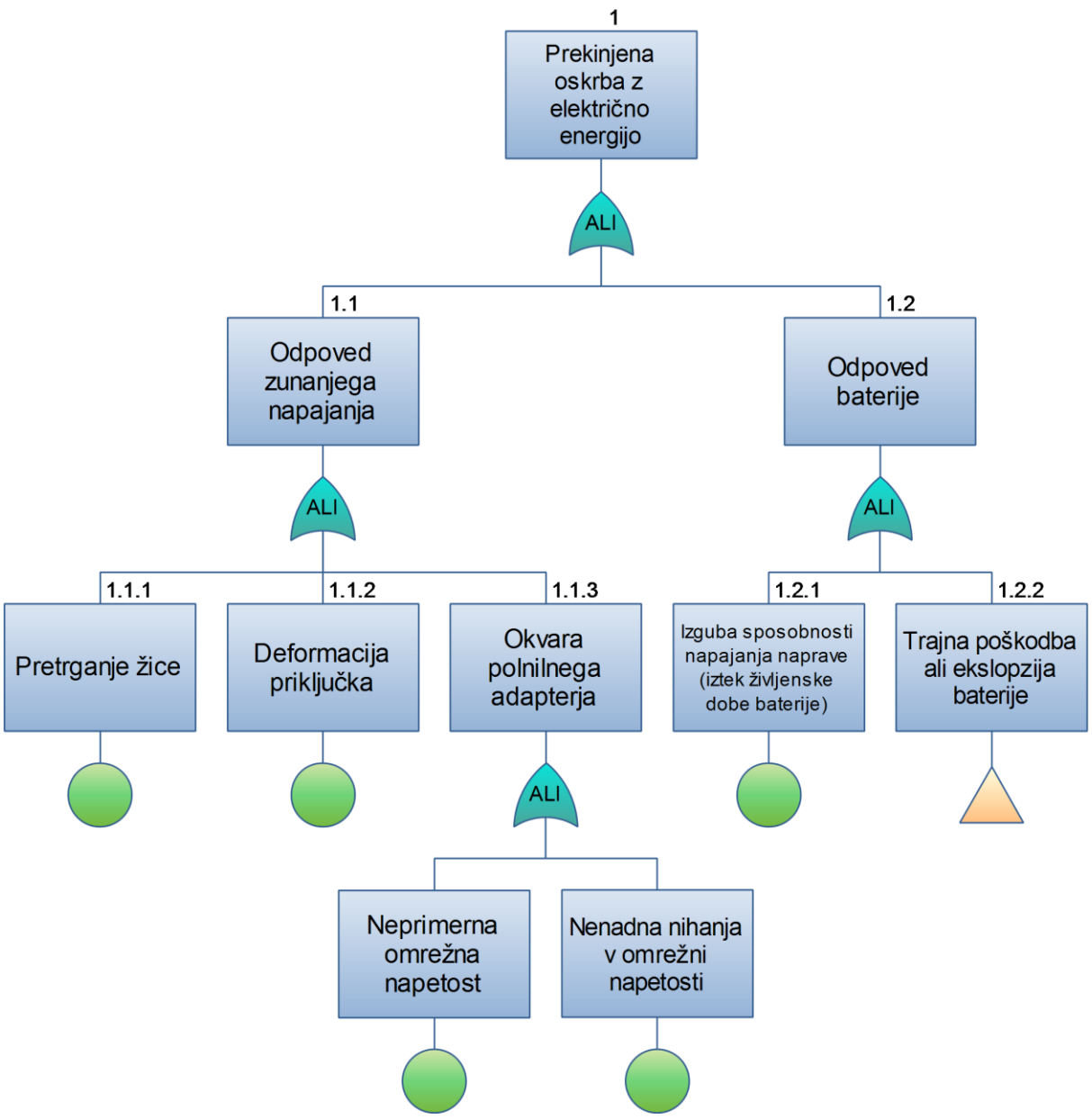


Kot korenski dogodek FTA diagrama izberemo odpoved mobilnega telefona G1. Do tega stanja lahko pridemo zaradi različnih vzrokov. Dogodke lahko razdelimo v štiri glavne veje drevesa, ki v sebi združujejo sorodne teme:

- Prekinjena oskrba z električno energijo
- Onemogočena uporabnikova interakcija
- Odpoved notranjih komponent ključnih za delovanje naprave
- Odpoved notranjih komponent ki dajejo napravi funkcionalnost.

V nadaljevanju si bomo pobliže ogledali vse štiri veje drevesne strukture in opisali vzroke in posledice za posamezne dogodke.

FTD 1 – Prekinjena oskrba z električno energijo



G1 se napaja iz zamenljive litij-ionske baterije, katero polnimo s pomočjo priloženega adapterja, ki ga priključimo na električno omrežje. Tako lahko odpovesta baterija, kar povzroči takojšnji konec delovanja naprave ali polnilec, kar pomeni, da bo naprava delovala do izpraznitve baterije. Kot kritična upoštevamo oba dogodka – dasiravno odpoved adapterja ne pomeni nujno takojšnjega prenehanja delovanja, je avtonomija naprave ob intenzivni uporabi kratka (približno 2-3h). Potrebno je redno polnjenje, kar pomeni da je G1 brez delujočega polnilca tako-rekoč neuporaben.

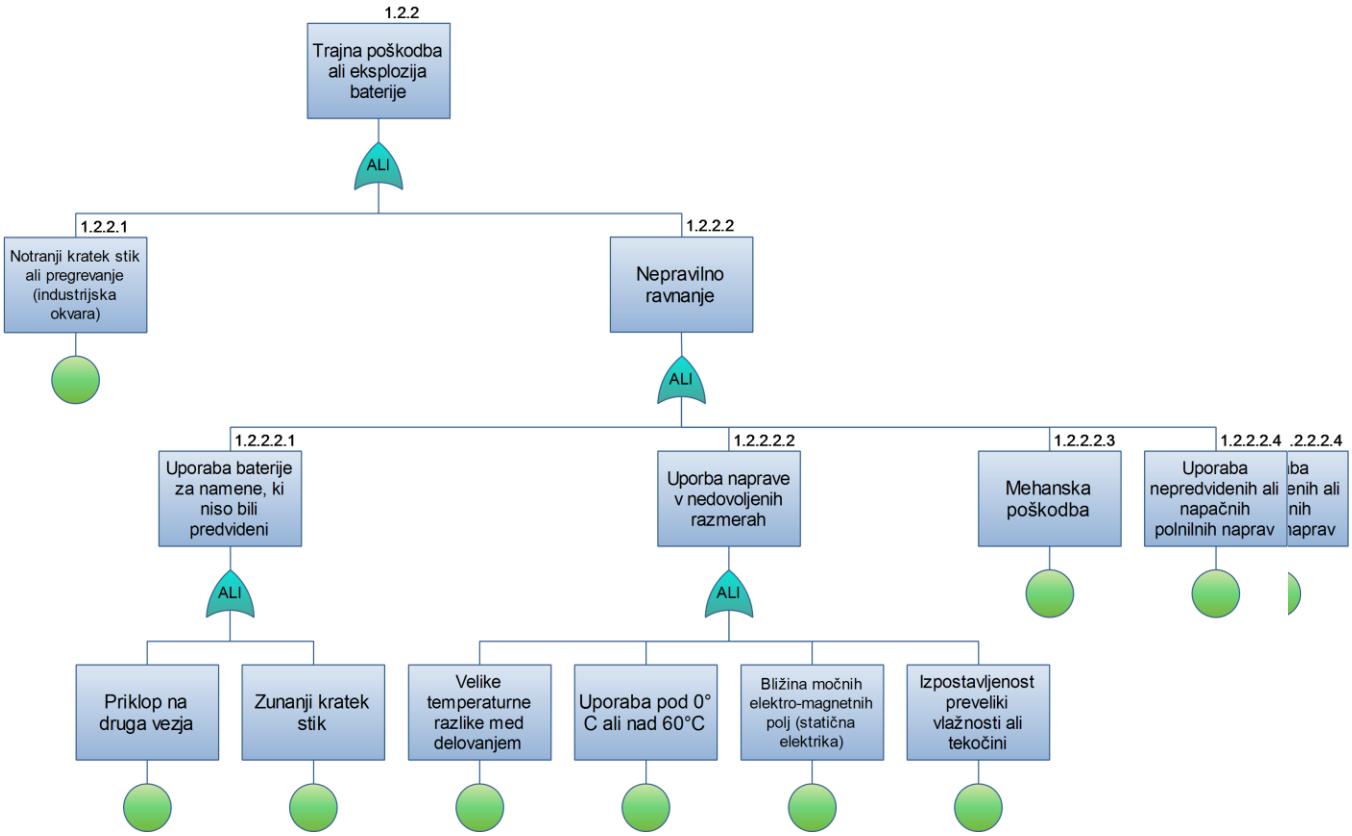
FTD 1.1 – Odpoved zunanjega napajanja

Tu je navedenih nekaj najverjetnejših vzrokov, ki bi lahko privedli do odpovedi polnilnega adapterja. Ena od možnosti je pretrganje kabla, druga pa, da je zaradi grobega ravnanja prišlo do deformacije priključka na napravi in tako ni mogoče več priklopiti telefona na polnilec. Ker imajo različne države različne napetosti v električnih omrežjih, lahko priklop polnilca v električno omrežje za katero ni bil izdelan privede do takojšnjega uničenja le-tega. Nevarna so tudi razna nihanja omrežne napetosti npr. zaradi udara strele, kar seveda povzroči takojšnje uničenje adapterja.

FTD 1.2 – Odpoved baterije

Baterija lahko odpove iz različnih vzrokov, ena možnost je, da se ji je iztekla življenjska doba tj. njena kapaciteta se je skozi življenjski cikel tako zmanjšala, da ni več sposobna napajati napravo. Druga opcija je, da se je baterija med uporabo kritično poškodovala in je zato izgubila sposobnost hranjenja električne energije oz. napajanja naprave.

FTD 1.2.2 – Trajna poškodba ali eksplozija baterije



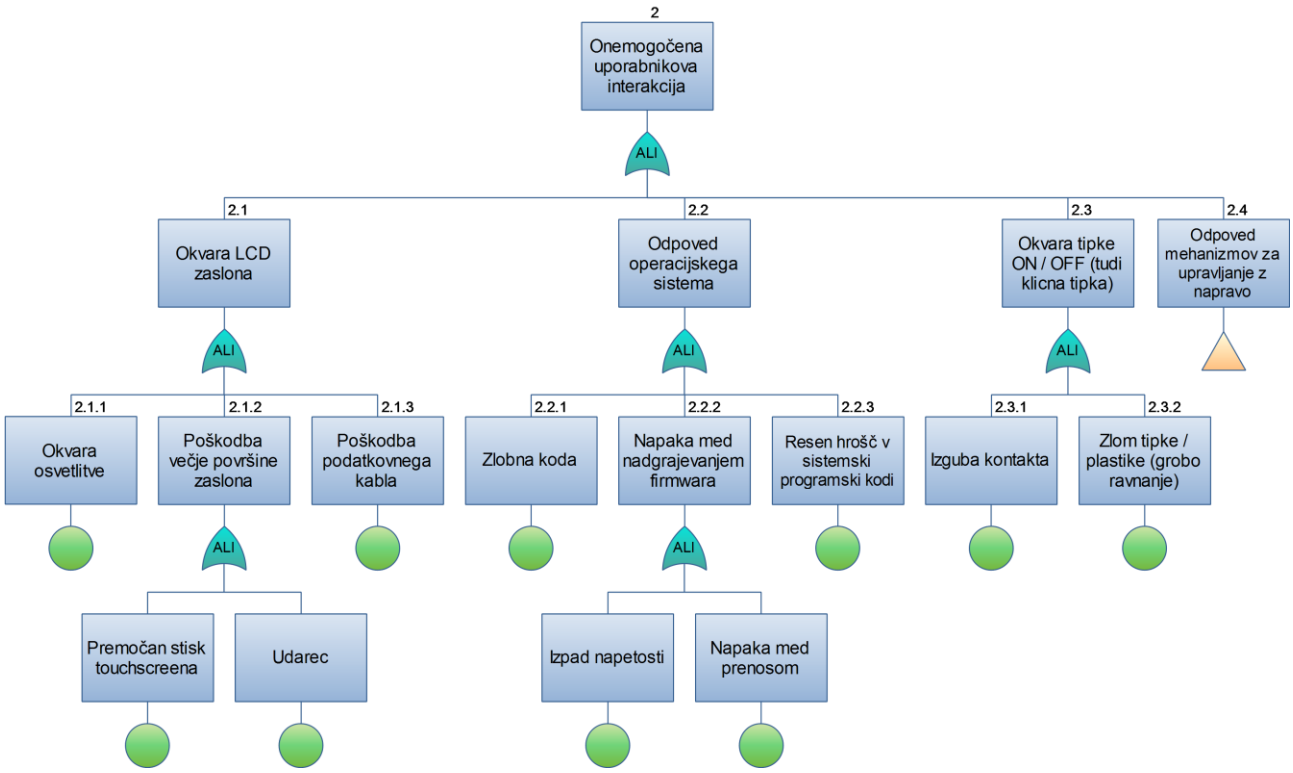
Do odpovedi baterije lahko pride zaradi nepravilnega ravnanja uporabnika oz. neupoštevanja navodil proizvajalca. Obstaja tudi verjetnost, da je prišlo do napake pri izdelavi. Ob kritični proizvodni napaki lahko baterija postane tempirana bomba. Znani so nekateri primeri, ko je prišlo do eksplozije litij-ionske baterije (predvsem baterij za prenosnike) in je nato proizvajalec pozval potrošnike naj baterije iz določene serije vrnejo.

FTD 1.2.2.2 – Nepravilno ravnanje

Okvaro baterije lahko povzroči tudi malomarno ravnanje uporabnika. Možne so razne mehanske poškodbe zaradi grobega ravnanja, ki privedejo do odpovedi. Uporaba improviziranih polnilnih naprava ali napačnih polnilcev lahko tudi močno skrajša življenjsko dobo baterije, če jo ne celo uniči. Koriščenje baterije za namene, ki niso bili predvideni tj. napajanje drugih naprav (vezij) oziroma povzročitev zunanjega kratkega stika lahko ravno tako ogrozi brezhibno delovanje baterije in s tem celotne naprave.

Potrebno je upoštevati tudi proizvajalčeva navodila glede okolja v katerem baterija deluje. Odpoved lahko povzročijo ekstremne zunanje temperature (nad 60°C, pod 0°C), nevarne so tudi hitre spremembe temperature okolja. Močna elektromagnetna polja lahko uničijo tako baterijo, kot seveda vse ostale elektronske komponente naprave. Velika vlažnost oziroma padec naprave v vodo je ravno tako lahko usoden tako za baterijo kot celotno napravo.

FTD 2 – Onemogočena uporabnikova interakcija



Za nemoteno uporabo naprave G1 je potrebna stalna interakcija med napravo in uporabnikom. V primeru, da pride do poškodb ključnih delov, je naprava neuporabna. Zaslon predstavlja pomembno komponento pri uporabi naprave, medtem ko tipkovnica ni nujna. Tipkovnico lahko nadomestimo, saj nam naprava lahko zazna črke napisane z roko. Zato je nismo uvrstili na drevo.

FTD 2.1 – Okvara LCD zaslona

Zaslon nam predstavlja izhod naprave, zato je pomembno, da deluje brezhibno ter natančno. Do poškodbe večje površine zaslona pride zaradi uporabnikove neprevidnosti, če premočno stisne zaslon občutljiv na dotik oz. če zaslon dobi udarec (padec na tla, udarec v mizo med nošenjem v žepu, …). Zaradi udarca je možno, da se pokvari tudi osvetlitev LCD zaslona. Osvetljen je iz zadnje strani z LED diodami. LCD je tako praktično neuporaben, saj je branje z njega zelo oteženo. V močnem soncu in temi pa praktično nemogoče. LCD preneha delovati, če pride do poškodbe podatkovnega kabla. Kabel se lahko samo iztakne iz ležišča, lahko pa ga tudi mehansko poškodujemo.

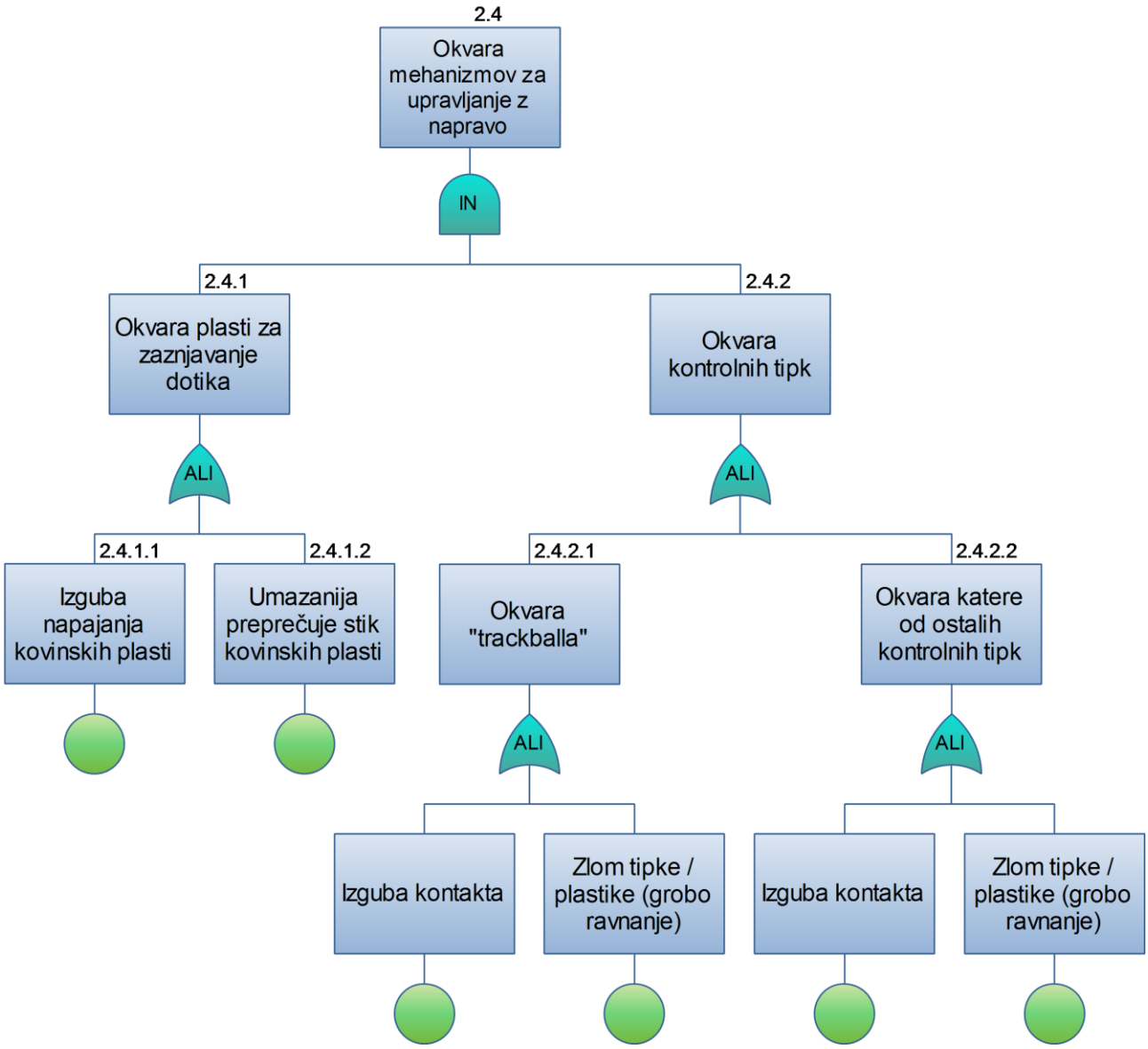
FTD 2.2 – Odpoved operacijskega sistema

G1 lahko preneha delovati tudi zaradi programskih vzrokov. Verjetnost je, da naprava »zmrzne« med posodabljanjem »firmware«-a. To se lahko zgodi zaradi izpada napajanja ali napake med prenosom. Ker ima naprava sodoben OS, ni imuna na zlonamerno kodo (virusi, spyware, malware). Težavo se da rešiti z proti-virusnim programom za Android. Naprava lahko ne deluje, če uporabimo programsko opremo z resnim hroščem v programski kodi, oz. še ne preizkušeno programsko opremo ali pa če že sistemska programska oprema (OS) vsebuje kakšno kritično pomanjkljivost. V najhujših primerih je potreben izbris vseh podatkov in nalaganje novega »firmware«-a.

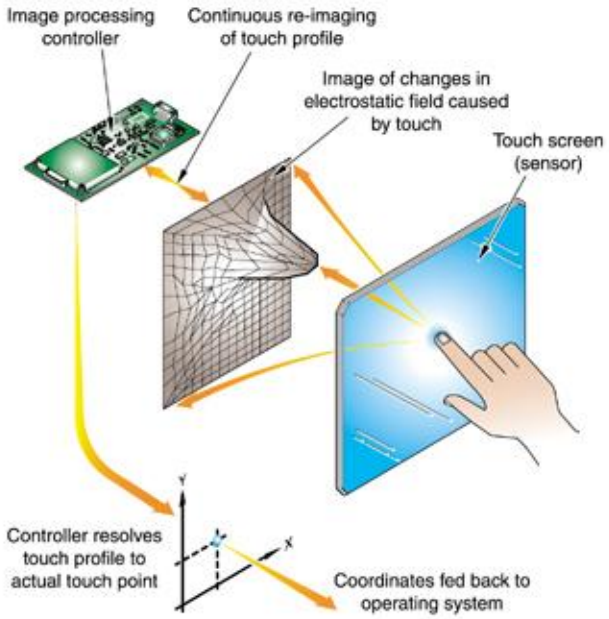
FTD 2.3 – Okvara tipke ON/OFF (tudi klicna tipka)

Problem interakcije z napravo bi imeli tudi, če bi se nam pokvarila glavna tipka za vklop/izklop naprave, tipka ON/OFF. Prav tako bi se odpoved lahko zgodila zaradi izgube kontakta (slab nanos prevodnega materiala, dotrajanost) ter zloma tipke ob grobem ravnanju.

FTD 2.4 – Okvara mehanizmov za upravljanje z napravo



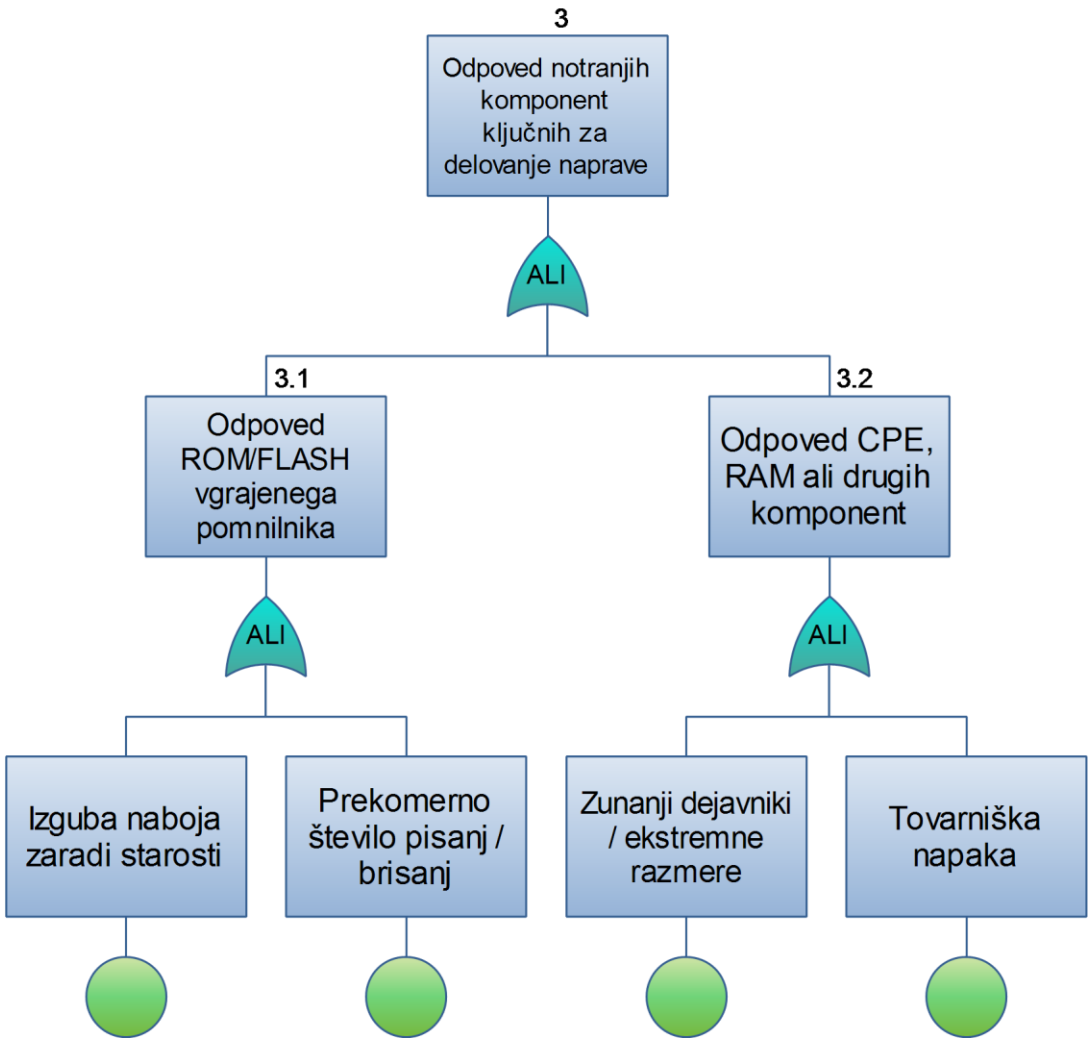
Resna težava bi bila, če bi odpovedala plast zaslona, ki skrbi za zaznavanje dotika. V napravo je vgrajen LCD zaslon občutljiv na dotik, ki temelji na kapacitivni tehnologiji. Plast za zaznavanje dotika preneha delovati, če zaslon izgubi napajanje oz. se med plasti prikrade umazanija, ki moti normalno delovanje. Problem pri interakciji z napravo bi imeli tudi, če bi odpovedale kontrolne tipke ali »trackball«. Odpoved bi se lahko zgodila zaradi izgube kontakta (slab nanos prevodnega materiala, dotrajanost) ter zloma tipke ob grobem ravnanju.



Slika 3.1: Shematski prikaz delovanja kapacitivnega zaslona občutljivega na dotik

Pri kapacitivnem sistemu je v kapacitivni plasti, ki je položena čez stekleno ploščo shranjen električni naboj. Pri dotiku zaslona se delček naboja prenese na uporabnika in s tem se naboj na kapacitivni plasti zmanjša. To zmanjšanje naboja je merjeno v vezjih, ki se nahajajo v vseh štirih kotih zaslona. Krmilnik izračuna oddaljenost od vsakega kota zaslona in poda koordinate dotika gonilnikom, ki to informacijo prevedejo tako, da jo operacijski sistem razume. Dobra lastnost te vrste zaslonov na dotik je da prepušča 90 % svetlobe kar prinaša jasno in čisto sliko. Taki zasloni naj bi zdržali več 100 milijonov dotikov na eno piko. Prednosti kapacitivnih zaslonov so tudi velika resolucija, jasnost in čistost slike ter neobčutljivost na maščobo, vlago ter umazanijo.

FTD 3 – Odpoved notranjih komponent ključnih za delovanje naprave

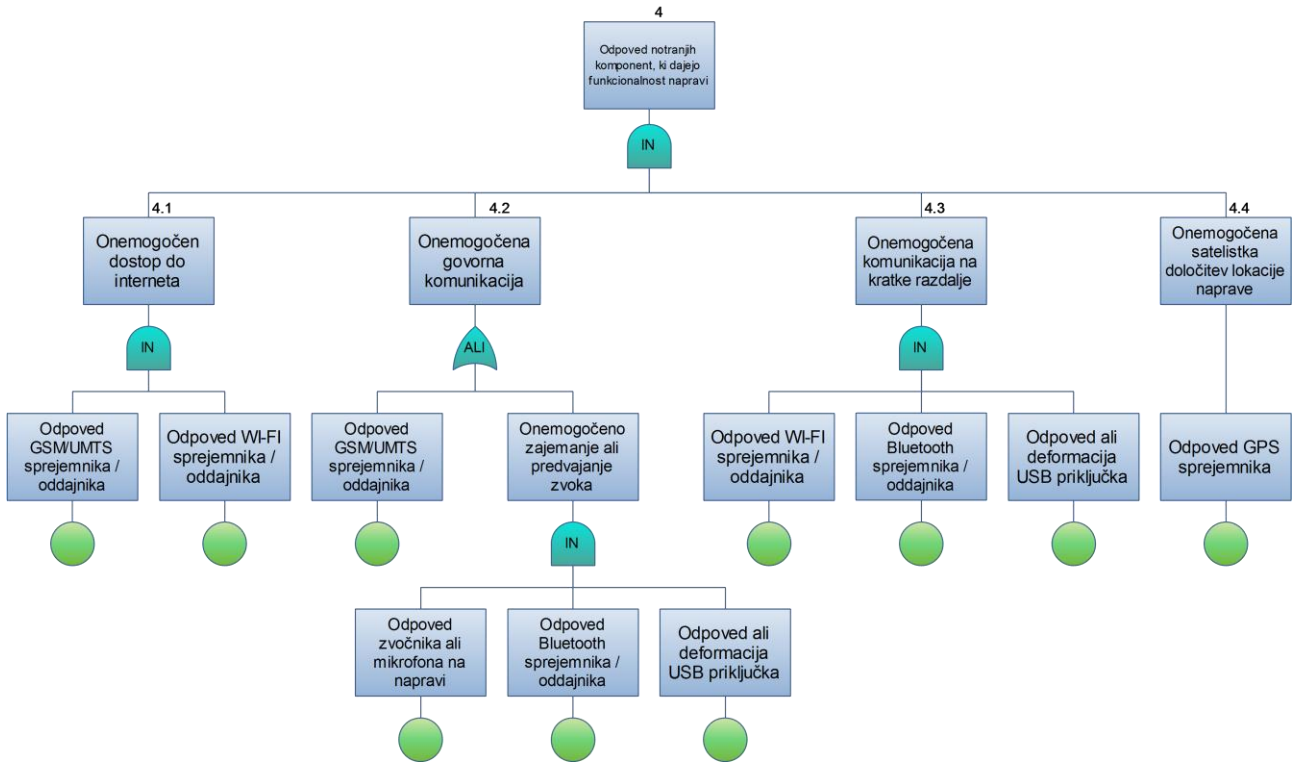


Za delovanje naprave in samo uporabnost potrebujemo pomnilnik. G1 ima vgrajena 192 MB RAM pomnilnika ter 256 MB Flash pomnilnika. Možno je vgraditi še dodaten pomnilnik na microSD kartici, ki pa ni ključen za delovanje naprave.

Pomnilnik nam lahko odpove zaradi izgube naboja, starosti, prekomerno število brisanj, ekstremnih razmer ter tovarniške napake. Število brisanj za flash EEPROM vezja je okoli 100000 kar je težko dosegljivo ob normalni uporabi. Tolikšno število dostopov je v kratkem času mogoče doseči ob nenormalni uporabi. Na primer, če se okužimo z zlonamerno kodo, ki nam napada en blok.

Naprava ima vgrajen Qualcomm MSM 7201A 528 MHz procesor, ki je ključnega pomena za delovanje naprave. Kot integrirano vezje je zanesljiva komponenta, vendar je občutljiv na pregrevanje (neprimerno zračenje, poviševanja takta procesorja) in previsoko napetost na vezju.

FTD 4 – Odpoved notranjih komponent, ki dajejo funkcionalnost napravi



G1 lahko uporabljamo na veliko različnih načinov. Zaradi visoke zmogljivosti ga lahko primerjamo s pravim, malo manj zmogljivim, računalniškim sistemom, kot pa le z mobilnim telefonom. Vsebovane funkcije segajo od aplikacij operacijskega sistema (za pisanje besedil in dokumentov, za računanje, za datum in čas, za kompas ter zemljevide, …) preko video kamere, zaslona, mikrofona in zvočnikov (za snemanje in ogled vizualnih in zvočnih posnetkov, za govorno komunikacijo) do povezav z zunanjim svetom (za klicanje, satelitsko določanje pozicije, …), prenosa podatkov od/do drugih naprav ter dostopom do interneta. Katere funkcionalnosti so bolj in katere manj pomembne je odvisno od posameznega uporabnika, toda G1 je v osnovi še vedno mobilni telefon oz. naprava za komuniciranje (sicer bi uporabnik najverjetneje kupil oz. izbral napravo, ki se bolje prilega njegovim načrtovanim potrebam po funkcionalnostih) zato so notranje komponente, ki to zagotavljajo ključnega pomena za delovanje celotne naprave. Ko vsi načini komunikacije odpovejo, smatramo da je naprava neuporabna (sicer pa je spet odvisno od uporabnika do uporabnika – za nekoga je lahko postane naprava neuporabna če že samo npr. Bluetooth odpove). V nadaljevanju so našteti okvirno vsi načini za interakcijo z zunanjim svetom, ki jih omogoča G1.

FTD 4.1 – Onemogočen dostop do interneta

Odpoved GSM/UMTS sprejemnika/oddajnika ali krmilnika, ki ga nadzoruje onemogoči globalni dostop do interneta, medtem ko odpoved WI-FI sprejemnika/oddajnika onemogoči dostop preko pristopnih točk (access points). Razlogi za odpoved teh komponent so enaki kot za večino elektronskih vezij / naprav – predvsem tovarniške napake oz. slaba kakovost, obrabljenost s časom, ekstremni pogoji ali fizične poškodbe (npr. zaradi nemarnosti ali nepazljivosti).

FTD 4.2 – Onemogočena govorna komunikacija

Govorna komunikacija je seveda takoj onemogočena, če odpove GSM/UMTS sprejemnik/oddajnik. Druga možnost je, da nam odpove zajemanje ali predvajanje zvoka. To pomeni, da odpovesta zvočnik ali mikrofon na G1, toda v tem primeru lahko na napravo še vedno priključimo slušalke z mikrofonom preko MiniUSB priključka ali Bluetooth povezave. Ko tudi ta dva načina odpovesta, je govorna komunikacija onemogočena.

FTD 4.3 – Onemogočena komunikacija na kratke razdalje

Tu imamo v mislih predvsem kabelsko povezavo preko USB kabla ter WI-FI ali Bluetooth brezžični povezavi kratkega dosega na osebni ali prenosni računalnik oz. na vse naprave, ki to omogočajo (dlančniki, netbooki, drugi mobilni telefoni, ...). To je ključna funkcionalnost za učinkovit prenos podatkov na/z G1. Če se zaradi fizične poškodbe deformira oz. uniči MiniUSB priključek, je kabelska povezava onemogočena, če pa nam odpovesta še WI-FI ter Bluetooth sprejemnika / oddajnika ostanemo tudi brez brezžične povezave.

FTD 4.4 – Onemogočena določitev lokacije naprave

Pregledovanje zemljevidov ter satelitsko določanje naprave (GPS) in njenega uporabnika je ena od najpomembnejših funkcij G1. Okvara GPS sprejemnika to onemogoči, razlogi za odpoved pa so enaki kot pri prejšnjih treh tipih oddajnikov / sprejemnikov.

3.2 Interpretacija rezultatov

Popolna FTA analiza bi morala vsebovati še verjetnosti za odpoved vsakega dogodka na diagramu in seveda skupno verjetnost za korenski dogodek – odpoved naprave. Dogodke na diagramu lahko klasificiramo v dve skupini – tiste, ki so posledica uporabnikovega ravnanja z napravo ter tiste, ki so posledica proizvodnega procesa, torej so odgovornost proizvajalca naprave. Za prvo skupino je določanje verjetnosti praktično nemogoče, saj bi morali izvajati raziskave nad širšo populacijo ter sklepat povzetke o njihovem značaju in ravnanju, poleg tega pa bi bile ogromne razlike med različnimi skupnostmi ali geografskimi lokacijami (npr. verjetnost da bo naprava izpostavljena temperaturam pod 0°C je veliko večja v severnih državah kot pa v ekvatorialnih). Za dogodke iz druge skupine je določanje verjetnosti večinoma izvedljivo, toda običajno le proizvajalcu naprave, ki lahko statistično beleži število reklamacij in rezultate testiranj posameznih vzorcev ali komponent. Torej tudi ti podatki nam niso na razpolago, poleg tega pa tudi ne moremo vedeti točno katere vse komponente vsebuje naprava, predvsem na nižjih nivojih. Zaradi teh razlogov smo se že na začetku odločili, da bomo analizo „uporabniško“ orientirali – vanjo vključili dogodke, ki so po našem mnenju najbolj verjetni, da se lahko zgodijo povprečnemu uporabniku – npr. pod možnost, da odpove CPE smo našteli „tovarniška napaka“, kar je za povprečnega uporabnika čisto dovolj, če pa bi delali analizo s stališča proizvajalca bi najverjetneje morali našteti veliko več dogodkov (npr. napaka v določen proizvodnem procesu, napaka v arhitekturi …) in nas dogodki kot so „zunanji dejavniki / ekstremni pogoji“ ne bi zanimali, saj bi mi sami določili te pogoje (pogoje za uporabo) – torej pravila, ki bi se jih moral uporabnik držati.

Iz diagramov lahko razberemo kateri deli oz. komponente so bistvene za zagotavljanje delovanja naprave in iz hierarhije dogodkov lahko razberemo kakšne so odvisnosti med posameznimi komponentami. Povprečen uporabnik se seznani s tem kaj lahko pričakuje, da se bo z napravo zgodilo skozi čas (s staranjem) in na kaj mora biti pazljiv med njeno uporabo. Vidimo, da so skoraj vse relacije tipa OR – to pomeni, da ima naprava zelo nizko redundanco – skoraj vsak od naštetih osnovnih dogodkov bo pokvaril napravo. Edini redundanci sta pri mehanizmih za upravljanje z napravo (oznaka 2.4) – toda tudi tu bi vsaka okvara močno zmanjšala uporabnost naprave – in pri komponentah, ki dajejo napravi funkcionalnost (oznaka 4) – katera je tako ali tako subjektivna stvar vsakega posameznika. Za redundanco oz. povečanje zanesljivosti lahko poskrbi proizvajalec, toda v tem primeru se očitno ne splača – to je logično, saj je pri takih tipih naprav najbolj verjetno, da se bo le-ta pokvarila zaradi uporabnikovega ravnanja (v primeru mobilnih telefonov zaradi kakšne fizične poškodbe) in ne zaradi proizvodnih napak. To ne pomeni, da je G1 nezanesljiva naprava (v kategoriji mobilnih telefonov), temveč dogodki, ki smo jih izbrali dokazujejo ravno nasprotno, da se bo naprava najbrž prej uničila zaradi nepravilne uporabe kot pa zaradi prirojenih napak. V resnici pa bo čas (ob seveda dovolj velikem številu uporabnikov) najboljši pokazatelj zanesljivosti te naprave.

4. FMEA

Failure Mode and Effects Analysis (*FMEA*) so metodologije, namenjene predvidevanju možnih načinov okvar (ang. failure modes) proizvoda ali procesa, zato, da bi ocenili tveganje posameznih okvar, jih na podlagi tveganja razvrstili in izvedli ukrepe za odpravo ali ublažitev najbolj resnih posledic (ang. effects analysis).

Pri načinih odpovedi se osredotočamo na kakršnekoli napake ali defekte v procesu, pri snovanju ali izdelavi izdelka. Posvetimo se predvsem tistim napakam in defektom, ki vplivajo na končnega uporabnika. Pri analizi posledic pa se obravnava posledice načina odpovedi.

FMEA se zelo pogosto uporablja v proizvodnji dejavnosti v različnih fazah življenjskega loka izdelka, vedno pogosteje pa se uporablja tudi v storitveni dejavnosti. Čeprav je bila FMEA v 50-ih letih razvita za vojaško industrijo, se danes uporablja v mnogih dejavnostih npr. pri razvoju programske opreme in proizvodnji hrane, v zdravstvu, itd., zato obstajajo različne implementacije FMEA analize. Raznolike organizacije in industrije so postopke prilagodile, da ustrezajo specifičnim zahtevam njihovih izdelkov oz. procesov in tako je nastalo več standardov in smernic. Med bolj pomembnimi standardi je *MIL-STD-1629A*, ki se uporablja v vojaški industriji in *SAE J1739* ter *AIAG FMEA-3*, ki sta bila razvita za potrebe avtomobilske industrije.

Navkljub razlikam med različnimi standardi in smernicami, so vsem FMEA analizam skupni sledeči koraki:

- Izbor izdelka ali procesa za analizo
- Identifikacija funkcij in načinov, posledic in povzročiteljev odpovedi izbranega izdelka ali procesa
- Ocena tveganja identificiranih težav
- Razporeditev težav glede na tveganje
- Določitev ukrepov za zmanjšanje tveganja oz. odpravo odpovedi
- Izvedba ukrepov
- Ponovna ocena tveganja

4.1 Ocena tveganja

Tipična FMEA vključuje tudi metodo za oceno tveganja (ang. *risk evaluation method*) potencialnih problemov identificiranih v korakih analize načinov odpovedi. Dve izmed pogostejših metoda sta ti. *Risk Priority Numbers* in *Criticality Analysis*.

4.1.1 Risk Priority Numbers

Pri metodi *Risk Priority Number* morajo biti izvedeni naslednji koraki:

- S številom od 1 do 10 ocenimo resnost posledic za vsako možno odpoved (ang.Severity oz. S). Ocena 1 pomeni, da posledice odpovedi za uporabnika praktično niso zaznavne. Bolj ko se z oceno bližamo številu 10, bolj to pomeni, da odpoved proizvod ali proces pri trenutni zasnovi postane neuporaben morda celo ogroža uporabnikovo zdravje.
- S številom od 1 do 10 ocenimo pogostost vsakega vzroka odpovedi (ang. Occurence oz. O). Ocena 1 pomeni, da je pogostost oz. verjetnost vzroka odpovedi pri izdelku zelo majhna. Bolj ko se z oceno bližamo številu 10, bolj pogost je vzrok za odpoved izdelka ali procesa pri trenutni zasnovi.
- S številom od 1 do 10 ocenimo verjetnost, da bo vzrok odpovedi zaznan preden bo posledice opazil oz. čutil končni uporabnik (ang. Detection oz. D). Ocena 1 pomeni, da je vzrok odpovedi zlahka zaznati. Ocena 10 pomeni, da je pri trenutni zasnovi izdelka oz. procesa vzrok odpovedi praktično nemogoče zaznati.
- Izračunamo število RPN, ki je zmnožek teh treh ocen,
  RPN = Resnost (S) x Pogostost (O) x Zaznavnost (D)

Na podlagi števila RPN, ki ga izračunamo za vsako možno odpoved, lahko možne odpovedi razporedimo v tabelo in najvišje uvrščene poizkušamo odpraviti, zmanjšati njihovo resnost ali pogostost ali pa povečamo sposobnost zaznavanje vzroka odpovedi. Izboljšava lahko predstavlja ponovna zasnova dela procesa oz. izdelka, menjava problematične komponente, povečanje redudance, omejitev območja delovanja izdelka, itd. Po izvedbi ukrepov, ki spremenijo izdelek ali proces, moramo ponovno oceniti možnost odpovedi, preračunati RPN in izvesti teste, s katerimi potrdimo izboljšave.

4.1.2 Critically Analysis

Standard *MIL-STD-1629A* definira dve metodi analize kritičnosti. Kvalitativno in kvantitativno.

Kvantitativni metodi se izvaja po slečih korakih:

- Definiramo zanesljivost za vsak objekt in na podlagi ocen izračunat napoved pričakovanih odpovedi pri danem obratovalnem času
- Identificiramo razmerje objekta, ki je potencialno nezanesljiv in je lahko odgovoren za odpoved
- Ocenimo resnost posledic posameznega načina odpovedi
- Izračunamo kritičnost vsakega potencialnega načina odpovedi z množenjem treh faktorjev,
  Kritičnost načina odpovedi = Pričakovane odpovedi x Razmerje nezanesljivosti x Verjetnost izgube
- Izračunamo kritčnost vsakega objekta s seštevkom vseh kritičnosti načinov odpovedi,
  Ocena kritičnosti = Vsota kritičnosti načinov odpovedi

Pri **kvalitativni analizi kritičnosti** ocenimo tveganje in prioriteto odpravljanja tveganj, po sledečih korakih:

- Ocenimo resnost posledic potencialnega načina odpovedi
- Ocenimo pogostost pojavljanja potencialnega načina odpovedi
- Primerjamo načine odpovedi v *matriki kritičnosti*, v kateri horizontalna os označuje resnost in vertikalna os označuje pogostost načina odpovedi

FMEA metodo z analizo kritičnosti pogosto označimo kot *Failure Modes, Effets and Criticality Analysis* oz. *FMECA.*

4.1.3 Uporaba in prednosti FMEA / FMECA analize

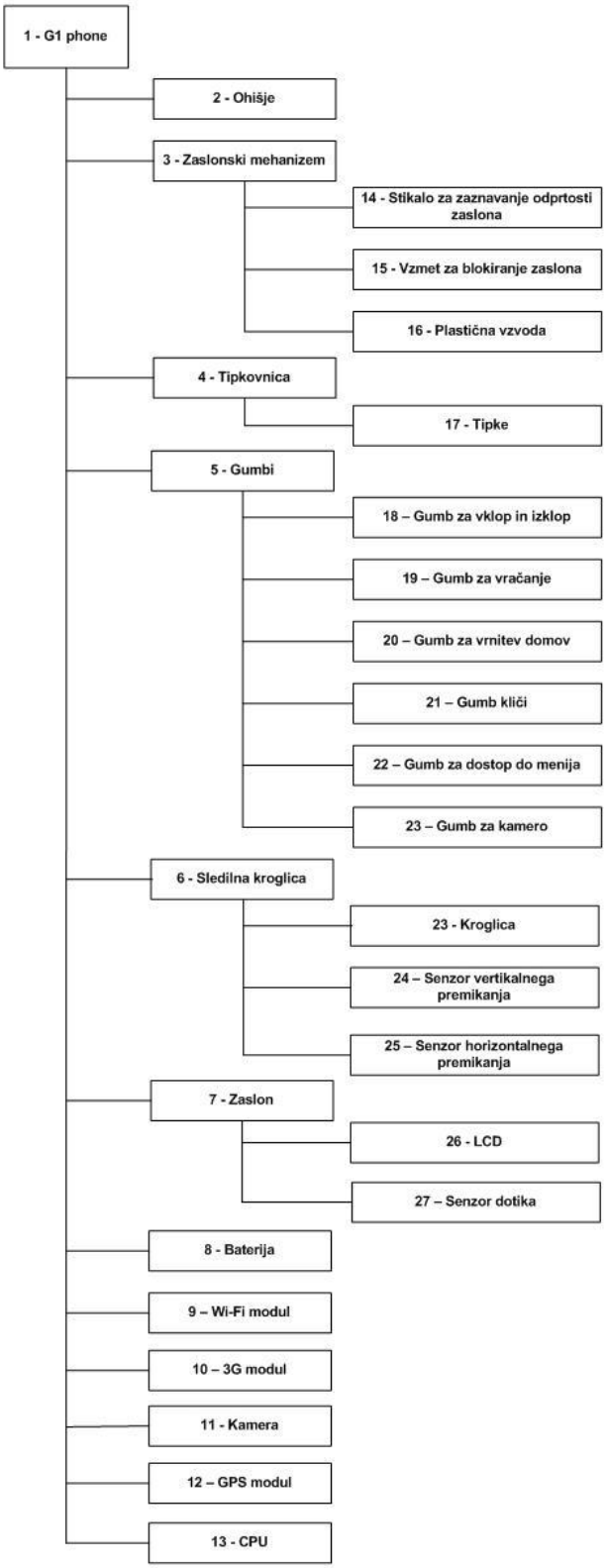
Uporaba FMEA lahko prispeva k izboljšanju zasnove izdelka ali procesa, kar prinaša večjo zanesljivost, boljšo kakovost, večjo varnost, zmanjšanje stroškov izmeta in reklamacij in povečano zadovoljstvo uporabnikov. Izvedene analize nam zagotavljajo bazo znanja načinov odpovedi in informacije, ki lahko služijo kot pomoč pri odpravljanju težav v prihodnosti ali pa kot orodje za usposabljanja novih inženirjev.

4.2 FMEA analiza naprave HTC G1

FMEA analizo smo izvedli s pomočjo orodja XFMEA, ki se je izkazal za zelo učinkovitega. Nudi nam vse funkcionalnosti, ki jih definira standard. Omogoča nam tudi prilagoditev postopkov za specifične potrebe projekta.

Analizo smo izvedli v več fazah. Najprej smo določili elemente sistema in njihove funkcije. Povezali smo jih v smiselno drevo, ki predstavlja hiearahijo sistema. Ker smo analizo delali v študijske namene in nismo imeli realnih podatkov o odpovedih, smo le te smiselno določili kar sami. Analizo smo nadaljevali po postopku, ki ga opisuje standard FMEA in je predstavljen zgoraj.

Hiearhija sistema



4.2.1 FMEA tabela

Item/Function	Failure	Effect	Si	Cause	Oi	Di	RPNi		Action	Sr	Or	Dr	RPNr
1 - G1 phone													
1.2 - Ohišje													
Zaščita občutljivih komponent pred mehanskimi poškodbami	Fizična poškodba ohišja	Nedelovanje celotne naprave	8	Padec naprave z višine	6	4	192						
				Slaba kvaliteta materjala	3	6	144						
		Odpadanje pokrova baterije	6	Polomljeni zatiči	6	3	108						
1.3 - Zaslonski mehanizem													
1.3.14 - Stikalo za zaznavanje odprtosti zaslona													
Obračanje pogleda	Stikalo ne spremeni stanja	Pogled se ob odprti tipkovnici ne spremeni	7	Stikalo za zaznavanje odprtosti zaslona ne deluje	4	3	84						
				Stikalo za zaznavanje odprtosti zaslona se včasih zatakne	5	8	280						
1.3.15 - Vzmet za blokiranje zaslona													
Stabilizacija zaslona v končnih položajih	Vzmet ne opravlja svoje funkcije	Zaslon ne stoji v končnih položajih	8	Vzmet manjka	3	2	48			8			
				Vzmet je prekratka	5	5	200		Instalacija dodatne kamere, ki pregleduje vzmeti ob vstopanju v proces sestavljanja naprave	8	2	5	80
				Vzmet je počena	3	4	96			8			
1.3.16 - Plastična vzvoda													
Odpiranje zaslona	Mehanizem je ukrivljen	Zaslon ni poravnan z ohišjem	4										
	Mehanizem je v okvari	Tipkovnica je nedostopna	7	Zatič, ki sprošča zaslon se je zablokiral	3	4	84						
1.4 - Tipkovnica													
1.4.17 - Tipke													
Vnos znakov	Pokrov tipke odpade	Onemogočen vnos znaka	5	Slaba kvaliteta lepila	4	7	140						
				Mehanska poškodba pokrova tipke	3	4	60						
	Tipka ne deluje	Onemogočen vnos znaka	5	Mehanska poškodba tipke	4	5	100						
1.5 - Gumbi													
1.5.18 - Gumb za vklop in izklop													
Vklop/izklop telefona	Gumb ne deluje	Vklop/Izklop telefona ni možen	9	Mehanska poškodba tipke	4	4	144			9			
				Napačna montaža tipke	6	4	216		Modifikacija	9	1	4	36

									roke, ki montira tipko				
Prekinitev klica	Gumb ne deluje	Prekinitev klica je otežena	5	Mehanska poškodba tipke	4	4	80						
				Napačna montaža tipke	6	4	120						
1.5.19 - Gumb za vračanje													
Vračanje v prejšnji meni	Gumb ne deluje	Vračanje je zelo oteženo	6	Mehanska poškodba tipke	4	4	96						
				Napačna montaža tipke	6	4	144						
1.5.20 - Gumb za vrnitev domov													
Vrnitev domov	Gumb ne deluje	Vračanje domov je možno samo z gumbom za nazaj	3	Mehanska poškodba tipke	4	4	48						
				Napačna montaža tipke	6	4	72						
1.5.21 - Gumb kliči													
Sproži klic	Gumb ne deluje	Klicanje je oteženo	4	Mehanska poškodba tipke	4	4	64						
				Napačna montaža tipke	6	4	96						
1.5.22 - Gumb za dostop do menija													
Odpiranje menija	Gumb ne deluje	Dostop do menuja je otežen	4	Mehanska poškodba tipke	4	4	64						
				Napačna montaža tipke	6	4	96						
		Dostop do menija nekaterih aplikacij ni možen	6	Mehanska poškodba tipke	4	4	96						
				Napačna montaža tipke	6	4	144						
1.5.28 - Gumb za kamero													
Hitri vklop kamere	Gumb ne deluje	Hitri vklop kamere ni možen, vklop kamere možen le preko osnovnega menija	3	Mehanska poškodba tipke	4	4	48						
				Napačna montaža tipke	6	4	72						
Zajem posnetka ob pritisku	Gumb ne deluje	Zahteva za zajem slike ni možna, zajem slike možen le preko gumba na zaslonu	3	Mehanska poškodba tipke	4	4	48						
				Napačna montaža tipke	6	4	72						
1.6 - Sledilna kroglica													
1.6.23 - Kroglica													
Izbiranje predmetov/objektov na zaslonu s pritiskom kroglice	Odpoved tipke	Izbira s kroglico ni možna	5	Prah pod kroglico	5	5	125						
				Vlaga se je iz kroglice prenesla na tipko	5	4	100						
1.6.24 - Senzor vertikalnega premikanja													
Vertikalna navigacija po meniju	Odpoved vertikalnega senzorja	Vertikalna navigacija po meniju ni možna	6	Prah pod kroglico	5	5	150						
				Vlaga se je iz kroglice prenesla na senzor	5	4	120						
				Mehanska poškodba senzorja	5	4	120						

1.6.25 - Senzor horizotalnega premikanja													
Horizontalna navigacija po meniju	Odpoved horizontalnega senzorja	Horizontalna navigacija po meniju ni možna	3	Prah pod kroglico	5	5	75						
				Vlaga se je iz kroglice prenesla na senzor	5	4	60						
				Mehanska poškodba senzorja	5	4	60						
1.7 - Zaslon													
1.7.26 - LCD													
Prikaz slike	Odpoved delov zaslona	Otežena uporaba naprave	7	Vlaga v zaslonski matriki	4	4	112						
				Mehanska poškodba zaslona	6	4	168						
	Odpoved celotnega zaslona	Uporaba naprave ni možna	9	Vlaga v zaslonski matriki	4	4	144			9			
				Mehanska poškodba zaslona	6	4	216		Namestitev dodatne absorbcijske prozorne folije med zaslon in ohišje telefona	9	4	4	144
				Ni stika na konektorju	4	5	180			9			
1.7.27 - Senzor dotika													
Navigacija po menijih in aplikacijah	Netočna detekcija dotika	Otežena uporaba naprave	7	Ukrivljena površina zaslona	4	3	84						
	Detekcija dotika ne deluje	Zelo otežena uporaba naprave	8	Slab stik na konektorju	4	3	96						
				Mehanska poškodba površine za detekcijo	6	4	192						
1.8 - Baterija													
Napajanje naprave	Obcutno zmanjsanje kapacitete v kratkem času	Oteženo delovanje naprave ob uporabi baterijskega napajanja	5	Izpostavljenost baterije visokim temperaturam	3	8	120						
	Vžig baterije	Naprava je uničena (zgori)	10	Previsoka temperatura baterije (zaradi kovinskih delcev v elektrolitu), posledično se posamezna celica z elektrolitom vname - zaradi kratkega stika	5	7	350		Zamenjava dobavitelja baterij	10	2	7	140
	Prekinjanje napajanja	Naprava se sama od sebe ugaša	7	Oksidacija konektorja	5	5	175						
1.9 - Wi-Fi modul													
Povezovanje z brezžičnimi omrežji	Nedelovanje modula	Nezmožnost povezovanja z brezžičnimi omrežji	6	Slaba integracija modula	3	4	72						
				Fizična poškodba	3	5	90						
1.10 - 3G modul													
- Hitri prenos podatkov (HSDPA) - Telefoniranje - Pošiljanje SMS in MMS sporočil	Odpoved UMTS antene	Nezmožnost delovanja naprave na frekvencah 3G standarda (nezmožnost uporabe 3g omrežja)	7	Fizična poškodba kontaktov zaradi tresljajev	8	4	224		Instalacija dodatnega varovala konektorja antene	7	1	4	28

1.11 - Kamera													
Fotografiranje	odpoved (avtomatskega) fokusiranja slike	Slaba kvaliteta posnetih slik, težko prepoznati objekte na sliki	5	Odpoved vezja (programa), ki fokusiranje zahteva	3	3	45						
				Odpoved motorja, ki fokusiranje izvaja	4	5	100						
	odpoved CCD senzorja (kamere)	Nezmožnost zajema slike	7	Odpoved CCD senzorja (slaba integracija)	2	2	28						
				Slaba kvaliteta izdelave CCD senzorja	2	5	70						
Snemanje videa	odpoved CCD senzorja (kamere)	Nezmožnost zajema videa	7	Odpoved CCD senzorja (slaba integracija)	2	2	28						
				Slaba kvaliteta izdelave CCD senzorja	2	5	70						
1.12 - GPS modul													
Določanje geografskega položaja	Odpoved modula	Spremljanje položaja naprave ni mogoče	7	Slaba integracija, slab stik	3	2	42						
	Napačna interpretacija podatkov o položaju naprave	Spremljanje položaja naprave je nesmiselno zaradi "napačnih" podatkov	6	Odpoved/okvara sprejemnika	4	4	96						
				Okvara ure sprejemnika	2	3	36						
1.13 - CPU													
Poganjanje OS in aplikacij	Odpoved CPU, posledično OS ne deluje	Naprava,njene osnovne funkcije in njeni moduli ne delujejo	9	Pregretje procesorja zaradi slabe postavitve procesorja na vezje ali preobloženega vezja	4	5	180						
				Fizična poškodba procesorja	2	3	54						
				Slaba integracija	2	2	36						

Tabela 4.1: FMEA tabela

4.2.2 Pregled ocen tveganj

Cause #	Cause	Occ (Init)	Occ (Rev)	Det (Init)	Det (Rev)	Cause RPN (Initial)	Item #	Item Name
1.2.1.1	Previsoka temperatura baterije (zaradi kovinskih delcev v elektrolitu), posledično se posamezna celica z elektrolitom vname - zaradi kratkega stika	5		7		350	1.8	Baterija
1.1.1.2	Stikalo za zaznavanje odprtosti zaslona se včasih zatakne	5		8		280	1.3.14	Stikalo za zaznavanje odprtosti zaslona
1.1.1.1	Fizična poškodba kontaktov zaradi tresljajev	8	1	4	4	224	1.10	3G modul
1.1.1.2	Napačna montaža tipke	6	1	4	4	216	1.5.18	Gumb za vklop in izklop
1.2.1.2	Mehanska poškodba zaslona	6	4	4	4	216	1.7.26	LCD
1.1.1.2	Vzmet je prekratka	5	2	5	5	200	1.3.15	Vzmet za blokiranje zaslona
1.1.1.1	Padec naprave z višine	6		4		192	1.2	Ohišje

1.2.1.2	Mehanska poškodba površine za detekcijo	6		4		192	1.7.27	Senzor dotika
1.2.1.3	Ni stika na konektorju	4		5		180	1.7.26	LCD
1.1.1.1	Pregretje procesorja zaradi slabe postavitve procesorja na vezje ali preobloženega vezja	4		5		180	1.13	CPU
1.3.1.1	Oksidacija konektorja	5		5		175	1.8	Baterija
1.1.1.2	Mehanska poškodba zaslona	6		4		168	1.7.26	LCD
1.1.1.1	Prah pod kroglico	5		5		150	1.6.24	Senzor vertikalnega premikanja
1.1.1.2	Slaba kvaliteta materjala	3		6		144	1.2	Ohišje
1.1.1.1	Mehanska poškodba tipke	4		4		144	1.5.18	Gumb za vklop in izklop
1.1.1.2	Napačna montaža tipke	6		4		144	1.5.19	Gumb za vračanje
1.1.2.2	Napačna montaža tipke	6		4		144	1.5.22	Gumb za dostop do menija
1.2.1.1	Vlaga v zaslonski matriki	4		4		144	1.7.26	LCD
1.1.1.1	Slaba kvaliteta lepila	4		7		140	1.4.17	Tipke
1.1.1.1	Prah pod kroglico	5		5		125	1.6.23	Kroglica
1.1.1.3	Mehanska poškodba senzorja	5		4		120	1.6.24	Senzor vertikalnega premikanja
2.1.1.2	Napačna montaža tipke	6		4		120	1.5.18	Gumb za vklop in izklop
1.1.1.2	Vlaga se je iz kroglice prenesla na senzor	5		4		120	1.6.24	Senzor vertikalnega premikanja
1.1.1.1	Izpostavljenost baterije visokim temperaturam	3		8		120	1.8	Baterija
1.1.1.1	Vlaga v zaslonski matriki	4		4		112	1.7.26	LCD
1.1.2.1	Polomljeni zatiči	6		3		108	1.2	Ohišje
1.1.1.2	Vlaga se je iz kroglice prenesla na tipko	5		4		100	1.6.23	Kroglica
1.1.1.2	Odpoved motorja, ki fokusiranje izvaja	4		5		100	1.11	Kamera
1.2.1.1	Mehanska poškodba tipke	4		5		100	1.4.17	Tipke
1.1.1.3	Vzmet je počena	3		4		96	1.3.15	Vzmet za blokiranje zaslona
1.1.1.1	Mehanska poškodba tipke	4		4		96	1.5.19	Gumb za vračanje
1.1.1.2	Napačna montaža tipke	6		4		96	1.5.21	Gumb kliči
1.1.1.2	Napačna montaža tipke	6		4		96	1.5.22	Gumb za dostop do menija
1.1.2.1	Mehanska poškodba tipke	4		4		96	1.5.22	Gumb za dostop do menija
1.2.1.1	Slab stik na konektorju	4		3		96	1.7.27	Senzor dotika
1.2.1.1	Odpoved/okvara sprejemnika	4		4		96	1.12	GPS modul
1.1.1.2	Fizična poškodba	3		5		90	1.9	Wi-Fi modul
1.2.1.1	Zatič, ki sprošča zaslon se je zablokiral	3		4		84	1.3.16	Plastična vzvoda
1.1.1.1	Ukrivljena površina zaslona	4		3		84	1.7.27	Senzor dotika

1.1.1.1	Stikalo za zaznavanje odprtosti zaslona ne deluje	4		3		84	1.3.14	Stikalo za zaznavanje odprtosti zaslona
2.1.1.1	Mehanska poškodba tipke	4		4		80	1.5.18	Gumb za vklop in izklop
1.1.1.1	Prah pod kroglico	5		5		75	1.6.25	Senzor vertikalnega premikanja
1.1.1.2	Napačna montaža tipke	6		4		72	1.5.20	Gumb za vrnitev domov
1.1.1.1	Slaba integracija modula	3		4		72	1.9	Wi-Fi modul
1.1.1.2	Napačna montaža tipke	6		4		72	1.5.28	Gumb za kamero
2.1.1.2	Napačna montaža tipke	6		4		72	1.5.28	Gumb za kamero
1.2.1.2	Slaba kvaliteta izdelave CCD senzorja	2		5		70	1.11	Kamera
2.1.1.2	Slaba kvaliteta izdelave CCD senzorja	2		5		70	1.11	Kamera
1.1.1.1	Mehanska poškodba tipke	4		4		64	1.5.21	Gumb kliči
1.1.1.1	Mehanska poškodba tipke	4		4		64	1.5.22	Gumb za dostop do menija
1.1.1.2	Vlaga se je iz kroglice prenesla na senzor	5		4		60	1.6.25	Senzor vertikalnega premikanja
1.1.1.3	Mehanska poškodba senzorja	5		4		60	1.6.25	Senzor vertikalnega premikanja
1.1.1.2	Mehanska poškodba pokrova tipke	3		4		60	1.4.17	Tipke
1.1.1.2	Fizična poškodba procesorja	2		3		54	1.13	CPU
1.1.1.1	Mehanska poškodba tipke	4		4		48	1.5.20	Gumb za vrnitev domov
1.1.1.1	Mehanska poškodba tipke	4		4		48	1.5.28	Gumb za kamero
2.1.1.1	Mehanska poškodba tipke	4		4		48	1.5.28	Gumb za kamero
1.1.1.1	Vzmet manjka	3		2		48	1.3.15	Vzmet za blokiranje zaslona
1.1.1.1	Odpoved vezja (programa), ki fokusiranje zahteva	3		3		45	1.11	Kamera
1.1.1.1	Slaba integracija, slab stik	3		2		42	1.12	GPS modul
1.2.1.2	Okvara ure sprejemnika	2		3		36	1.12	GPS modul
1.1.1.3	Slaba integracija	2		2		36	1.13	CPU
1.2.1.1	Odpoved CCD senzorja (slaba integracija)	2		2		28	1.11	Kamera
2.1.1.1	Odpoved CCD senzorja (slaba integracija)	2		2		28	1.11	Kamera

Tabela 4.2: Pregled ocen tveganja

4.2.3 Ukrepi za znižanje najvišjih tveganj

Iz zgornje tabele lahko razberemo vzroke odpovedi, ki so urejeni padajoče po oceni tveganja. Določili smo najvišjo oceno tveganja (RPN=200), ki je še sprejemljiva za končni produkt. Vse vzroke, ki imajo oceno tveganja nad to mejo smo morali ublažiti. To smo naredili z sprejetjem ukrepov (actions), ki jih prikazuje naslednja tabela.

Action #	Action	Due Date	Person Responsible	Action Taken	Completion Date	Action Category	Action Priority	Item #	Item Name	Cause #	Cause
1	Zamenjava dobavitelja baterij	13.5.2009	Mirko Strela Boljše baterije d.o.o.	Specifikacija nove baterije, izdelava in sprememba procesa sestavljanja	13.5.2009	Testing	High	1.8	Baterija	1.2.1.1	Previsoka 42rocess42ure baterije (zaradi kovinskih delcev v elektrolitu), posledično se posamezna celica z elektrolitom vname – zaradi kratkega stika
2	Instalacija dodatnega varovala konektorja antene	1.5.2009	Jelka Novak HTC	Iskanje dobavitelja varovala, sklenitev pogodbe, prireditev proizvodne linije, da doda tudi varovalo	14.5.2009	Manufacturing	Medium	1.10	3G modul	1.1.1.1	Fizična poškodba kontaktov zaradi tresljajev
3	Modifikacija roke, ki montira tipko	6.5.2009	Jože Musič HTC	Brušenje nastavka in namestitev dodatnega zatiča	15.5.2009	Manufacturing	High	1.5.18	Gumb za vklop in izklop	1.1.1.2	Napačna montaža tipke
4	Namestitev dodatne absorbcijske prozorne folije med zaslon in ohišje telefona	1.5.2009	Marko Starina HTC	Nastavljanje stroja za razanje folije, modifikacija postopka sestavljanja naprave tako da se med zaslon in ohišje doda še folija	30.4.2009	Manufacturing	Medium	1.7.26	LCD	1.2.1.2	Mehanska poškodba zaslona
5	Instalacija dodatne kamere, ki pregleduje vzmeti ob vstopanju v 42rocess sestavljanja naprave	8.5.2009	Uroš Gorenc HTC	Nabava kamere, nastavitev zaznavanja defektnih vzmeti, testiranje, inštalacija v 42rocess sestavljanja	7.5.2009	Design		1.3.15	Vzmet za blokiranje zaslona	1.1.1.2	Vzmet je prekratka

Tabela 4.3: Sprejeti ukrepi

4.2.4 Prikaz zmanjšanja tveganja zaradi sprejetih ukrepov

Po sprejetih ukrepih smo še enkrat ocenili faktorje S, O in D. Pri tem smo ugotovili da se je tveganje zaradi ukrepov ustrezno zmanjšalo. Nove ocene tveganja najbolj kritičnih vzrokov prikazuje spodnja tabela.

Cause #	Cause	Occ (Init)	Occ (Rev)	Det (Init)	Det (Rev)	Cause RPN (Initial)	Cause RPN (Revised)	% Reduction in Cause RPN	Item #	Item Name
1.2.1.2	Mehanska poškodba zaslona	6	4	4	4	216	144	33,33	1.7.26	LCD
1.2.1.1	Previsoka temperatura baterije (zaradi kovinskih delcev v elektrolitu), posledično se posamezna celica z elektrolitom vname - zaradi kratkega stika	5	2	7	7	350	140	60	1.8	Baterija
1.1.1.2	Vzmet je prekratka	5	2	5	5	200	80	60	1.3.15	Vzmet za blokiranje zaslona
1.1.1.2	Napačna montaža tipke	6	1	4	4	216	36	83,33	1.5.18	Gumb za vklop in izklop
1.1.1.1	Fizična poškodba kontaktov zaradi tresljajev	8	1	4	4	224	28	87,5	1.10	3G modul

Tabela 4.4: Nove ocene tveganja

Ob izdelavi seminarske naloge smo uvideli nov način za izboljšanje produktov, pri katerih razvoju bomo morda nekoč sodelovali. FMEA analiza se je kljub njeni enostavnosti izkazala kot zelo učinkovita metoda za določanje, dokumentacijo in zmanjšanje tveganja za odpoved.

Za izdelavo FMEA analize v splošnem nebi potrebovali posebnega orodja (zadostuje že preprosta tabela), vendar smo ugotovili, da XFMEA orodje zelo olajša FMEA analizo in skrajša potreben čas za izvedbo. Nudi nam tudi učinkovit način za timsko izdelavo analize. Vsekakor orodje priporočamo vsakomur, ki se loti analize netrivialnega produkta.

5. Zaključek

Napravo smo najprej spoznali z vidika povprečnega uporabnika – pregledali smo zunanjost, kontrolne tipke, načine upravljanja ter vse njene funkcije. Ugotovili smo, da je dobro grajena, kompaktna in izredno zmogljiva kar se tiče funkcionalnosti saj združuje sposobnosti mobilnega telefona, dlančnika, GPS navigacijskega sistema, glasbenega predvajalnika in snemalnika ter fotoaparata oz. video-kamere, na drugi strani pa, da so največje slabosti velikost ter izredno omejeno trajanje baterije, ki bi lahko bila močnejša. Nato smo poiskali natančne specifikacije tako strojne, kot programske opreme ter ugotovili, da se napravo lahko v obeh vidikih primerja s kompleksnim računalniškim sistemom (kot je npr. PC). Razdelali smo vse strojne komponente (zaradi varnostnih razlogov nismo lastnoročno fizično razstavili naprave, temveč smo natančne podatke, sheme in slike poiskali na spletu) in ugotovili sposobnosti operacijskega sistema.

Ko so bili vsi zgoraj opisani podatki na razpolago, smo začeli zmogljivostno analizo naprave. Opravili smo tri tipe analize: statično analizo programske opreme, FTA analizo (Fault Tree Analysis) ter FMEA analizo (Failure Mode and Effect Analysis). Pri analizi programske opreme ter FMEA smo spoznali tudi različne metode analiziranja. Vsi pridobljeni rezultati so okvirne narave zato jih bralec ne sme jemati dobesedno. To je predvsem zaradi neizkošenosti pri uporabi zgoraj naštetih metod in zaradi velikih težav pri pridobivanju natačnih ter zanesljivih podatkov o napravi oz. vplivnih dejavnikih. Zaradi teh razlogov namen tega seminarja ni podajanje natančih rezultatov ter izračunov ampak predvsem prikaz kako se zgoraj omenjene analize opravlja, čemu koristijo – za kaj se jih uporablja, kakšne rezultate nam vračajo ter katere lastnosti naprave razkrivajo. Pod lastnosti naprave smatramo sposobnosti, zmogljivosti ter zanesljivosti naprave same in njenih komponent, tako strojnih kot programskih. Za natančnejše ugotovitve pri vsaki analizi naj si bralec ogleda posamezna poglavja v seminarju.

6. Literatura in viri

Android:

http://www.openhandsetalliance.com/index.html

http://developer.android.com/

http://source.android.com/

T-Mobile G1:

http://www.htc.com/www/

http://forum.xda-developers.com/showthread.php?t=440904

http://www.phonewreck.com/wiki/index.php?title=T-Mobile_G1

http://www.phonewreck.com/2008/12/09/t-mobile-g1-review-and-teardown/

http://mikechannon.net/PDF%20Manuals/HTC%20Dream%20SM%20(A04).pdf

Analiza programske opreme:

Hoang Pham: System Software Reliability

http://en.wikipedia.org/wiki/Cyclomatic_complexity

http://en.wikipedia.org/wiki/Halstead_complexity_measures

FTA:

http://en.wikipedia.org/wiki/Fault_tree_analysis

http://www.fault-tree.net/

http://www.fmeainfocentre.com/presentations/fta.pdf

FMEA:

http://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis

http://www.fmeainfocentre.com/

Uporabljena programska oprema:

CMTJava, analiza izvorne kode in izračun metrik, http://www.testwell.fi/

Edraw Max V4.6, načrtovanje FTA diagramov, http://www.edrawsoft.com/

XFMEA, FMEA analiza, http://xfmea.com/