

Návrh a vývoj testů webové aplikace

Bc. David Petura

Diplomová práce
2017



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David Petura**
Osobní číslo: **A13500**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Návrh a vývoj testů webové aplikace**

Téma anglicky: **The Design and Development of Web Application Tests**

Zásady pro vypracování:

1. Vytvořte rešerši na téma testování webových a mobilních aplikací.
2. Uveďte základní principy a charakteristiky testování.
3. Provedte analýzu významu testů při návrhu a implementaci webových aplikací.
4. Na vybraných aplikacích provedte sadu testů navržených v teoretické části.
5. Analyzujte realizovaný test a stanovte priority.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KOHL, Jonathan. Tap Into Mobile Application Testing [online]. [cit. 2016-11-17]. Dostupné z: <https://leanpub.com/testmobileapps>
2. PATTON, Ron. Testování softwaru. Vyd. 1. Praha: Computer Press, 2002. ISBN 80-722-6636-5. The Essential Guide to Mobile App Testing [online]. , 1-48 [cit. 2016-11-17]. Dostupné z: core.ecu.edu/STRG/materials/uTest_eBook_Mobile_Testing.pdf
3. MÜLLER, Thomas, Debra FRIEDENBERG, Armin BEER a Martin KLONK. Certified Tester: Foundation Level Syllabus [online]. 2011. , 1-78 [cit. 2016-11-17]. Dostupné z: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html>
4. PAGE, Alan, Ken JOHNSTON a Bj ROLLISON. Jak testuje software Microsoft. Vyd. 1. Brno: Computer Press, 2009, 384 s. ISBN 978-80-251-2869-5.
5. GOUSSET, Mickey. Řízení životního cyklu aplikací ve Visual Studiu 2010. Vyd. 1. Překlad Jan Pokorný. Brno: Zoner Press, 2010, 671 s. ISBN 978-80-7413-102-8.

Vedoucí diplomové práce: **doc. Ing. Jiří Vojtěšek, Ph.D.**

Ústav řízení procesů

Datum zadání diplomové práce: **3. února 2017**

Termín odevzdání diplomové práce: **16. května 2017**

Ve Zlíně dne 3. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen s příjím-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 15.5.2017


.....
podpis diplomanta

ABSTRAKT

Cílem diplomové práce je vybrání webové aplikace, která je vhodná pro testování jak na klasickém počítači tak i mobilních zařízeních. V teoretické části je shrnuto testování jako takové, strategie testování a použité testovací nástroje. Praktická část se zabývá návrhem testovacích scénářů, projití těchto scénářů na jednotlivých zařízeních a také konečným zhodnocením výsledků.

Klíčová slova: Mobilní aplikace, Testování, Android, iOS, Testovací scénář

ABSTRACT

The aim of this thesis is to choose right application, which is suited for testing on computer and also on mobile devices. The theoretical part consists of describing testing, testing strategies and used tools for testing. The practical part is concerned with creation of test cases and going through each one of them with different devices and also final evaluation of results.

Keywords: Mobile application, Testing, Android, iOS, Test case

Na tomto místě bych rád poděkoval konzultantovi své diplomové práce doc. Ing. Jiřímu Vojtěškovi, Ph.D. za pomoc a veškeré náměty.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 TESTOVÁNÍ	12
1.1 CO JE TESTOVÁNÍ	12
1.2 PROČ TESTOVAT	12
1.3 CHARAKTERISTIKY TESTOVÁNÍ	13
1.3.1 Pojem testování	13
1.3.2 Přístupy k testování software	13
1.3.3 Statické a dynamické testování	14
1.3.4 Úrovně testování	15
1.3.5 Typy testů	16
1.4 IDENTIFIKACE CHYB	18
1.4.1 Chyba typu: Program spadl a přestal fungovat	19
1.4.2 Logická chyba	19
1.4.3 Podivné chování aplikace	20
1.4.4 Ztráta dat	20
1.4.5 Použitelnost	20
1.5 CÍL TESTOVÁNÍ	20
2 VÝVOJ SOFTWARE VE SPOJENÍ S TESTOVÁNÍM	22
2.1 MODELÝ ŽIVOTNÍHO CYKLU VÝVOJE SOFTWARE	22
2.1.1 Shromažďování / analýza požadavků	22
2.1.2 Design	23
2.1.3 Kódování	23
2.1.4 Testování	23
2.1.5 Implementace a nasazení	23
2.2 TYPY ŽIVOTNÍHO CYKLU INFORMAČNÍHO SYSTÉMU	23
2.2.1 Model velkého třesku	24
2.2.2 Vodopádový model	24
2.2.3 Spirálový model	24
2.2.4 Agilní model	25
2.2.5 V-Model	25
3 TESTOVACÍ STRATEGIE	26
3.1 TESTOVACÍ PLÁN	26
3.2 TESTOVACÍ PŘÍPAD	26
3.3 TESTOVACÍ SCÉNÁŘ	26
3.4 TEST REPORT	27
3.5 NÁVRH TESTOVACÍCH PŘÍPADŮ	27
3.6 REALIZACE TESTŮ	27
3.7 VYHODNOCOVÁNÍ VÝSLEDKŮ	28
4 OPERAČNÍ SYSTÉMY MOBILNÍCH ZAŘÍZENÍ	29

4.1	ANDROID.....	29
4.2	VERZE ANDROIDU.....	29
4.3	IOS 30	
4.4	VERZE IOS.....	30
II	PRAKTICKÁ ČÁST	31
5	TESTOVÁNÍ WEBOVÝCH APLIKACÍ WALMART	32
5.1	POPIS APLIKACE	33
5.2	TESTOVACÍ PROSTŘEDÍ.....	33
5.2.1	Testovací prostředí Windows.....	33
5.2.2	Testovací prostředí MacOS.....	34
5.2.3	Testovací prostředí Android.....	34
5.2.4	Testovací prostředí iOS.....	35
5.3	PŘÍPRAVA A KONFIGURACE APLIKACÍ.....	36
5.4	ANALÝZA POŽADAVKŮ A STATISTIKA CHYBOVOSTI JEDNOTLIVÝCH KOMPONENT	36
6	TESTOVACÍ PLÁN.....	38
6.1	SEZNÁMENÍ S APLIKACÍ.....	38
6.2	ANALÝZA POŽADAVKŮ	38
6.3	OBLASTI, KTERÉ MAJÍ BÝT TESTOVÁNY	38
6.3.1	Úvodní strana	38
6.3.2	Stránky jednotlivých produktů	39
6.3.3	Okno po přidání do košíku.....	39
6.3.4	Košík	39
6.3.5	Objednávka	39
6.3.6	Historie objednávek	39
6.4	OBLASTI, KTERÉ NEMAJÍ BÝT TESTOVÁNY	39
6.4.1	Vyhledávání	39
6.4.2	Kategorie	39
6.4.3	Zbytek webové stránky	39
6.5	PŘÍSTUP A STRATEGIE.....	40
6.6	PASS A FAIL KRITÉRIA	40
6.7	VÝSTUPY Z TESTOVÁNÍ	40
7	MANUÁLNÍ TESTOVACÍ PŘÍPADY	41
7.1	ÚVODNÍ STRANA	41
7.2	ZÁHLAVÍ	42
7.3	ZÁPATÍ	43
7.4	STRÁNKA PRODUKTU	43
7.5	OKNO PO PŘIDÁNÍ DO KOŠÍKU.....	45
7.6	NÁKUPNÍ KOŠÍK.....	45
7.7	OBJEDNÁVKA	46
8	PROVEDENÍ NAVRHNUTÝCH TESTŮ	47

8.1	WINDOWS 7 – CHROME	47
8.2	WINDOWS 7 – IE11	49
8.3	WIN7 – FIREFOX.....	51
8.4	MACOS X – SAFARI.....	52
8.5	MACOS X – CHROME	53
8.6	SAMSUNG GALAXY S3 + ANDROID 4.1.2 + NATIVNÍ PROHLÍŽEČ	54
8.7	SAMSUNG GALAXY S5 + ANDROID 5.0 + NATIVNÍ PROHLÍŽEČ	56
8.8	SAMSUNG GALAXY S6 EDGE + ANDROID 6.0.1 + NATIVNÍ PROHLÍŽEČ.....	58
8.9	SAMSUNG GALAXY S8+ + ANDROID 7.0 + CHROME	59
8.10	SAMSUNG GALAXY TAB 4 10“ + ANDROID 5.0.2 + CHROME	61
8.11	APPLE IPHONE 5C + IOS 9.0 + SAFARI.....	62
8.12	APPLE IPHONE 6 + IOS 10.0.3 + SAFARI.....	64
8.13	APPLE IPAD MINI + IOS 9.3.2 + SAFARI.....	65
8.14	APPLE IPAD AIR + IOS 8.1.3 + SAFARI.....	66
9	ANALÝZA REALIZOVANÉHO TESTU	69
9.1	STANOVENÍ PRIORITY	70
	ZÁVĚR	71
	SEZNAM POUŽITÉ LITERATURY.....	72
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	74
	SEZNAM OBRÁZKŮ	75
	SEZNAM TABULEK.....	77
	SEZNAM PŘÍLOH.....	78

ÚVOD

Testeři mobilních aplikací a vývojáři se musí přizpůsobit stále měnícímu se vývoji mobilního prostředí a trhu. Existuje více jak 2 tisíce výrobců, 100 verzí operačních systémů a 600 mobilních operátorů. Výsledkem je obrovské množství odlišného prostředí pro testování. Nové technologie se objevují častěji, než kdykoli předtím a jsou okamžitě použity pro podnikání. Velikost displeje se pohybuje v rozmezí od jednoho a půl palce na chytrých hodinkách a více než patnáct palců na tabletech. Výpočetní výkon je stále nižší než má tradiční počítač a vývojáři musí vždy zohledňovat spotřebu energie. Přesto je standardním požadavkem bezproblémové chování aplikace bez zpoždění nebo zamrznutí na všech dostupných zařízeních.

Velké množství zařízení a vysoké nároky činí mobilní testování zajímavější a obtížnější. Chytré telefony nebo tablety navíc nejsou vhodné pro dlouhé psaní nebo pro některé náročné úkoly a z tohoto důvodu jsou mobilní aplikace pro vývojáře a specialisty UX mnohem náročnější. Nové verze mobilních operačních systémů se vyskytují častěji, než jsme zvyklí v případě osobních počítačů, takže i projektoví manažeři mají potíže ve své práci, protože musí sladit data vydání s aktualizacemi OS.

Čím dál častěji se dospívá k názoru, že kvalita aplikace, je právě to, co může společností pomoci k získání více zákazníků. Proto společnosti začínají zavádět oddělení zvané quality assurance, jejichž hlavním cílem je zaručení kvality produktu a s tím i udržitelnost na trhu.

Cílem této práce je v menším měřítku přiblížit, jak se takové oddělení quality assurance staví k problému testování na velkém množství zařízení, jaké postupy je potřeba dodržovat a v konečném důsledku jak vypadá výstup z testování.

I. TEORETICKÁ ČÁST

1 TESTOVÁNÍ

Webové aplikace a obecně softwarové programy jsou nyní běžnou součástí života, od běžných webových stránek až po aplikace ve zdravotnictví. Nicméně tyto aplikace jsou náchylné k chybovosti a většina lidí přišla do styku s aplikací, která nepracovala správně, nebo nefungovala vůbec. Od softwaru jako takového se očekává, že za všech okolností bude vždy pracovat správně a pokud ne, tak může dojít ke ztrátě zákazníka, tím pádem i peněz, času a reputace.[2]

Testováním se tyto nedostatky snažíme odstranit, nebo alespoň minimalizovat. Pro bližší seznámení s testováním jsou v dalších kapitolách popsány a rozebrány jednotlivé postupy, jak takové testování vypadá.[3]

1.1 Co je testování

Testování softwaru je proces spouštění programu nebo aplikace, s cílem najít chyby softwaru. Nicméně tato definice navozuje pocit, že testování je řízený proces, neboť musí být prováděno vždy s určitým záměrem. Jako první věc je naplánovat testy a to jak jejich návrh, tak poté i spouštění a vyhodnocení. [1]

Osobně bych testování nengeneralizoval jako řízený proces, ale popsal bych to spíše obecněji jako proces používaný k identifikaci správnosti, úplnosti a bezchybnosti vyvíjeného počítačového softwaru. Zahrnuje řadu činností prováděných s úmyslem najít chyby v softwaru tak, aby mohly být opraveny před tím, než je software propuštěn ke koncovým uživatelům. Jednoduše řečeno testování softwaru je kontrola, zda softwarový systém je bez vady.[1]

1.2 Proč testovat

Vývoj softwaru ve srovnání s výrobou jiných produktů je specifický tím, že každý systém je jiný a většinou je přizpůsoben konkrétnímu zákazníkovi nebo konkrétní cílové skupině, pokud chceme říci, že produkt je pro maloobchod, je potřeba také zohlednit specifika testování aplikací pro maloobchod. Kvůli jedinečnosti každého projektu a samozřejmě proto, že všichni jsou špatní - není možné dále zdokonalit proces vývoje do jeho ideální podoby a vytvořit dokonalý a bezchybný software. Testování je nezbytnou součástí každého projektu, bez něhož si nemůžeme být jisti kvalitou produktu, který se vyvíjí. [3]

I přes zřejmý význam testování a jeho dopad na kvalitu je to právě tato aktivita, kterou někteří dodavatelé softwaru podceňují. [3]

Zprvė uvedeme úlohu testování a řízení rizik. Pokud je testování provedeno nesprávně, pak jeho výstupy ve formě informací o produktu budou neplatné nebo nedostatečné, což může vést ke špatnému posouzení rizika a k následnému nesprávnému řízení projektu. Je pochopitelné, že s kritičností rozvoje softwaru je rostoucí potřeba minimalizovat rizika a možné, často velmi vysoké, ztráty. To se však dá dělat jen s obtížemi, pokud nebude kladen dostatečný důraz na procesy zajištění kvality a řízení kvality.[3]

1.3 Charakteristiky testování

Pro pochopení testování je potřeba uvést i charakteristiky a rozdělení. Na aplikaci se může i nemůže, pokud mluvíme o white-box a black-box testování, pohlížet jako na jeden velký celek, ale je potřeba si uvědomit, že aplikace se skládá z více funkčních i nefunkčních částí.[5]

1.3.1 Pojem testování

Vysvětlení pojmu testování softwaru je relativně složité. Neexistuje jedna jediná a všemi uznávaná definice (ostatně jako u mnoha jiných pojmů z oblasti informačních a komunikačních technologií, dále již jen ICT), význam je značně proměnlivý v závislosti na kontextu použití. O testování existuje rozšířená představa, že se stává pouze z provádění testů. To je sice částí testování, ale nejedná se o jedinou testovou aktivitu. Aktivity spojené s testováním existují před i po provádění testů. Pokud by testování softwaru bylo definováno výčtem aktivit, které jej představují, bylo by třeba výčet podstatně rozšířit. Ani pak by ale výsledek nebyl uspokojivý.[5]

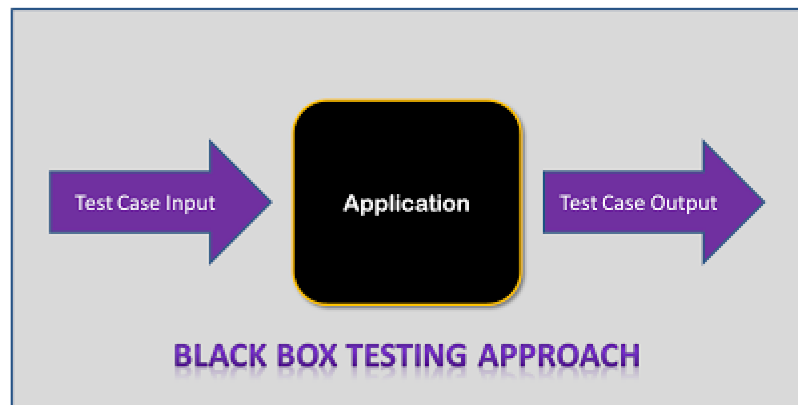
Jako testování je možno nazvat jakoukoliv aktivitu odhalující nesprávné chování programu, které je v rozporu se specifikací požadavků. Pohlížet na testování softwaru při jeho popisu pouze jako na konečný souhrn několika samostatných činností se jako příliš vhodné nejeví, protože nelze s jistotou říci, že výčet aktivit bude úplný.[5]

1.3.2 Přístupy k testování software

Tři základní přístupy k testování softwaru jsou, black-box, white-box a grey-box. Pojem grey-box je kombinací Black-box a White-box.[3]

Black-box

Black-box testování je technikou testování softwaru, ve které je zkoušena funkčnost testovaného softwaru bez pohledu na vnitřní strukturu kódu, details implementace a znalost interních cest softwaru. Tento typ testování je založen výhradně na softwarových požadavcích a specifikacích.[2]



Obr. 1 – Přístup k black-box testování [11]

White-box

White-box testování je testovací technikou, která zkoumá strukturu programu a odvozuje zkušební data z programové logiky a kódu. [9]

Grey-box

Grey-box testing (testování šedé skřínky) je kombinací obou předchozích přístupů. Tester má určitou znalost daného produktu a také toho z čeho je složen. Tester také má zkušenosti s danou problematikou a je schopen sám určit příčinu chyby, ale není nadále schopen přesněji definovat tento problém.[6]

1.3.3 Statické a dynamické testování

Při statickém testování se kód nevykonává, ale spíše se ručně kontroluje kód podle požadované dokumentace a návrhové dokumentace, aby se zjistily chyby, a proto název tedy "statický".[5]

Při dynamickém testování tester spouští kód ve smyslu, že daný software běží a dá se s ním pracovat. Kontroluje se funkční chování softwarového systému, využití paměti, CPU a celkový výkon systému, a proto název tedy "dynamický".[9]

1.3.4 Úrovně testování

Testovací úrovně jsou v podstatě identifikovány chybějícími oblastmi a zabraňují překrývání a opakování mezi vývojovými fázemi životního cyklu. V modelech životního cyklu vývoje softwaru jsou definovány fáze, jako je shromažďování a analýza požadavků, návrh, kódování nebo implementace, testování a nasazení. Každá fáze prochází testováním, proto existují různé úrovně testování. Různé úrovně testování jsou: [9]

Unit testy

Spouštění a vyhodnocování unit testů provádějí vývojáři, aby se ujistili, že jejich kód funguje správně a splňuje specifikace uživatele. Testují se jednotlivé třídy, funkce, rozhraní a postupy.[5]

Testy komponent

Nazýváno také jako testování modulů, jako komponentu si můžeme představit více tříd a funkcí zkombinovaných do jedné funkční komponenty. Například v aplikaci studentských záznamů existují dvě komponenty, které uchovávají záznamy studentů a další komponenta je načtení výsledků studentů. Obě komponenty jsou vyvinuty odděleně, a když jsou testovány jedna po druhé a ne dohromady, nazýváme to jako testování komponent.[5]

Integrační testování

Integrační testování se provádí, jakmile jsou dvě komponenty integrovány. Integrační testování testuje, jak se chovají dvě různé komponenty, které tvoří nějaký větší celek. Existuje několik typů integračních testů, jako jsou Big bang integration testing, Top down, Bottom up, Functional incremental.[3]

Systémové testování

Při testování systému ověřují testeři v zásadě kompatibilitu aplikace se systémem.[1]

Alfa-testování

Alfa Testování se provádí na straně výrobce softwaru a provádí se až na konci procesu výroby, za účelem zjištění posledních chyb, než software bude předán zákazníkovi k otestování.[10]

Beta testování

Beta testování je typ testování před komerčním uvolněním. Beta testování je poslední etapa testování a obvykle může zahrnovat zaslání produktu na zkušební testy k zákazníkovi anebo mimo společnost za účelem expozice v reálném světě. [1]

Akceptační testování

Testování aplikace z hlediska dodání finálního produktu zákazníkovi, který se na základě výsledků rozhodne, zda se převzetím produktu souhlasí.[10]

1.3.5 Typy testů

Testy softwaru jsou rozděleny do mnoha kategorií a typů. Samotnému dělení se věnuje mnoho článků a knih a každý autor se na problematiku dívá odlišně. Různé typy testování softwaru jsou prováděny s cílem dosáhnout různých cílů při testování softwarové aplikace. Můžete také číst o různých technikách testování softwaru, které mohou být spojeny s různými typy testování softwaru.[3]

Regresní testy

Regresní testování se provádí vždy po nahrání větších úprav do zdrojového kódu aplikace. Cílem regresních testů je nalézt závady, které byly zavedeny opravou chyb nebo zavedením nové funkce. Regresní testy jsou ideálními kandidáty pro automatizaci.[3]

Přírůstkové testy

Přírůstkové testy, nebo někdy označované také jako progresní, se zaměřují na testování nových funkcí nebo modulů, které byly k projektu přidány.[10]

Smoke testy

Jedná se o typ testování prováděného testery, aby se zjistilo, zda je nová verze poskytovaná vývojovým týmem dostatečně stabilní, tzn., že hlavní funkcionality funguje podle očekávání k provedení dalšího nebo podrobného testování. Smoke testování je určeno k nalezení defektů, které mohou zabránit testerům testovat aplikaci podrobněji.[10]

Load testy (zátěžové testy)

Testování zátěže je typ nefunkčního testování. Testování zátěže se provádí za účelem kontroly chování softwaru při normálním a špičkovém zatížení. Test zátěže se obvykle

provádí pomocí automatizovaných testovacích nástrojů. Zkouška zátěže má za cíl zjistit místa nebo problémy, které brání softwaru v tom, aby jeho výkony byly špičkové.[9]

Stress testy

Je typ testování výkonu, v němž je software vystaven špičkovým zatížením a dokonce i bodu zlomu, aby sledoval, jak se software bude chovat v bodě zlomu. Stresové testování také testuje chování softwaru s nedostatečnými zdroji, jako je CPU, paměť, šířka pásma sítě, prostor na disku atd. Stress testování umožňuje kontrolovat některé z atributů kvality, jako je robustnost a spolehlivost.[9]

Zotavovací testy

Zkouška obnovení se provádí za účelem zjištění, zda operace mohou pokračovat po katastrofě nebo po ztrátě integrity systému. Zahrnuje návrat do bodu, kdy byla známa celistvost systému, a poté přepracování transakcí až do okamžiku selhání.[9]

Bezpečnostní testy

Cílem testování zabezpečení je zajistit, aby software byl bezpečný vůči vnějším nebo vnitřním hrozbám od lidí a škodlivým programům. Bezpečnostní testování v podstatě kontroluje, jak dobrý je mechanismus autorizace softwaru, jak silná je autentizace, jak software udržuje důvěrnost údajů, jak software udržuje integritu dat, jaká je dostupnost softwaru v případě útoku. Testování vyžaduje dobrou znalost aplikace, technologie, síťování a bezpečnostních testovacích nástrojů. Se zvyšujícím se počtem webových aplikací se testování bezpečnosti stalo důležitější než kdy jindy.[3]

Výkonnostní testy

Testování softwaru a část výkonového inženýrství, která se provádí pro kontrolu některých kvalitativních atributů softwaru, jako je stabilita, spolehlivost a dostupnost. Na rozdíl od funkčních testů se testování výkonu provádí pro kontrolu nefunkčních požadavků. Testování výkonu kontroluje, jak dobře funguje software v očekávaném a maximálním pracovním zatížení. Existují různé varianty nebo sub typy výkonu, jako je zatěžování, zátěžové testy, testování objemu a testování konfigurace.[9]

Testy použitelnosti

Je typ testování softwaru, který se provádí, aby bylo jasné, jak je software vhodný pro uživatele. Cílem testování použitelnosti je umožnit koncovým uživatelům používat

software, sledovat jejich chování, emocionální reakci a shromáždit zpětnou vazbu o tom.[3]

Testování kompatibility

Testování kompatibility je jedním z testovacích typů prováděných testovacím týmem. Testování kompatibility kontroluje, zda je možné spustit software na jiném hardwaru, operačním systému, šířce pásma, databázích, webových serverech, aplikačních serverech, hardwarových perifériích, emulátorech, různých konfiguracích, procesorech, různých prohlížečích a různých verzích prohlížečů apod.[3]

Funkční testování

Funkční testování je formální typ testování prováděného testery. Funkční testování se zaměřuje na testování softwaru proti návrhovému dokumentu, případům použití a požadavkům dokumentu. Funkční testování je test typu černá skříňka a nevyžaduje vnitřní práci s tímto softwarem, na rozdíl od testování white-box.[4]

Srovnávací testy

Srovnávací test může tvořit jakýkoliv z výše uvedených testů, nebo jejich kombinace, která je opakovaně spuštěna za různých podmínek. Tento typ slouží například k porovnání výkonu dvou různých verzí aplikace, nebo stejné verze s různým nastavením. Je také vhodné porovnat různé konfigurace hardwaru, které máme k dispozici, popřípadě si tak otestovat změny v nastavení IIS nebo SQL serveru. Je také vhodné provádět srovnávací testy s konkurenčním softwarem.[4]

1.4 Identifikace chyb

Určit zda chyba je opravdu chybou, někdy nemusí být tak lehké jak se zdá, a proto je nejlepší dodržet přístup, ve kterém říkáme, že pokud se nám něco nezdá, tak to nahlásíme.[10]

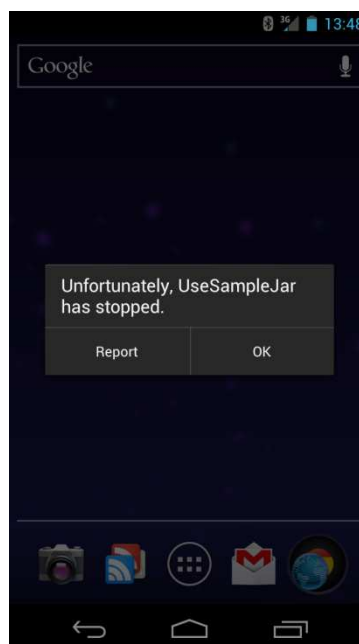
Hlášení všech chyb se může zdát jako zbytečné zatěžování procesu vývoje něčím, co ani nemusí být chyba, ale pokud se nad tím více zamyslíme, když to vadí jednomu, může to vadit i ostatním? Jak bylo mnohokrát řečeno, testování není pouze o hledání chyb, které něco dělají špatně, ale také chyb, které se týkají mimo jiné i použitelnosti. Každý tester by si měl ucelit, co která chyba znamená a jaký dopad může mít. Není nic horšího než

považovat kritickou chybu za chybu s nízkou prioritou jen kvůli tomu, že neznáme okolnosti.[10]

1.4.1 Chyba typu: Program spadl a přestal fungovat

Často také popisováno jako „crash“, viz obrázek 2. S tímto problémem se uživatelé setkávají poměrně často a může se projevit následujícími způsoby:[10]

- Program se prostě zavřel, bez jakékoliv notifikace
- Zobrazí se chybové hlášení, že program přestal pracovat
- Program se zasekne a nadále nereaguje na příkazy



Obr. 2 – Ukázky chyby typu
crash na zařízení Android[12]

Toto jsou jedny z nejkritičtějších chyb, které mají vždy nejvyšší prioritu. Důsledkem této chyby může uživatel přijít o data, která vyplňoval poměrně dlouhou dobu anebo se může stát, že se spustilo něco o čem uživatel ani nevěděl, jelikož neměl šanci to vidět.[10]

1.4.2 Logická chyba

Chyby, které jsou způsobeny špatnými výpočty, špatným použitím časového pásma anebo nefunkční službou pro polohu. Tyto chyby se nemusí odhalit na první pohled, ale je potřeba se nad nimi zamyslet, jestli se tyto informace zobrazují korektně. [10]

1.4.3 Podivné chování aplikace

Jedná se o chyby, které nejsou nijak problematické pro chod aplikace, ale mohou být otravné. Nejlépe se tyto chyby dají popsat na příkladech.[10]

- Aplikace se obnoví dvakrát místo jednou
- Zobrazí se nic neříkající chybová zpráva
- Aplikaci trvá příliš dlouho vykonat jednoduchý příkaz
- Psaní textu se zdá být pomalé

1.4.4 Ztráta dat

Častým problémem různých obchodních aplikací, jako jsou e-shopy, bývá také ztráta uživatelských dat. Nejčastějšími problémy je ztráta obsahu nákupního košíku, neuložení změn na účtu, přechod na další stránku smaže předchozí data a tak dále.[1]

1.4.5 Použitelnost

Chyby použitelnosti jsou nejhůře odhalitelné chyby, které existují. Pro tyto chyby je zapotřebí ohromné množství zkušeností anebo analytické programy, které dokážou s tímto pomoci. Použitelnost obecně nemá vliv na funkčnost aplikace, ale může mít vysoký vliv na spokojenost zákazníka a výdělečnost.[1]

1.5 Cíl testování

Testování softwaru má různé cíle. Hlavní cíle testování softwaru jsou následující: [3]

- Hledání defektů, které může programátor způsobit při vývoji softwaru.
- Získání důvěry a poskytování informací o úrovni kvality.
- Zabránit poruchám.
- Ujistění se, že konečný výsledek splňuje obchodní a uživatelské požadavky.

Testování softwaru pomáhá při finalizaci softwarové aplikace nebo produktu v souladu s požadavky obchodních a uživatelských požadavků. Je velmi důležité mít dobré zkušební pokrytí, abyste mohli úplně otestovat softwarovou aplikaci a ujistit se, že funguje dobře a podle specifikací.[2]

Jakmile je produkt doručen koncovým uživatelům nebo zákazníkům, měli by být schopni jej provozovat bez jakýchkoli stížností. Aby se tak stalo, měli by testeři vědět, jakým způsobem budou zákazníci používat tento produkt, a proto by si měli zapsat testovací

scénáře a navrhnout testovací případy. To hodně pomůže při plnění všech požadavků zákazníka.[2]

Cílem testování je také komplexní pokrytí aplikace testy a to znamená, aby testování bylo provedeno tak, že pokryje různé oblasti, jako je například funkčnost aplikace, kompatibilita aplikace s operačním systémem, hardware a různé typy prohlížečů, testování výkonu pro testování výkonu aplikace a testování zatížení.[2]

2 VÝVOJ SOFTWARE VE SPOJENÍ S TESTOVÁNÍM

Proces vývoje softwaru začíná na základě požadavků a vzhledem k tomu, že projekt se neustále vyvíjí, je vynaloženo větší úsilí na vytvoření či úpravu řešení, do něhož se zapojuje více lidí, a tím se zvyšují náklady na projekt. Chyby zjištěné na konci vývojového procesu mají tendenci vyžadovat značné úsilí, čím dříve je chyba identifikována, tím bude levnější problém vyřešit. Náklady na opravu chyby mohou představovat něco kolem logaritmické funkce, kde se náklady mohou zvýšit více než desetkrát, jak projekt postupuje ve fázích vývojového cyklu.[2]

Například chyba identifikovaná během konceptu stojí něco kolem nuly, ale když se tato chyba objeví až po implementaci nebo testování, průměrná cena opravy se může dostat na něco mezi 10 a 1000 krát více než v předchozím kroku. Když zákazníci naleznou tuto chybu ve výrobním prostředí, náklady na problém se zabývají všemi vedlejšími účinky, které se jí týkají a to je místo, kde se věci mohou dostat vážných problému.[6]

2.1 Modely životního cyklu vývoje softwaru

Životní cyklus vývoje softwaru je proces, který zajišťuje vybudování dobrého softwaru.

Každá fáze životního cyklu má svůj vlastní proces a výstupy, které se přivádějí do další fáze. Obvykle se jedná o 5 fází, které začínají analýzou a shromažďováním požadavků a ukončením implementace. Podívejme se podrobněji na každou fázi.[6]

2.1.1 Shromažďování / analýza požadavků

Tato fáze je kritická pro úspěch projektu. Očekávání (ať už jde o klienta nebo o váš tým) je třeba podrobněji zdokumentovat. Jedná se o iterativní proces s velkou komunikací mezi zúčastněnými stranami, koncovými uživateli a projektovým týmem. Pro shromažďování požadavků lze použít následující techniky:[2]

- Identifikujte a zachyťte požadavky zúčastněných stran pomocí dotazníků a průzkumů zákazníků.
- Vytvoření více případů použití popisuje každou akci, kterou uživatel provede v novém systému.
- Prototypy mohou být postaveny tak, aby ukázaly klientovi, jak bude vypadat konečný produkt.

V této části, už se může pomalu začít s návrhy testů dle požadavků, abychom dostali nějakou počáteční kostru testů.[2]

2.1.2 Design

Požadavky na technické navrhování jsou v této fázi připraveny vedoucími vývojovými pracovníky, které mohou zahrnovat architekty a vedoucí vývojáře. Požadavky se používají k definování způsobu, jak bude aplikace napsána. Technické požadavky budou obsahovat podrobné databázové tabulky, které budou přidány, nové transakce budou definovány, bezpečnostní procesy a hardwarové a systémové požadavky.[2]

2.1.3 Kódování

Tato fáze je skutečné kódování programu a unit testování vývojovým týmem. Po každé fázi může vývojář prokázat práci vykonanou byznys analytikovi a další vylepšení mohou být vyžadovány. V této fázi je důležité, aby vývojáři byli otevřeně orientovaní a flexibilní, pokud budou zavedeny jakékoliv změny. Toto je obvykle nejdelší fáze vývojového životního cyklu softwaru. Hotový produkt je vstup do testovací fáze.[2]

2.1.4 Testování

Po přenesení aplikace do testovacího prostředí budou provedeny různé typy testů včetně integrace a testování systému. Uživatelské testování je poslední částí testování a je prováděno koncovým uživatelem, aby systém zajistil jejich očekávání. V tomto okamžiku mohou být nalezeny vady a při analýze, návrhu nebo kódování může být zapotřebí více práce. Po odsouhlasení všech zúčastněných stran, může začít implementace a nasazení.[2]

2.1.5 Implementace a nasazení

Velikost projektu určí složitost nasazení. Může být vyžadováno školení pro koncové uživatele, operátory a pracovníky IT pohotovosti. Nasazení systému může být prováděno v etapách, které začínají jednou větví, pak se pomalu přidávají všechny větve.[2]

2.2 Typy životního cyklu informačního systému

V praxi se používá mnoho typů životního cyklu informačního systému (SDLC). Vždy záleží na daném produktu, který je vyráběn. Výběr konkrétního typu SDLC je kritickou částí rozhodování a to zejména proto, že ne každý SDLC je vhodný pro daný produkt.[2]

2.2.1 Model velkého třesku

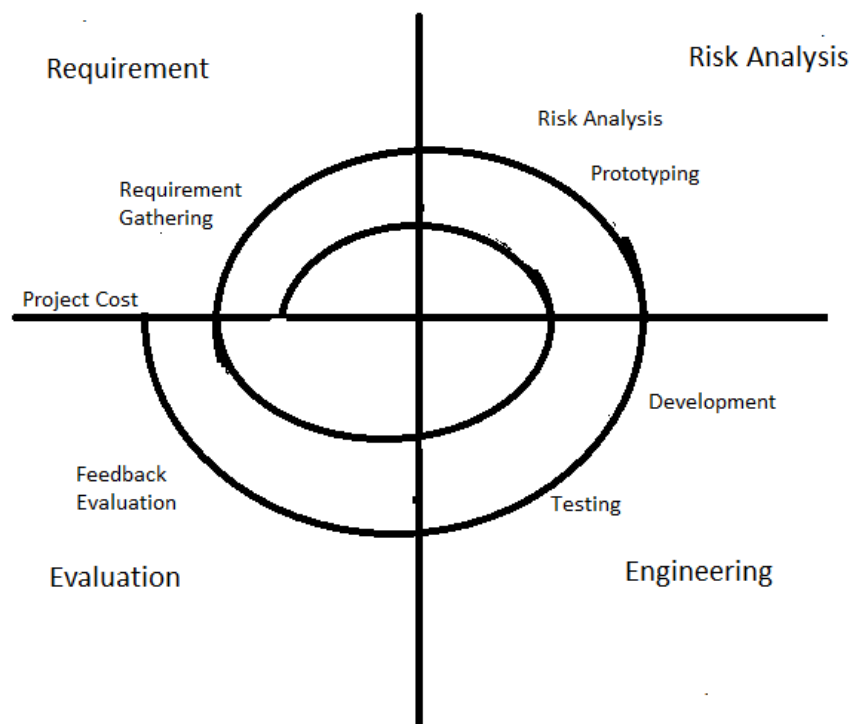
Model velkého třesku je jedním z modelů, který se neřídí žádnými pravidly. Jako vstup se využívá množství investovaných peněz, podle toho se pak určí velikost vývojového týmu. Výstupem je poté hotový produkt, který ve většině případů neodpovídá požadavkům klienta. Tento model se používá nejvíce, pokud nejsou předem jasně dané požadavky a specifikace.[2]

2.2.2 Vodopádový model

Označován také jako lineární-sekvenční model životního cyklu. Principem tohoto modelu je to, že jakmile jedna fáze skončí, tak ihned začíná další, bez jakéhokoliv překrývání a bez možnosti se vrátit do předchozí fáze.[2]

2.2.3 Spirálový model

Spirálový model se vyznačuje tím, že jednotlivé fáze cyklu se neustále opakují, dokud není produkt hotový, viz obrázek 3. Tento model je zejména vhodný pro velké projekty, kde je potřeba něco neustále vylepšovat a upravovat. Každé dodatečné úpravy se provádějí vždy v nové iteraci a nikdy ne v aktuální. V tomto modelu je kladen velký důraz na analýzu a minimalizaci rizik. [2]



Obr. 3 – Spirálový model SDLC [13]

2.2.4 Agilní model

Agilní model je kombinací iterativních a přírůstkových procesních modelů, kde nejdůležitějším předpokladem je reakce na změnu požadavků v průběhu vývoje. Agilní model probíhá v iteracích a každá iterace může trvat jeden až tři týdny. Každá iterace zahrnuje různé funkční týmy pracující současně v různých oblastech:[2]

- Plánování
- Analýza požadavků
- Design
- Kódování
- Unit testování
- Akceptační testování

2.2.5 V-Model

Model V je založen na modelu vodopádu, ale s tím rozdílem, že každá fáze tohoto modelu má k tomu také odpovídající fázi testování.[2]

To znamená, že pro každou jednotlivou fázi ve vývojovém cyklu existuje přímo spojená testovací fáze, která zajišťuje určitou míru kvality produktu před přechodem do další fáze.[1]

3 TESTOVACÍ STRATEGIE

Výběr testovací strategie je jeden z nejzákladnějších faktorů pro úspěšné testování a to jak pro usnadnění vynaloženého úsilí, správných testovacích plánů tak i také přesných odhadů.[6]

3.1 Testovací plán

Plán testování softwaru je dokument popisující rozsah a aktivity testování. Je základem pro formální testování jakéhokoliv softwaru nebo produktu v daném projektu. Testovací plán je první a základní věcí, kterou by měl každý testovací tým mít. Je to jakýsi základ pro každého člena testovacího týmu, ve kterém jsou obsaženy všechny odpovědi na otázky související s testováním daného produktu. Plán testování by měl minimálně obsahovat:[1]

- Cíle a definice očekávání provedených testů
- Odkazy na dokumentaci
- Seznam jednotlivých částí daného softwaru a jejich další rozdělení
- Rizika, která se softwarem souvisí
- Kategorie testů
- Kritéria vyhodnocení testu
- Seznam testovacích případů
- Seznam zkratk a vysvětlivek
- Definice dat, která jsou potřebná pro provádění testů

3.2 Testovací případ

Testovací případ je dokument, který obsahuje sadu zkušebních dat, předpokladů, očekávaných výsledků a dodatečných podmínek, které byly vytvořeny pro konkrétní zkušební scénář, aby bylo možné ověřit, zda jsou splněny konkrétní požadavky.[5]

Zkušební případ funguje jako výchozí bod pro provedení testu a po uplatnění sady vstupních hodnot aplikace má konečný výsledek a opouští systém v nějakém koncovém bodě nebo také známý jako post-podmínka provedení.[5]

3.3 Testovací scénář

Testovací scénář je seskupení testovacích případů podle nějakého kritéria, které rozdělují daný software to logických částí. Toto seskupení dokáže libovolného uživatele provést

danou částí softwaru a porozumět souvislostem v dané části. Ideální scénář je věrohodný, motivující, snadno ověřitelný a také komplexní.[2]

3.4 Test report

Testovací hlášení se skládá ze shromáždování údajů pro informování zúčastněných stran (podnikatelů, přímých uživatelů) o zjištěném stavu dokončení apod. Zjednodušeně řečeno lze říci, že jde o hmatatelný důkaz o tom, zda byly všechny požadavky implementovány do testované aplikace.[2]

Dle svých zkušeností se používají pro testovací hlášení a sledování pokroku nástroje pro správu testů (Zephyr, TestLink, Centrum kvality HP a jiné).[8]

Ve zprávě může být uvedeno míra pokrytí testů, míru selhání testů, celkové závady s příslušnými stavy a kritické nedostatky.[8]

3.5 Návrh testovacích případů

Navržení dobrého testovacího případu, je vždy základním bodem pro to, aby daný případ byl srozumitelný a jednoduchý pro vykonání a proto je dobré dodržovat pár pravidel: [7]

- Zisk informací - Pro získání různých druhů informací jsou vhodné různé typy testovacích případů.
- Testovací případ je tak dobrý, jak je dobrá jeho nejslabší část.
- Testovací případ by měl vždy vycházet z určitých vzorů. Například nemůžeme používat vzory pro testování použitelnosti tak jako vzory pro testování bezpečnosti.

Výše uvedené přístupy pro vytvoření testovacího scénáře jsou pouze orientační a nemohou být kontrolovány. Je třeba poznamenat, že každý scénář vyžaduje individuální přístup založený na aplikaci nebo softwaru.[7]

Testovací případ by měl být navržen s maximálním ohledem na dobrou opakovatelnost a sledovatelnost každého testu, škálovatelnost testu a snadnou organizaci testu.[5]

3.6 Realizace testů

Spouštění nebo vykonávání testů podle testovacího plánu může probíhat automaticky anebo manuálně. V závislosti na množství testovacích případů je toto asi nejnáročnější část testovací strategie a většinou se na ní podílí celé oddělení testerů. [7]

3.7 Vyhodnocování výsledků

Výsledky testů jsou neméně důležitou součástí testování, protože méně přehledné výsledky jsou nic neříkající a zavádějící. Pro to aby výsledky byly smysluplné, musí být jasně definované návrhy testů, jejich jednotlivé kroky a následné očekávané výstupy. Jinými slovy, při sledování konkrétního testovacího případu a jeho výsledku udržujete celkový pohled na řešení jako celek a neustále si uvědomujete primární cíl práce.[4]

Pokud například výsledek testu nemá smysl, musíte se vrátit k návrhu testu. Zeptejte se na základní otázky, zkontrolujte celý test a pokuste se najít alternativy k němu, protože špatný výsledek může také znamenat špatný test.[5]

4 OPERAČNÍ SYSTÉMY MOBILNÍCH ZAŘÍZENÍ

Operačních systémů existuje mnoho, ale nás bude zajímat zejména Android od firmy Google a iOS od firmy Apple. Jedná se o jedny z nejpoužívanějších operačních systémů na mobilních zařízeních.[7]

4.1 Android

Tento systém se vyskytuje především v chytrých telefonech předních výrobců, ale nyní také v dalších mobilních zařízeních jako jsou tablety či notebooky. V současnosti se jedná o nejvíce rozšířený operační systém s největším tržním podílem. Android je open-source software a je dostupný pod licencí GPLv2.[7]



Obr. 4 – Logo android [13]

4.2 Verze Androidu

První verze spatřila světlo světa v roce 2008 a to ve verzi 1.0 a následovaly další verze:[1]

Tabulka 1 – Verze Systému Android

Verze Android	Název verze
1.5	Cupcake
1.6	Donut
2.0/2.1	Eclair
2.2	Froyo
2.3/2.4	Gingerbread
3.0/3.1/3.2	Honeycomb

4.0-4.0.4	IceCreamSandwich
4.1/4.2/4.3	JellyBean
4.4	KitKat
5.0/5.1	Lollipop
6.0	Marhmallow
7.0/7.1	Nougat

4.3 iOS

Dalším operačním systémem je atraktivní systém iOS (dříve nazývaný iPhone OS) vyvíjený firmou Apple Inc. Původně určený systém výhradně pro revoluční smartphone iPhone, který se začal prodávat v roce 2007, se postupně rozšiřuje do dalších zařízení. Objevil se v hudebním přehrávači iPod Touch a nyní také v úspěšném tabletu iPad. V současné době ovšem co se týče chytrých telefonů iOS zaostává za svým největším konkurentem, jímž je výše zmíněný operační systém Android. Jako hlavní nevýhody zařízeních běžících na operačním systému iOS jsou izolovanost, používání nestandardních konektorů, menší informační hustota, slabší integrace a podpora Google služeb atd.[1]

4.4 Verze iOS

Stejně tak jak Android, tak i iOS se verze začaly číslovat od 1.0. Původní verze se nazývali iPhone OS a až od verze 4.0 se název OS změnil na iOS 4.0 a v tomto názvosloví pokračuje až dodnes čili do poloviny roku 2017, kdy aktuální verze iOS je 10.X.[7]

II. PRAKTICKÁ ČÁST

5 TESTOVÁNÍ WEBOVÝCH APLIKACÍ WALMART

Při rozhodování jakou webovou aplikaci si zvolit k otestování hraje roli mnoho faktorů. Jak již bylo zmíněno dříve, tak k důslednému otestování konkrétního softwaru je potřeba také jeho znalost. Pro mě osobně v tuto dobu je nejvhodnější si zvolit webové aplikace Walmart. Aplikace je vhodná v tom, že plně podporuje zobrazení na mobilních zařízeních a mám jistotu, že aplikace je plně funkční.

Hlavním důvodem výběru této aplikace je má zkušenost a znalost této aplikace. Toto je základním bodem pro to, aby na danou aplikaci mohli být navrženy testy a testovací plán. V případě výběru jiné známější aplikace pro širokou veřejnost žijící v Česku by bylo nutné si tuto aplikaci nejdříve důkladně nastudovat a vyzkoušet, nicméně toto není otázka jednoho nebo dvou měsíců, ale otázka let a ani v tomto případě by testy neodpovídali tomu, čemu mají, protože bez znalosti vnitřního kódu by tyto testy ztrácely na důvěryhodnosti a navíc by se mohl stát, že by vybraná aplikace, nebo její větší část, nebyla vhodná pro testování na konkrétních zařízeních.

První verzi aplikace je verze pro stolní počítače. Uživatel má možnost využívat verzi pro stolní počítače, pokud je jeho rozlišení stejné nebo větší než 1024 pixelů na šířku. Walmart v tomto případě zobrazuje záhlaví a zápatí v módu, kdy veškerá navigace na stránce je prováděna právě v těchto sekcích. Pro testování této verze byly zvoleny operační systémy Windows 7 a macOS X.

Druhou testovanou aplikací je zobrazení aplikace Walmart v rozlišení menším než 1024 pixelů, neboli v zobrazení pro mobilní zařízení. Stránka v tomto zobrazení ztrácí větší část záhlaví, zůstává zobrazeno pouze pole pro vyhledávání, logo Walmart, tlačítko pro přesměrování do nákupního košíku a tlačítko pro otevření levého menu, ve kterém je poté obsaženo vše, co obsahuje klasické záhlaví v případě verze pro klasické počítače. Testování bylo prováděno na mobilních zařízeních značky Samsung a Apple.

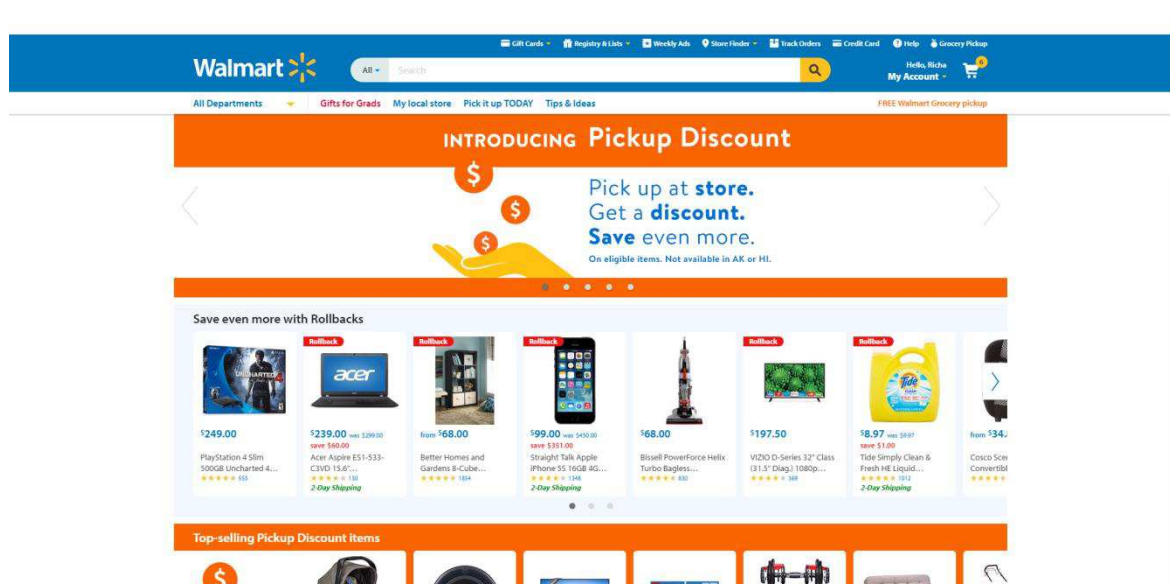
Práce s těmito verzemi je naprosto jiná a při vytváření testovacích případů byl brán zřetel na toto rozdílné zobrazení, tak aby testovací případy odpovídaly oběma verzím. Pokud bych měl například testovací případ na test funkcionality pro stolní počítače, kdy uživatel najede kurzorem myši na nějaký obrázek, tak se mu zobrazí náhled tohoto obrázku, ale ve skutečnosti tento obrázek nebyl vybrán, což pochopitelně na mobilním zařízení nejsme schopni udělat. Testy jsou tedy navrženy obecně, tak abychom se těmito nesrovnalostem vyhnuli.

5.1 Popis aplikace

Obecně se dá říci, že Walmart je Retail a E-tail aplikací, neboli jako více známý název e-shop. Walmart je vhodný pro jakéhokoliv zákazníka, který hledá cokoliv, co takový e-shop může nabídnout.

Vybraný typ aplikace je natolik komplexní, že se v reálném čase a rozsahu diplomové práce nedá zcela pokrýt, a proto na základě mých zkušeností a statistik byly vybrány nejdůležitější body, které je potřeba splnit pro akceptaci z minimálního hlediska.

Aplikace je postavena na mnoha technologiích, jako je Javascript, Electrode, HTML, CSS, HTML5 a podobně, proto se webová stránka nedá rozdělit přesně podle použitých technologií, ale bude rozdělena na jednotlivé sekce.



Obr. 5 – Úvodní stránka testované aplikace Walmart

5.2 Testovací prostředí

Pro testovací prostředí aplikace Walmart bude použit jak operační systém Windows a MacOS, tak i mobilní zařízení s operačním systémem Android a iOS.

5.2.1 Testovací prostředí Windows

Operační systém Windows bude standardní instalace verze Windows 7, protože je to momentálně jediný systém Windows, který mám k dispozici, s kombinací prohlížečů Internet Explorer 11 a Chrome.

5.2.2 Testovací prostředí MacOS

Operační systém MacOS bude standardní instalace verze X s kombinací prohlížečů Safari a Chrome.

5.2.3 Testovací prostředí Android

V důsledku zmíněném v předchozích kapitolách je potřeba pokrýt testy aplikace různými kombinacemi verzí Androidu a rozlišením telefonů.

Pro testování byly vybrány následující kombinace:

- Samsung Galaxy S3 + Android 4.1.2
- Samsung Galaxy S5 + Android 5.0
- Samsung Galaxy S6 Edge + Android 6.0.1
- Samsung Galaxy S8+ + Android 7.0, viz obr. 6
- Samsung GalaxyTab 4 10“ + Android 5.0.2
- Tovární prohlížeč systému Android + Chrome pro Android



Obr. 6 – Systémové informace telefonu
Galaxy S8+

5.2.4 Testovací prostředí iOS

Stejně tak jako prostředí Android, tak i pro iOS byly vybrány následující kombinace:

- Apple iPhone 5C + iOS 9.0, viz obr. 7
- Apple iPhone 6 + iOS 10.0.3
- Apple iPad mini + iOS 9.3.2
- Apple iPad Air + iOS 8.1.3
- Pouze prohlížeč Safari



Obr. 7 – Systémové informace telefonu iPhone 5C

5.3 Příprava a konfigurace aplikací

O přípravu a konfiguraci testovacích aplikací se stará implementátor na základě požadavků od testera, nicméně tester by měl být schopen si testovací aplikaci připravit i sám, jelikož i v těchto konfiguracích už mohou být chyby.

Příprava aplikace u produktu Walmart spočívá v tom, že se vždy testuje aktuální vývojová verze. Je potřeba zajistit aktualizaci všech souborů vzhledem k provedeným změnám ve zdrojovém kódu.

Nicméně pro účely diplomové práce byla použita produkční verze, která je veřejně dostupná všem uživatelům a to z důvodu dohody o mlčenlivosti.

5.4 Analýza požadavků a statistika chybovosti jednotlivých komponent

Největším problémem u komplexních aplikací je mít nějaký výchozí bod, ze kterého začít a podle kterého začít chystat testy. Jedním ze způsobů, jak zjistit, které části je potřeba co nejdříve pokrýt testy, je přehled požadavků. U aplikací, které běží už delší dobu, by měl být zaveden systém pro správu chyb, neboli požadavků.

Pro potřeby Walmart bylo potřeba zanalyzovat požadavky za uplynulý rok a vytvořit statistiku chybovosti jednotlivých oblastí a k nim přiřazených klíčových slov.

V systému bylo zaznamenáno více než 2500 požadavků za poslední rok, nicméně tyto požadavky bylo dále potřeba filtrovat podle potřeb. Použitý filtr vypadal následovně

- Pouze chyby typu: P0
- Pouze chyby typu: P1

Chyba typu P0 je nejzávažnější chybou, která brání vydání verze do produkčního prostředí, protože brání uživateli v používání aplikace, může se jednat o chybu typu, kdy nějaké tlačítko vůbec nereaguje a tím brání uživateli se dostat na požadovanou stránku. Typ P1 je také velice závažnou chybou, ale svou povahou nebrání uživateli v používání aplikace, jako takovou chybu si lze představit například nenačtení obrázků produktů.

Výše použitý filtr zúžil celkové množství vstupních požadavků z 2500 na číslo 703. Tyto vyfiltrované požadavky poté byly rozděleny do kategorií podle oblastí a bylo zjištěno, kterým oblastem je potřeba se nejvíce věnovat.

Pro porozumění následující statistiky je potřeba uvést pár základních věcí. Aplikace se skládá z oblastí a konkrétní oblasti mají přidružené komponenty (v případě správy požadavků se jim říká klíčová slova). Pro příklad jako jedna oblast může být stránka konkrétního produktu a na této stránce jsou další přidružené komponenty, jako jsou například obrázky produktu, část pro výběr množství, barev a cena produktu.

6 TESTOVACÍ PLÁN

Jak již bylo zmíněno v kapitole 5, testovací plán je základním bodem pro začátek testování. Je v něm popsáno, co bude testováno, jak to bude testováno, jaká budou použita zařízení a jaké budou výstupy.

6.1 Seznámení s aplikací

První nejdůležitější věcí je vůbec vědět, co testujeme a jak to vypadá. Proto je vždy potřeba daný produkt důkladně znát.

Tato část je jednou z nejtěžších částí, protože zabere nejvíce času. Nelze se seznámit s projektem během jednoho týdne, ale zabere to minimálně rok, než tester je schopen sestavit důkladný testovací plán.

Seznámení s aplikací probíhalo analýzou vstupních požadavků a z nich určení kritických oblastí a mimo jiné hlavně dlouhodobou znalostí konkrétní aplikace.

6.2 Analýza požadavků

Projít všechny požadavky za dané časové období zmíněné v předchozí kapitole a specifikovat nejkritičtější oblasti, které je potřeba pokrýt jako první.

Výsledkem je znalost systému a obecný přehled o kritických oblastech.

6.3 Oblasti, které mají být testovány

6.3.1 Úvodní strana

Úvodní strana obecně obsahuje nejdůležitější informace a má za účel informovat uživatele o možnostech kompletní stránky.

Záhlaví

Část v horní části obrazovky, která slouží uživateli pro navigaci na stránce.

Zápatí

Část v dolní části obrazovky, která slouží uživateli pro navigaci na stránce.

6.3.2 Stránky jednotlivých produktů

Je to nejdůležitější část stránky, která je e-shop. Na těchto stránkách uživatele tráví nejvíce času a je potřeba, aby fungovaly tak jak mají. Stránky jsou většinou komplexní a obsahují všechny informace o daném produktu.

6.3.3 Okno po přidání do košíku

Tato část se zobrazí uživateli, jakmile přidá předmět do košíku. V této části je možné ještě udělat drobné změny a zkontrolovat si správnost daného produktu.

6.3.4 Košík

Nákupní košík je část, kde uživatel vidí předměty přidané do košíku a může tyto předměty nadále upravovat anebo mazat.

6.3.5 Objednávka

Část, kde uživatel prochází jednotlivé body, které je nutné vyplnit k dokončení objednávky.

6.3.6 Historie objednávek

Po úspěšném dokončení objednávky uživatel může na této stránce vidět konečný rozpis objednaných věcí a popřípadě objednávku zrušit.

6.4 Oblasti, které nemají být testovány

6.4.1 Vyhledávání

Nízké riziko výskytu chyby, proto není potřeba testovat a neomezuje uživatele v objednávce.

6.4.2 Kategorie

Stránky, které sdružují předměty podle kategorie a jiných kritérií. Obecně uživatel tyto stránky moc nenavštěvuje a pokud ano, tak je tato strana bez chyb.

6.4.3 Zbytek webové stránky

Ostatní sekce jsou využívány pouze okrajově v porovnání s výše uvedenými a není jim potřeba věnovat pozornost.

6.5 Přístup a strategie

- Není potřeba žádných speciálních nástrojů
- Není potřeba mít speciální zkušenosti
- Žádné metrické statistiky nebudou vedeny
- Konfigurace je na straně klienta a bude testováno produkční prostředí
- Pouze jedno prostředí - produkce
- Hardware
 - 2x Stolní počítač
 - 9x mobilní zařízení
- Software
 - Windows 7 nebo 8 a vyšší
 - MacOS X
 - iOS 8 / 9 / 10
 - Android 4 / 5 / 6 / 7
- Regresní testování bude prováděno na každém zařízení
- Regresní testování bude prováděno pouze jednou

6.6 Pass a Fail kritéria

V tomto případě nebudou použity žádné kritéria, protože se jedná o produkční prostředí, které je dostupné uživatelům.

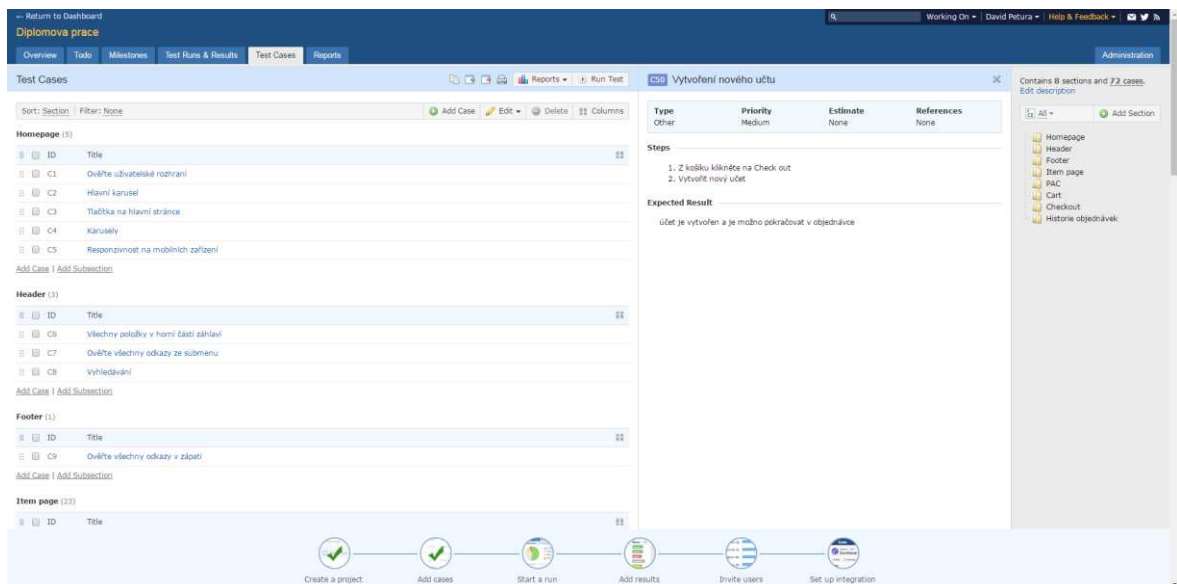
Jedná se pouze o regresní testování, které dokáže bezchybnost aplikace.

6.7 Výstupy z testování

- Dokument s testovacím plánem
- Testovací případy
- Error logy
- Reportování problémů

7 MANUÁLNÍ TESTOVACÍ PŘÍPADY

Pro tvorbu manuálních testovacích případů je použita webová aplikace TestRail viz obr. 8, která je volně dostupná po zaregistrování.



Obr. 8 – Ukázka aplikace TestRail s vytvořenými testovacími případy

Základním bodem pro testovací plán je analýza a rozhodnutí o nejkritičtějších oblastech. Z toho vyplynulo, že je potřeba zpracovat testy na pokrytí veškeré části týkající se úvodní stránky, stránky produktu, nákupního košíku a objednávky.

Obecně na walmart.com je nejnavštěvovanější stránkou stránka produktu a proto se jí v testovacích případech věnuje nejvíce pozornosti.

Na každé této části se může vyskytovat mnoho vlastností, komponent a funkcí. Abychom se v budoucnu vyhnuli dodatečnému dodělávání, tak jsme se rozhodli, že z dokumentace vezmeme vše, co dané části podporují, a navrheme testovací případy.

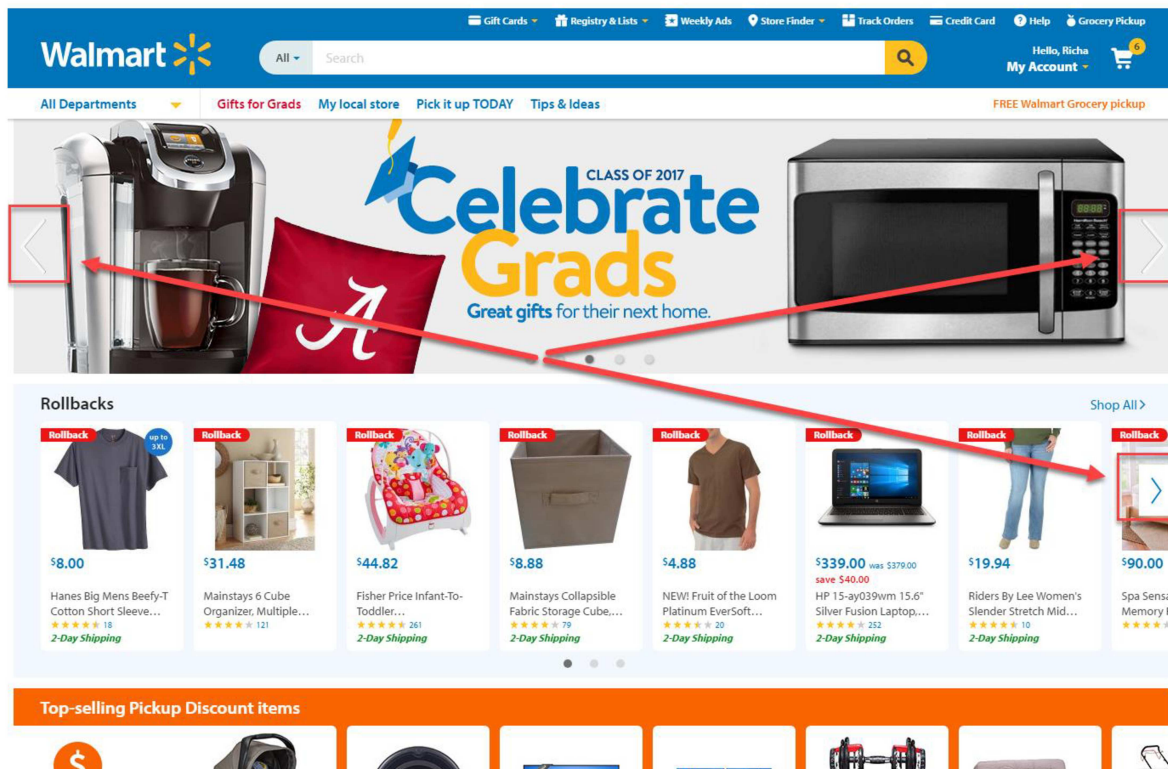
7.1 Úvodní strana

Úvodní strana je stránka, na kterou je uživatel přesměrován jako první a proto jí musí být věnována pozornost.

Na této stránce se testuje hlavně grafické rozhraní, protože funkcionality je velice omezená.

Jediným funkčním prvkem na této stránce jsou karusely, viz obr. 9, pro posun mezi jednotlivými položkami doleva a doprava.

Také je potřeba otestovat veškeré tlačítka, které jsou dostupné.



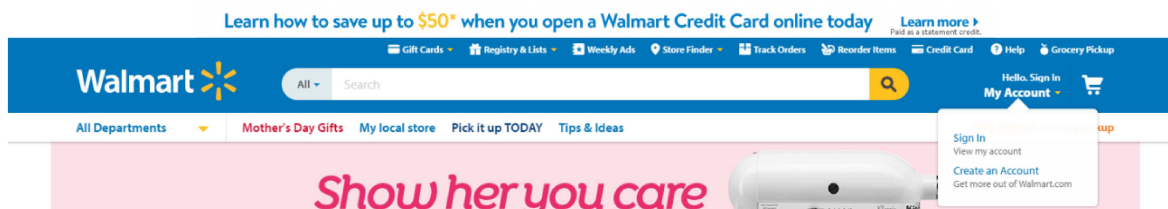
Obr. 9 – Karusely pro posun mezi seznamem produktů

7.2 Záhloví

Záhloví je součástí každé webové stránky. V záhlaví se vyskytují nejčastěji odkazy na další stránky a také obvykle pole pro vyhledávání, viz obrázek 10.

V této části je nutné otestovat zejména funkčnost všech odkazů a také editovatelnosti vyhledávacího pole.

Mimo jiné je zde část pro uživatele indikující aktuální stav přihlášení.

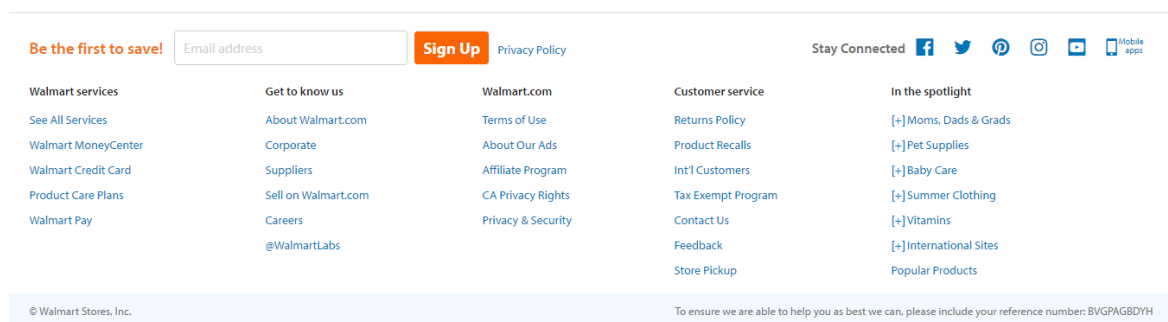


Obr. 10 – Záhloví na úvodní straně

7.3 Zápatí

Zápatí je součástí každé webové stránky viz obrázek 11. V zápatí se vyskytují nejčastěji odkazy, jako jsou uživatelská podpora, stránky třetích stran, Privacy and policy, Term of Services a podobně.

Je potřeba otestovat funkčnost těchto odkazů.



Obr. 11 – Zápatí na úvodní straně

7.4 Stránka produktu

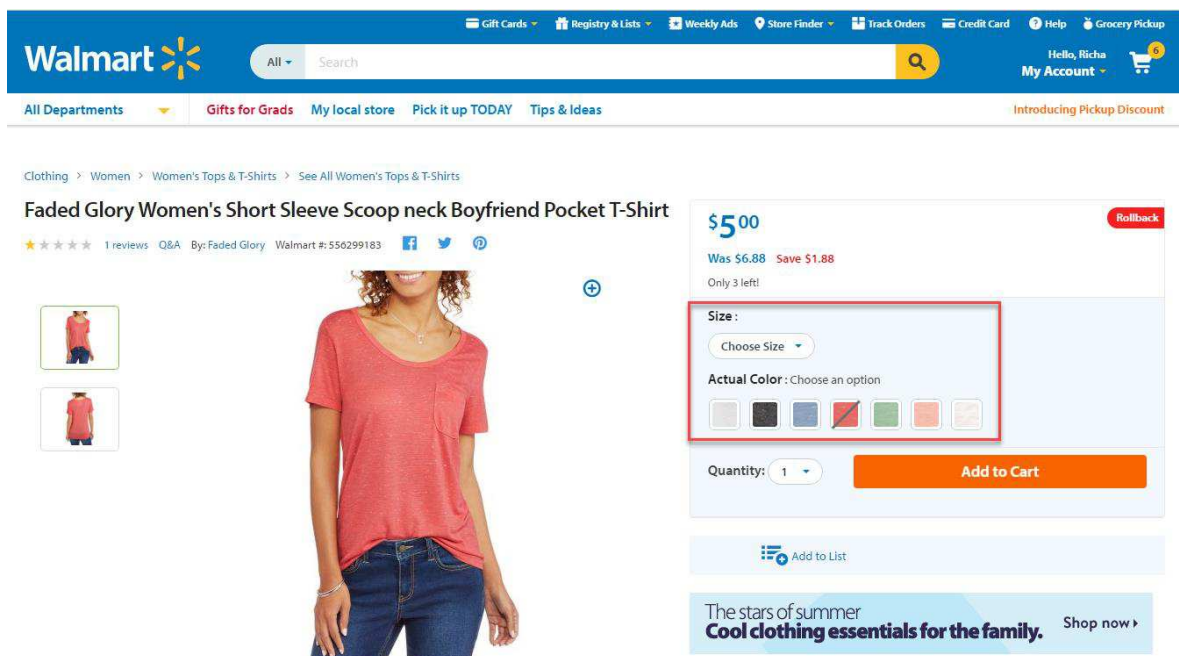
Nejkritičtější oblast, jelikož se jedná o stránku, kde uživatel přichází do styku s produktem, o který má zájem a je potřeba zajistit, že uživateli nic nezabrání v tom si tento produkt koupit.

Walmart podporuje velké množství různých produktů, které se z uživatelského hlediska nemusí nijak lišit, ale každý typ produktu má jiný kód a tudíž se po hlubším prozkoumání chová jinak.

Jedná se zejména o typy

- Regular – obyčejný produkt, který nenabízí žádné možnosti konfigurace a lze ho pouze přidat do košíku
- Variant – zde je možnost vybrat jednu možnost. Nejčastěji se dá vybrat například barva nebo velikost.
- Multi-Variant – jak výše zmíněné napovídá, tak se jedná o předmět, který má na výběr více možností, například u triček se jedná o velikost a barvu viz obrázek 12.
- Submap – produkt, který je momentálně v nějaké speciální nabídce a na stránce se uživateli nezobrazuje cena produktu, ale ju mu zobrazena až později v nákupním košíku.

- PPU – u těchto předmětů je cena udána jako cena za kus, ale je možnost si nakoupit pouze více těchto předmětů.
- Multi-seller – produkt, který nabízí také jiní prodejci, než je Walmart. Je možnost si tento produkt koupit od jiného výrobce než je Walmart a to za pomoci stránky walmart.com.
- Marketplace – produkt, který vůbec Walmart nemá v nabídce, ale je možné si ho objednat u jiného prodejce.
- Diapers – dětské plínky jsou podobné Multi-Variant ale liší se v tom, že změna možnosti také změni název produktu.
- Care plan – předmět u kterého lze navíc přidat pojištění.



Obr. 12 – Stránka produktu s typem Multi-Variant

Na stránce produktu je nejdůležitější testovat možnost přidání předmětu do košíku a od toho se odvíjí všechny manuální testy. Všechna další funkcionalita je sekundární, a proto je nejvíc prostoru věnováno právě přidání produktu do košíku.

Stránka obsahuje také další sekce jako jsou zákaznické recenze anebo dotazy a odpovědi.

Je potřeba také otestovat funkčnost všech tlačítek a z nich vyplývající funkcionalitu.

7.5 Okno po přidání do košíku

Tato část se zobrazí uživateli, jakmile přidá produkt do košíku. V této části je možné ještě udělat drobné změny a zkontrolovat si správnost daného produktu.

Je zobrazen také důležitý údaj o aktuálním stavu košíku, jako je celková hodnota nebo jednotlivé ceny daní, poštovného a různé informace, jestli je zákazník oprávněn k nějaké slevě.

Jak bylo zmíněno v předchozí kapitole, tak Walmart podporuje velké množství různých druhů produktů a proto je potřeba tuto část testovat pro každý jednotlivý produkt.

7.6 Nákupní košík

Košík slouží ke zřehlednění objednávky a také k poslední editaci daného zboží před objednávkou.

Lze tady provádět akce, jako je mazání předmětu z košíku anebo přesunutí pro pozdější nákup.

Košík také obsahuje informaci o ceně a jednotlivý rozpis, za co zákazník zaplatí, viz obrázek 13.

The screenshot displays the Walmart shopping cart interface. At the top, there's a blue navigation bar with the Walmart logo and various utility links. Below this, a promotional banner for a credit card offer is visible. The main content area shows the cart items. The first item is a 'Straight Talk Apple iPhone 5S 16GB 4G LTE Prepaid Smartphone' with a quantity of 1, priced at \$99.00 (was \$450.00, save \$351.00). Below the item name, there are options for shipping and pickup, and buttons for 'Save for later' and 'Remove'. The second item is 'Sensodyne Complete Protection Toothpaste, Extra Fresh, 3.4 oz' with a quantity of 1, priced at \$4.92 (was \$1.43, save \$3.49). Similar to the first item, it has shipping/pickup options and 'Save for later'/'Remove' buttons. On the right side, a summary box shows the 'Subtotal (18 items)' as \$1,098.42, 'Shipping' as Free, and 'Est. Taxes and fees' as 'Not Calculated'. The 'Est. total' is \$1,098.42. Below the summary is a prominent orange 'Check Out' button and a message: 'Congrats — you get free shipping! Free pickup is available as soon as today. More about shipping and pickup'.

Obr. 13 – Nákupní košík zobrazující celkovou cenu a předměty v košíku

Jedním z neméně důležitých bodů je také změna poštovního směrovacího čísla neboli ZIP kódu. Změnou ZIP kódu může dojít k tomu, že předmět, který byl dostupný je nyní

nedostupný a tím pádem se musí změnit obsah košíku. Uživatel je v tomto případě dotázán, jestliže chce pokračovat v nákupu bez tohoto předmětu, anebo ještě změnit ZIP kód.

7.7 Objednávka

Stránka samotné objednávky provede uživatele vyplněním všech důležitých bodů, jako je doručovací adresa, typ poštovního a volba způsobu platby, tak aby bylo možné objednávku dokončit. Největším problémem v tomto případě je platba samotná. Walmart nabízí mnoho možností jak zaplatit a tím může dojít k různým problémům jako je nepřesměrování na danou platební bránu anebo odmítnutí méně používaného typu kreditních karet. Krátký výčet možnosti jak zaplatit je například VISA, MasterCard, Visa Checkout, Cash, Paypal a AMEX.

Při objednávce je například také možné si zvolit možnost přihlášení anebo vytvoření nového účtu viz obrázek 14. Od těchto možností se pak dále odvíjí chování objednávky. V případě přihlášení má uživatel možnost hned potvrdit objednávku, protože všechny informace o sobě již vyplnil. V případě nového účtu je uživatel nucen přidat adresu doručení, zvolit způsob platby anebo také druh dopravy popřípadě vyzvednutí na pobočce.

The screenshot displays the Walmart Checkout interface. At the top, a blue header contains the Walmart logo and 'Checkout' text, along with a shopping cart icon. A yellow warning banner states: 'For security reasons, your session has expired. Please sign in again.' Below this, the page is divided into three main sections. On the left is the 'Sign In' section, which includes input fields for 'Email address (required)' and 'Password (required)', a 'Show' button for the password, a 'Forgot password?' link, and an orange 'Sign In' button. A Norton Secured logo is visible at the bottom of this section. The middle section is for 'Create new account', featuring a sub-header 'Enjoy faster checkout with an account.' and a '*required field' label. It contains input fields for 'First name*', 'Last name*', 'Email address*', and 'Password*', with a 'Show' button for the password. Below these fields is a disclaimer: 'By clicking Create Account, you acknowledge you have read and agreed to our Terms of Use and Privacy Policy.' and an orange 'Create Account' button. At the bottom of this section is a checkbox for 'Email me about Rollbacks, special pricing, hot new items, gift ideas and more.' On the right side of the page, a summary box shows: 'Subtotal \$1,098.42', 'Shipping Free', 'Est. tax Not Calculated Based on 94066 change', and a bold 'Est. total \$1,098.42' with a 'See item details' link.

Obr. 14 – Objednávka – přihlášení a vytvoření nového účtu

8 PROVEDENÍ NAVRHNUTÝCH TESTŮ

Navrhnuté testy v kapitole 7 byly provedeny na zařízeních zmíněných v kapitole 6.

Pro průchod testovacími případy bylo využito prostředí webové aplikace TestRail, kde jsou spravovány jednotlivé testovací případy.

Nejdříve bylo nutné vytvořit si nový testovací plán pro každé zařízení. Testovací plán vždy obsahuje sadu všech 72 testovacích případů, celkem tedy bylo za potřebí si vytvořit 14 různých testovacích plánů pojmenovaných podle konkrétního zařízení.

Při každém průchodu je potřeba si otevřít daný testovací plán, kde se poté zobrazí všechny testovací případy. Po kliknutí na testovací případ se otevře nová stránka, viz obrázek 15, kde jsou popsány jednotlivé kroky pro průchod tímto testovacím případem. Je také zobrazen očekávaný výsledek testu a po projití testovacího případu vždy musí uživatel docílit očekávaného výsledku.

The screenshot shows a TestRail test case interface. At the top, there is a header bar with the text 'C61 Způsob platby - Discovery'. Below this, the test case title 'Checkout' is displayed. A table provides details about the test case:

Type	Priority	Estimate	References
Other	Medium	None	None

Below the table, there are sections for 'Preconditions', 'Steps', and 'Expected Result':

- Preconditions:** Uživatel je přihlášen a má v košíku nějaké předměty.
- Steps:**
 1. Z košíku klikněte na tlačítko Checkout
 2. V Shipping and pickup sekci vybrat možnost Shipping
 3. Pokračovat dále
 4. Vybrat libovolnou adresu dodání
 5. Pokračovat dále
 6. Jako druh platby zvolit Credit Card
 7. Přidat novou kartu typu Discovery
 8. Kliknout na tlačítko Potvrdit objednávku
- Expected Result:** Objednávka je potvrzena a uživatel je přesměrován na stránku s přehledem objednávek

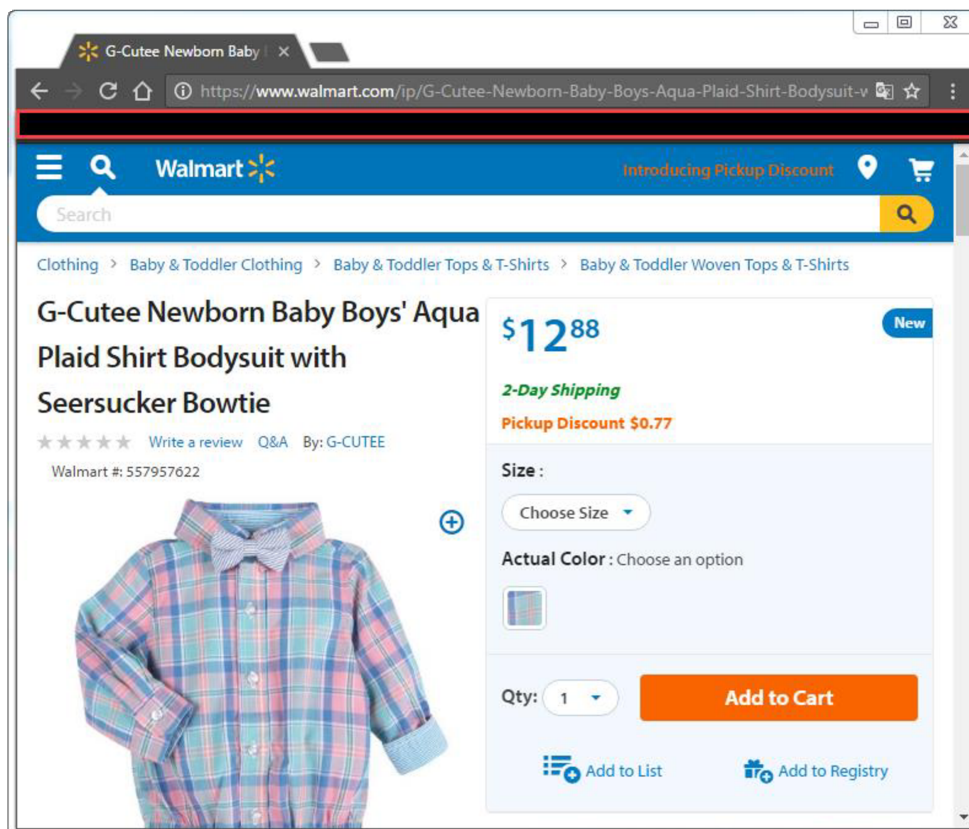
Obr. 15 – Testovací případ v aplikace TestRail

8.1 Windows 7 – Chrome

Jedno z nejvíce používaných prostředí, toto prostředí je i přes své stáří stále používané a obecně uznávané, jako jedno z nejspolehlivějších a proto je v takovém prostředí očekáváno nejméně chyb.

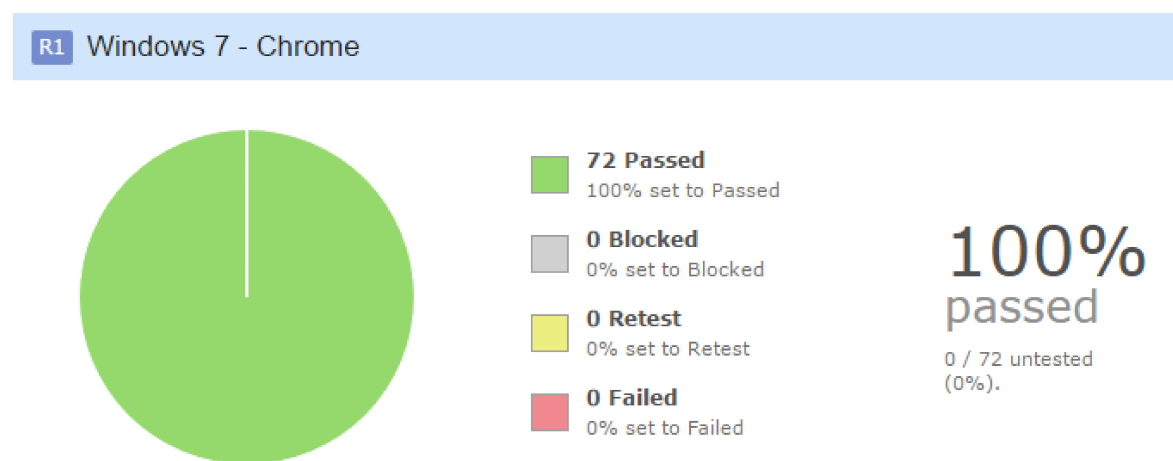
Při testování na prohlížeči Chrome se očekává rychlost, rychlá odezva a bezchybnost. Testování desktopové verze aplikace Walmart se neobjevily žádné problémy, speciálně jsem se zaměřil na testování rychlosti při rolování stránkou nahoru a dolů, ale ani tady se neprojeví menší problémy.

Speciální vlastností aplikace Walmart je responzivnost, viz obrázek 16.



Obr. 16 – Testování responzivnosti prohlížeče Chrome

Responzivnost znamená, že zmenšování a zvětšování okna prohlížeče automaticky překreslí grafické rozhraní stránky, ale ani tady se mi nepodařilo nalézt jediný problém.



Obr. 17 – Test plán - Výsledek testovacího plánu Windows 7 + Chrome

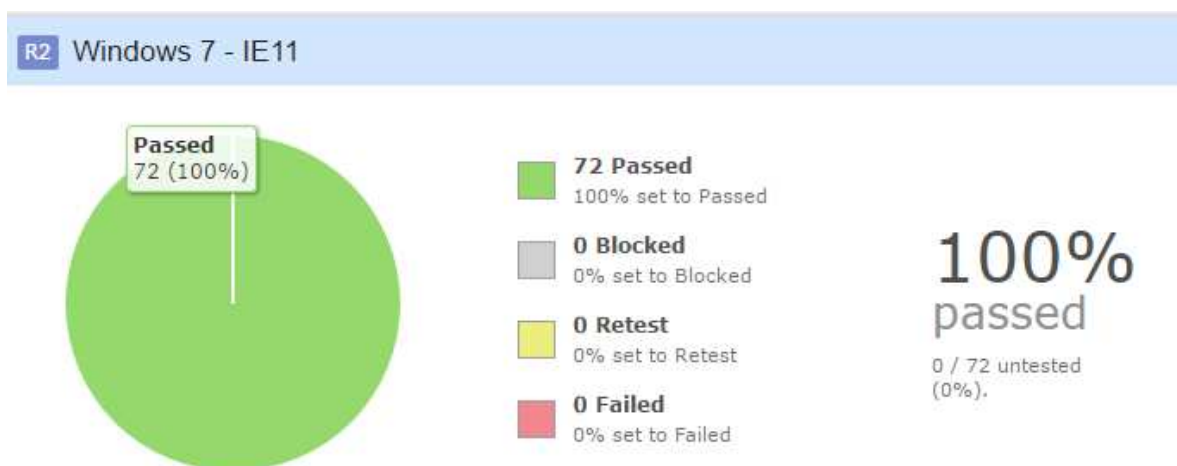
Z testu tedy vyplývá, že všech 72 testovacích případů prošlo bez chyby. Dokonce bylo provedeno uživatelské testování použitelnosti na rámec zmíněných testů, ale ani tady nebyla odhalena jediná chyba.

Tabulka 2 - Windows 7 + Chrome

Windows 7 + Chrome				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.2 Windows 7 – IE11

Internet Explorer bývá často cílem kritiky jako nejhorší prohlížeč v historii, nicméně toto možná byla pravda u předchozích verzí, jako je IE9 a nižší. V aktuální době Internet Explorer 11, je velice spolehlivý a podporuje veškeré funkcionality stejně tak jako Chrome.



Obr. 18 – Test plán - Windows 7 + IE11

Testování Internet Exploreru 11 probíhalo podobně jako testování prohlížeče Chrome. Jediný výrazný rozdíl oproti chování Chrome je při rychlém rolování stránkou nahoru a dolů. Bohužel IE11 není tak vyladěný jak Chrome a při větších stránkách může docházet

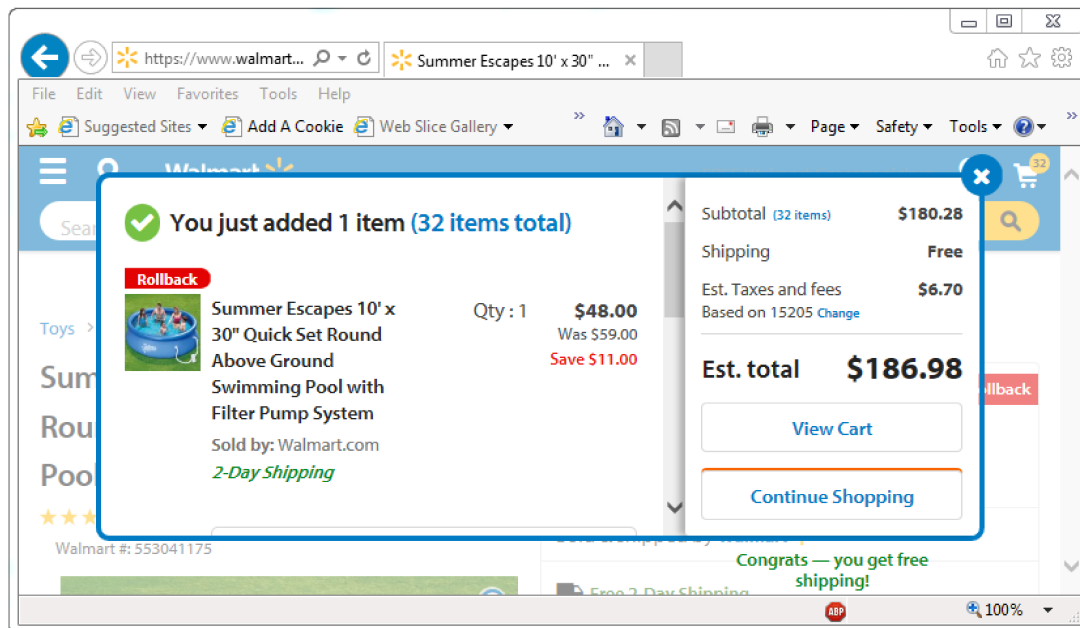
k očividnému zpomalení při rolování. Nicméně toto není chyba stránky Walmart, ale prohlížeče a proto tato chyba nebyla zavedena do systému pro správu chyb.

Tabulka 3 - Windows 7 + IE11

Windows 7 + IE11				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

Abych potvrdil nebo vyvrátil tvrzení, že Internet Explorer je nejhorším prohlížečem v historii, tak jsem provedl i rychlý test na Windows XP spolu s Internet Explorer 8 a v tomto případě se opravdu potvrdily obrovské nedostatky a nekompatibilita se stránkou Walmart.

Opětovně byl proveden test responzivnosti viz obrázek 19, ale tentokrát na jiné oblasti než v případě Chrome. Pro responzivnost byla vybrána oblast okno po přidání produktu do košíku.

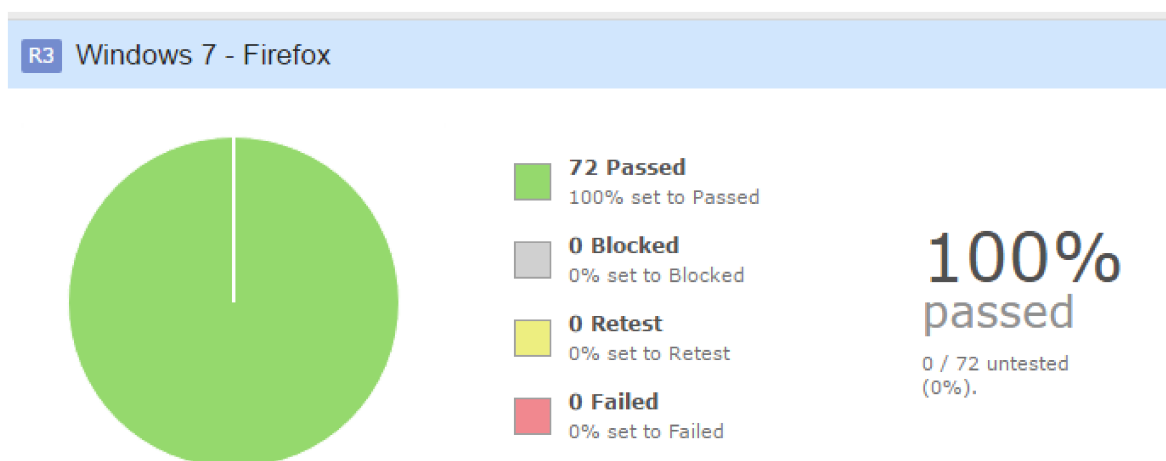


Obr. 19 – Responzivnost okna po přidání do košíku v prohlížeči IE11

8.3 Win7 – Firefox

Firefox je obecně znám jako velice dobrá alternativa k Chrome, proto se v tomto prohlížeči neočekávají žádné chyby, stejně tak jak je tomu u Chrome.

Při testování nedocházelo k žádným problémům s rychlostí nebo odezvou, rolování stránky je bezchybné stejně jako v Chrome.



Obr. 20 – Test plán - Windows 7 + Firefox

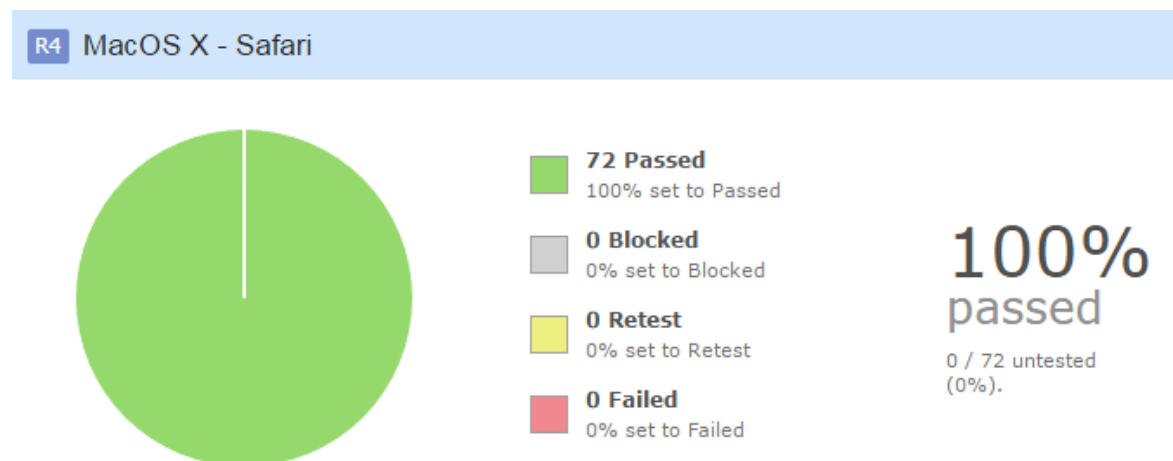
Bylo použito také jedno z limitních testování a to v tom případě, kdy uživatel kliká na tlačítka rychle opakovaně za sebou, nicméně tento problém je eliminován rychlostí prohlížeče a nevyskytly se žádné chyby.

Tabulka 4 - Windows 7 + Firefox

Windows 7 + Firefox				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.4 MacOS X – Safari

Safari je nejpoužívanějším prohlížečem na operačním systému MacOS. Safari je známo svou stabilitou, rychlostí a bezproblémovým chodem.



Obr. 21 – Test plán - MacOS X + Safari

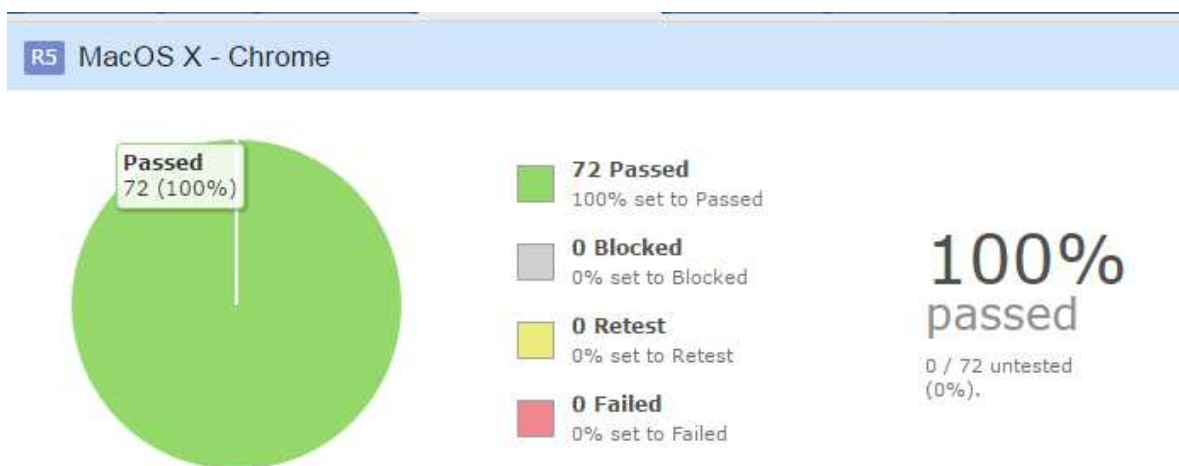
Pro testování byl použit MacBook Air 13“ s rozlišením 1440x900 pixelů, toto rozlišení je lehce odlišné než 1920x1080 na stolním PC a proto by se daly očekávat chyby v grafickém rozhraní. Nicméně jak bylo zmíněno aplikace Walmart je plně responzivní takže ani v tomto případě se nevyskytly žádné chyby.

Tabulka 5 - MacOS X + Safari

MacOS X - Safari				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.5 MacOS X – Chrome

Chrome je druhým nejpoužívanějším prohlížečem na MacOS. Testování probíhalo podobně jak v předchozích prohlížečích, byly provedeny všechny testovací případy a navíc i testy responzivnosti.



Obr. 22 – Test plán - MacOS X + Chrome

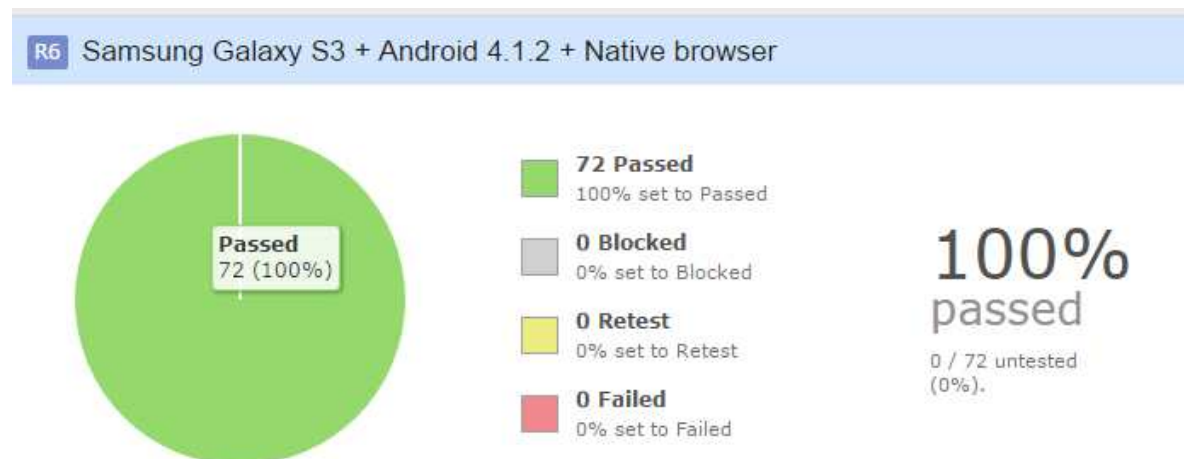
Rychlost a doba odezvy je stejná jako při testování prohlížeče Chrome na Windows, nebyly nalezeny tedy žádné problémy.

Tabulka 6 - MacOS X + Chrome

MacOS X - Chrome				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.6 Samsung Galaxy S3 + Android 4.1.2 + Nativní prohlížeč

Galaxy S3 je jedním z nejstarších a mezi nimi i nejpobulárnějších telefonů od firmy Samsung, tento telefon je stále používán kvůli své rychlosti a bezproblémovému chodu.



Obr. 23 – Test plán - Samsung Galaxy S3 + Android 4.1.2 + Native broker

Jak již bylo zmíněno, je nutné projít testovací případy na všech možných kombinacích a Android 4 je poslední podporovaná verze stránkou Walmart a tudíž by měla podporovat vše, co podporuje i na verzi vyšší než Android 5.

Jedná se zejména o responzivnost, funkčnost tlačítek, zobrazení speciálních vyskakovacích oken a hlavně rychlosti. O rychlosti by se dalo spekulovat, ale vše je v rámci uživatelské přívětivosti. Android 4 se nedá srovnávat s novými zařízeními podporujícími Android 7, ale i přes jeho stáří, je rychlost dostačující.

Tabulka 7 - Samsung Galaxy S3 + Android 4.1.2 + Native browser

Samsung Galaxy S3 + Android 4.1.2 + Native browser				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

Android 4 jsem využil hlavně k otestování stránky, kdy telefon je otočen na ležato a v tomto režimu jsem prošel celý testovací plán, viz obrázek 24.

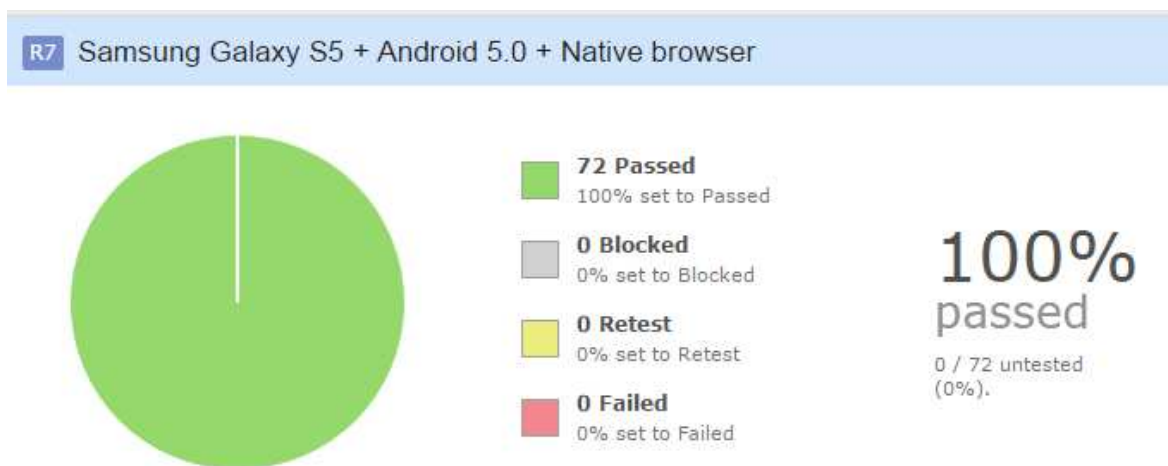


Obr. 24 – Testování zobrazení obrázku v režimu na ležato

Hlavním bodem testování na tomto zařízení bylo ověření vyskakovacích oken a prohlížení obrázků produktů a také ověření, že nízké rozlišení telefonu nezpůsobí žádné problémy s grafickým rozhraním.

8.7 Samsung Galaxy S5 + Android 5.0 + Nativní prohlížeč

Galaxy S5 je nástupcem velice populárního modelu S4, bohužel se mu jej nikdy nepodařilo předčít, protože tento model nenabízel žádné výrazné zlepšení oproti předchozímu modelu, jedinou výraznou novinkou byla voděodolnost, ale tato vlastnost nehraje roli na poli testování webových aplikací.



Obr. 25 - Samsung Galaxy S5 + Android 5.0 + Native broker

Nicméně Android 5 na tomto zařízení je výrazně svižnější než dříve zmíněný Android 4, proto na tomto zařízení bylo testováno také přibližování neboli zoom.

Přibližování částečně a pouze na některých stránkách způsobí změnu rozložení grafického rozhraní a ani v tomto případě by nemělo dojít například k překrývání textu anebo jeho úplné zmizení, viz obrázek 26.



Google Home is fun

Obr. 26 – Přiblížení stránky a ověření, že vše se zobrazuje správně

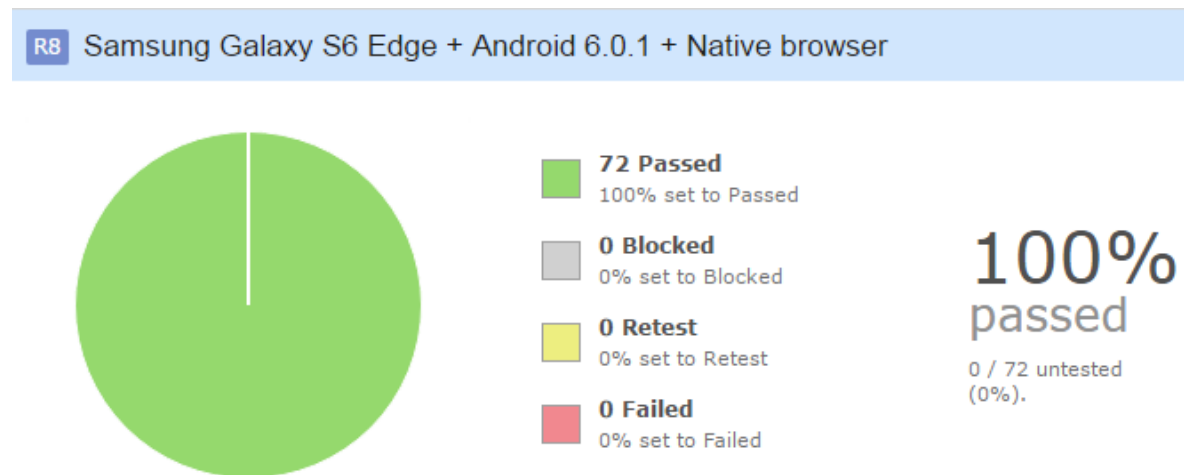
Obdobně jako u předchozích zařízení nedošlo k žádné chybě a byl kladen velký důraz na responzivnost a rychlost načítání stránek.

Tabulka 8 - Samsung Galaxy S5 + Android 5.0 + Native browser

Samsung Galaxy S5 + Android 5.0 + Native browser				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.8 Samsung Galaxy S6 Edge + Android 6.0.1 + Nativní prohlížeč

S6 Edge se stal velice populární jako první telefon se zaoblenými kraji. Rozdíl oproti starším zařízením je i ve větším rozlišení a to 1140x2560 pixelů.



Obr. 27 - Samsung Galaxy S6 Edge + Android 6.0.1 + Nativní prohlížeč

S6 Edge se dá stále ještě považovat za high end telefon a proto je u něj předpoklad bezproblémového chodu a rychlosti. Proto při testování byl kladen vysoký důraz právě na testování rychlosti.

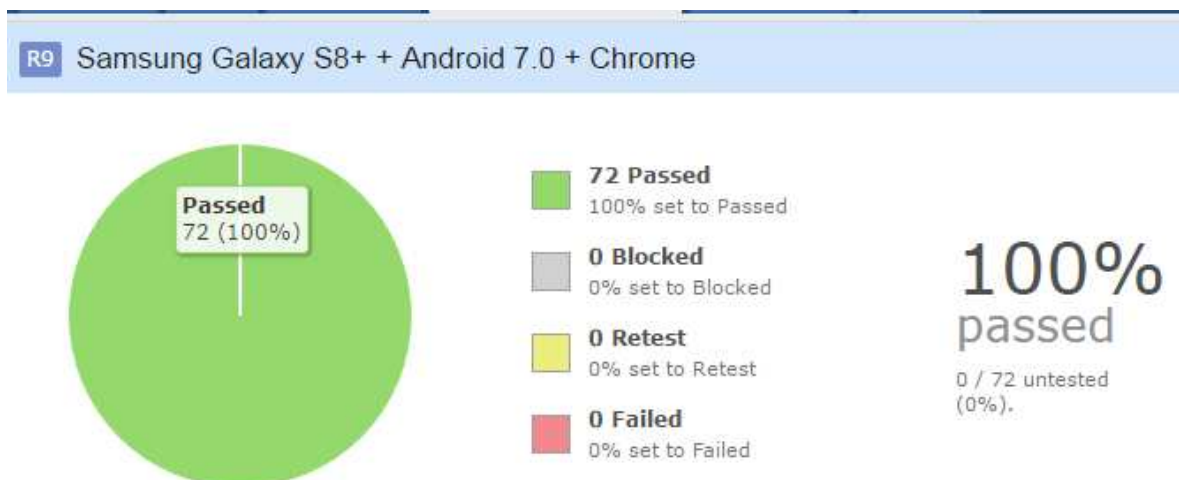
Tabulka 9 - Samsung Galaxy S6 Edge + Android 6.0.1 + Nativní prohlížeč

Samsung Galaxy S6 Edge + Android 6.0.1 + Native browser				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.9 Samsung Galaxy S8+ + Android 7.0 + Chrome

Galaxy S8 plus je posledním vydaným telefonem od společnosti Samsung, hlavní předností je jeho obrovský displej téměř bez okrajů a proto si mohli dovolit na něj umístit dosud největší možné rozlišení 1440x2960.

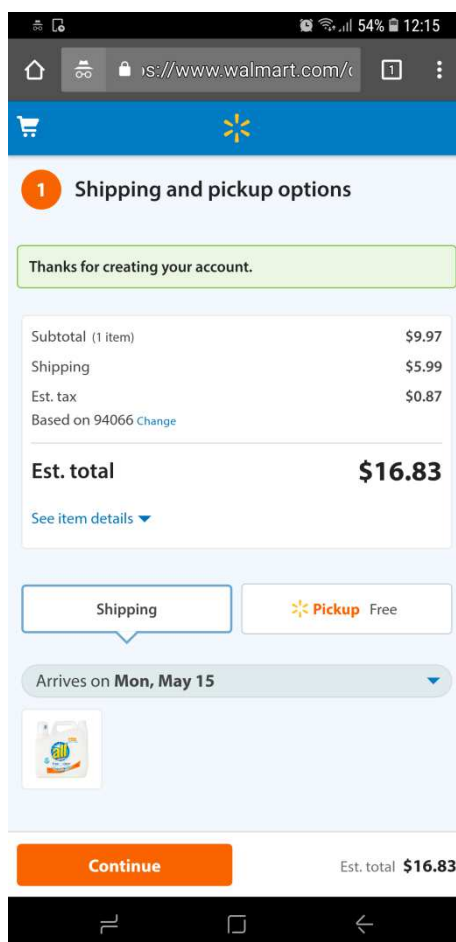
Dalším rozdílem oproti předchozímu testování zařízení podporující Android byla změna prohlížeče na prohlížeč Chrome. Chrome je plně podporovaný prohlížeč na Androidu a proto by se neměli ani tady vyskytovat žádné chyby.



Obr. 28 - Samsung Galaxy S8+ + Android 7.0 + Chrome

Výše zmíněné rozlišení dává možnost nalézt zejména chyby v grafickém rozhraní, protože se jedná o dosud neexistující rozlišení.

Při testování jsem se hlavně zaměřil tedy na problémy s grafickým rozhraním a největší důraz byl kladen na nákupní košík a stránku s objednávkou, viz obrázek 29, a to jak v módu na ležato tak klasickém módu na stojato.



Obr. 29 – Testování objednávky
na Samaung Galaxy S8+

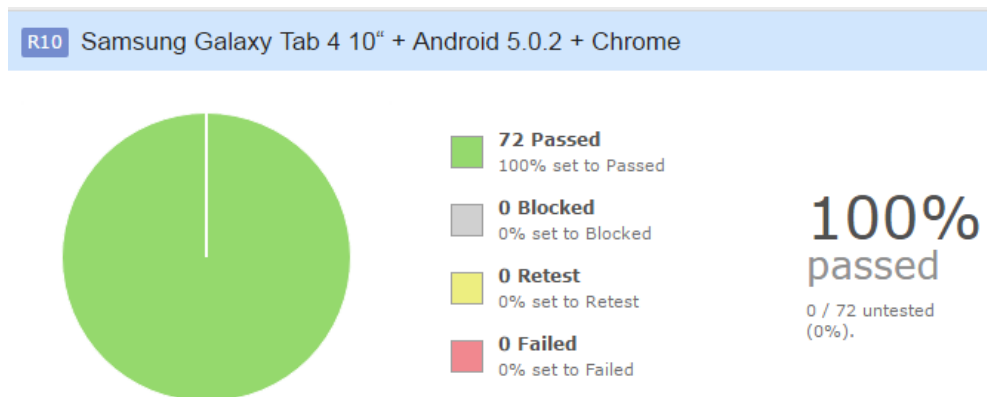
Tabulka 10 - Tabulka 11 - Samsung Galaxy S8+ + Android 7.0 + Chrome

Samsung GalaxyS8+ + Android 7.0 + Chrome				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%

Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.10 Samsung Galaxy Tab 4 10“ + Android 5.0.2 + Chrome

Galaxy Tab 4 je mým jediným tabletem běžícím na systému Android. Zvolil jsem testování v prohlížeči Chrome, protože nativní prohlížeč na tomto zařízení se potýká s velkými problémy a je skoro nepoužitelný.



Obr. 30 - Samsung GalaxyTab 4 10“ + Android 5.0.2 + Chrome

Chrome, oproti nativnímu prohlížeči, je rychlé a nepotýká se s problémy při vykreslování stránek.

Nejdůležitějším bodem testování tabletu je přechod mezi módy na ležato a stojato, protože dochází k velkému překreslení grafického rozhraní z mobilní verze na desktopovou verzi díky rozdílu rozlišení v jednotlivých módech.

Tabulka 12 - Samsung GalaxyTab 4 10“ + Android 5.0.2 + Chrome

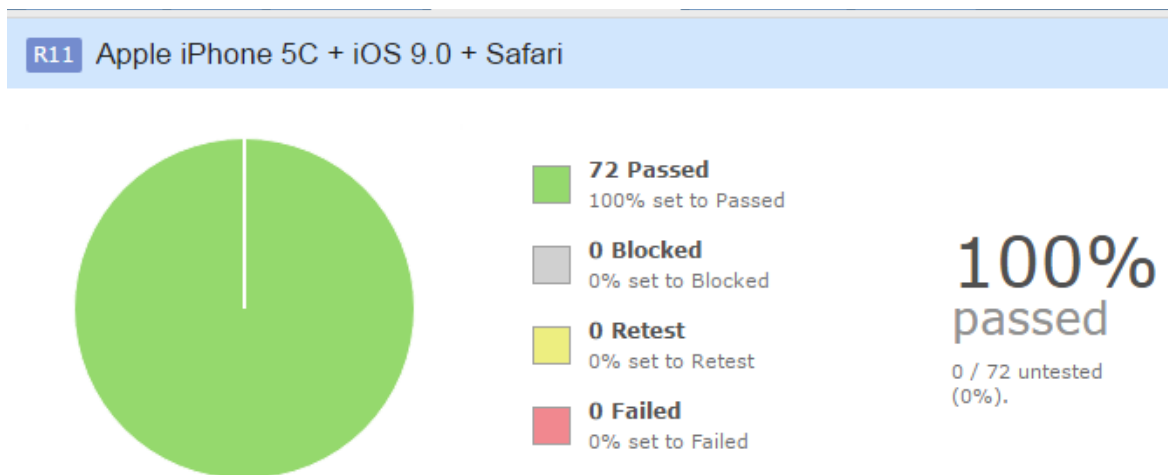
Samsung GalaxyTab 4 + Android 5.0.2 + Chrome				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%

Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

Testování tedy probíhalo ve smyslu otestovat každou stránku jak v módu naležato tak i na stojato, což zabralo neúměrně moc času a proto jsem to při dalším testování tabletu již vynechal

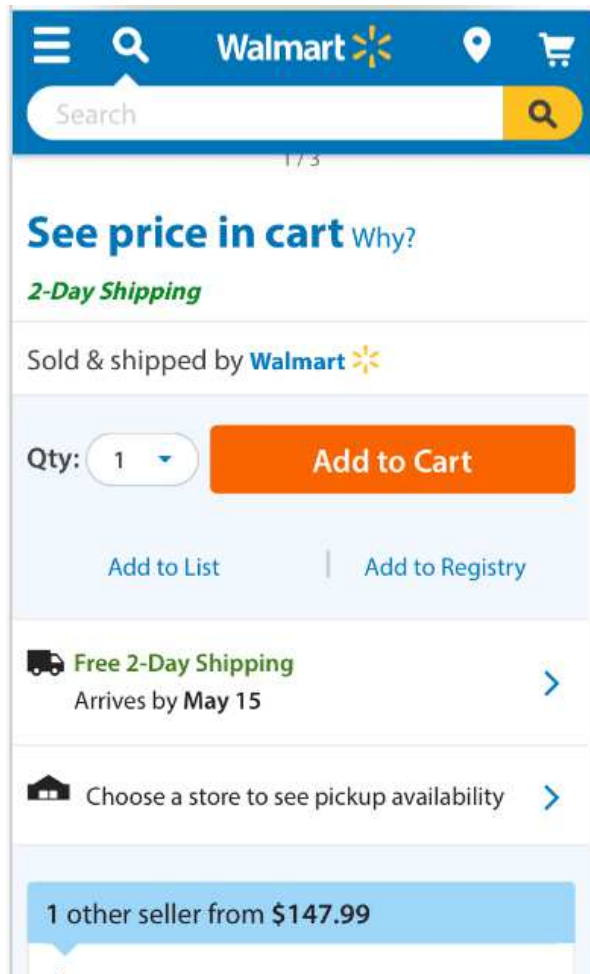
8.11 Apple iPhone 5C + iOS 9.0 + Safari

iPhone 5C je obdobou iPhone 5 ale pouze s rozdílem barvy krytu telefonu, nicméně hlavním důvodem volby tohoto zařízení je jeho nízké rozlišení 640x1136 pixelů, jedno z nejmenších na trhu a zároveň podporující veškerou funkcionalitu stránky Walmart, viz obrázek 32.



Obr. 31 - Apple iPhone 5C + iOS 9.0 + Safari

Na tomto zařízení byl kladen velký důraz na grafické rozhraní a to hlavně na stránky produktů.



Obr. 32 – Submap produkt na zařízení
iPhone 5C

Také byla věnována pozornost zejména responzivnímu chování při mnohonásobném klikání na tlačítka, zdali se zařízení zachová tak jak má a nedojde k neočekávané chybě.

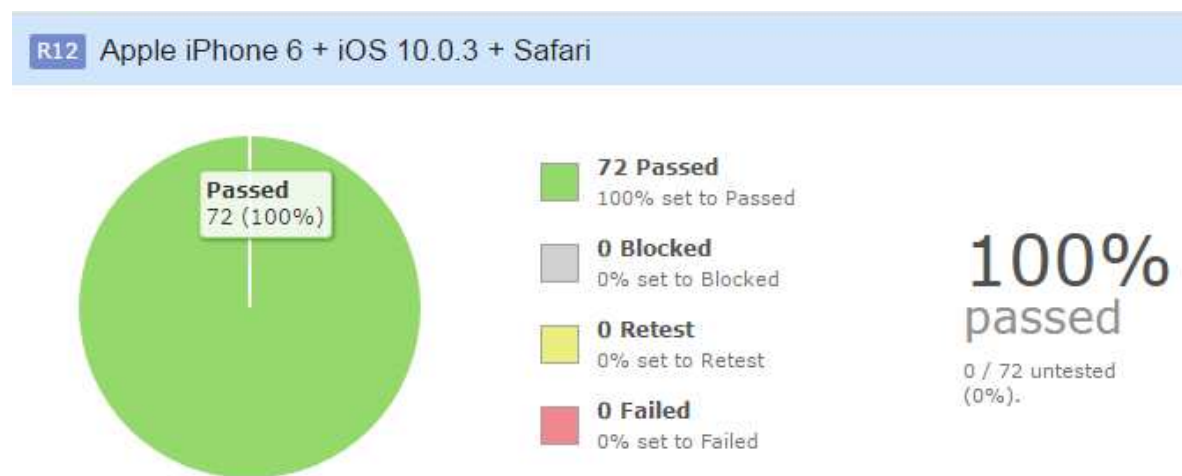
Tabulka 13 - Apple iPhone 5C + iOS 9.0 + Safari

iPhone 5C + iOS 9.0 + Safari				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%

Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.12 Apple iPhone 6 + iOS 10.0.3 + Safari

iPhone 6 se stal velice populárním díky své velikosti oproti předchůdcům verze 5. Telefon také změnil kompletně svůj design a z hranatých zařízení se staly zaoblené a padnoucí do ruky.



Obr. 33 - Apple iPhone 6 + iOS 10.0.3 + Safari

iPhone 6 je svým rozlišením velice podobný verzi 5 a proto neočekáváme žádné výraznější problémy nebo rozdíly oproti iPhone verze 5. Nicméně byl zvolen jiný přístup testování a to, že celý testovací plán jsem prošel v módu na ležato, abych docílil jiných výsledků oproti iPhone 5C, nicméně ani tady se nevyskytly žádné chyby jak v grafickém rozhraní, tak ani v rychlosti a odezvě.

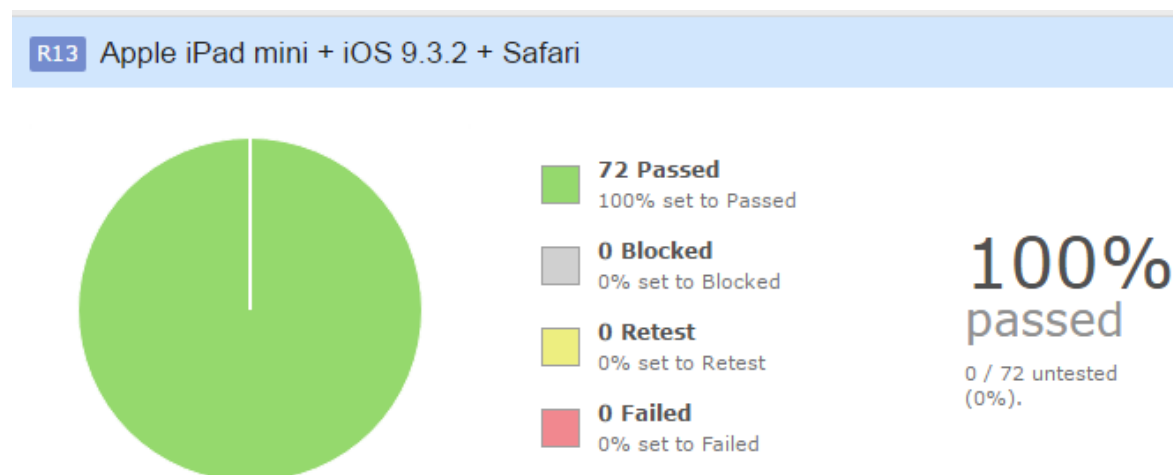
Tabulka 14 - Apple iPhone 6 + iOS 10.0.3 + Safari

iPhone 6 + iOS 10.0.3 + Safari				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%

Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.13 Apple iPad mini + iOS 9.3.2 + Safari

iPad mini je prvním tabletem s velikostí displeje 7,9“ od firmy Apple. Toto zařízení je poměrně staré, vyráběno v roce 2012 a proto by se od něj dali očekávat problémy s rychlostí, ale rychlost není ovlivněna hardwarem v případě webových aplikací, ale spíše je ovlivněna prohlížečem.



Obr. 34 - Apple iPad mini + iOS 9.3.2 + Safari

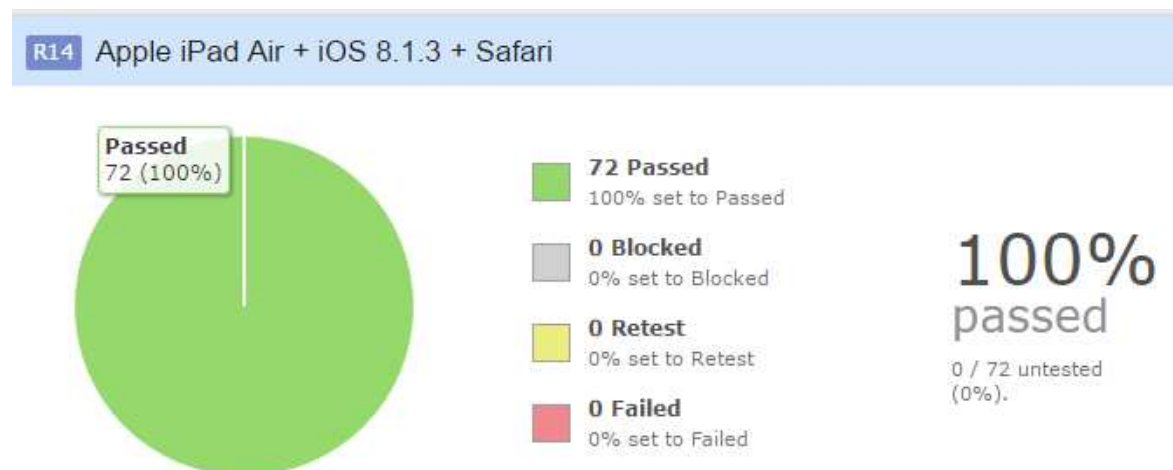
Jelikož testování probíhalo v prohlížeči Safari, tak se žádné rychlostní problémy neočekávaly a to se mi také potvrdilo. Nicméně opět bylo hlavním bodem testování přechodu mezi módem na ležato a na stojato, sice ne v takové míře jako při testování tabletu se systémem Android, ale i tak stálo za to tento přechod vyzkoušet. Je nutno podotknout, že rozlišení tohoto tabletu v obou módech podporuje pouze zobrazení webové stránky v módu pro mobilní zařízení.

Tabulka 15 - Apple iPad mini + iOS 9.3.2 + Safari

iPad mini + iOS 9.3.2 + Safari				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

8.14 Apple iPad Air + iOS 8.1.3 + Safari

iPad Air jsem zvolil z důvodu toho, že běží na operačním systému iOS verze 8. Verze 8 v dnešní době ustupuje do postraní a čím dál tím více nebývá podporována různými aplikacemi.



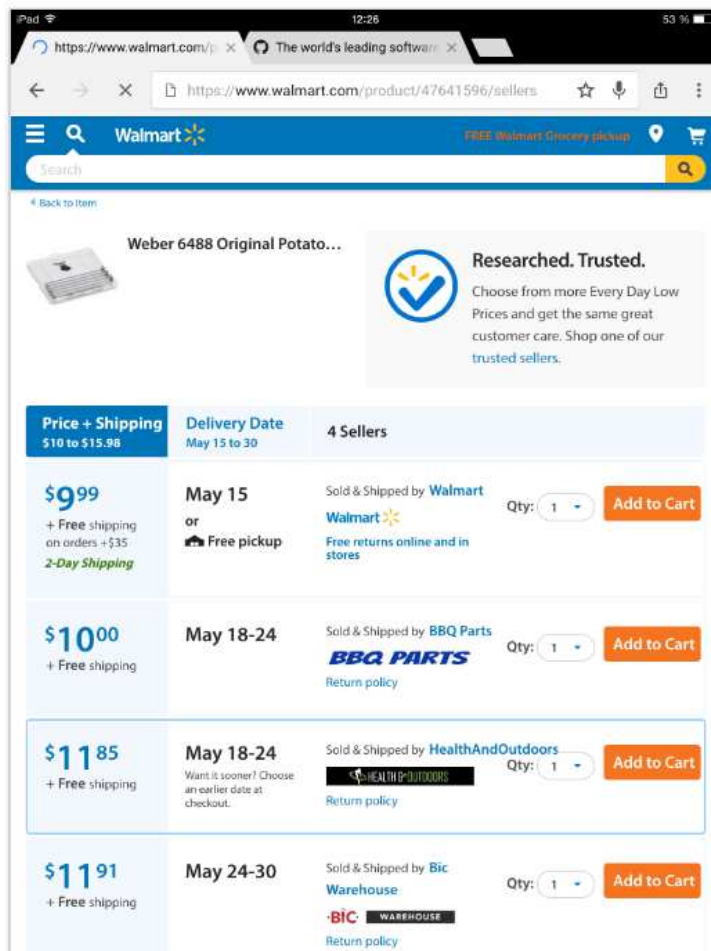
Obr. 35 - Apple iPad Air + iOS 8.1.3 + Safari

Při testování tohoto tabletu jsem zvolil podobný přístup jako v předchozím testování tabletů a zaměřil jsem se na rychlost, grafické rozhraní, viz obrázek 36, a přechod mezi módem na ležato a stojato.

Tabulka 16 - Apple iPad Air + iOS 8.1.3 + Safari

iPad Air + iOS 8.1.3 + Safari				
Sekce	Počet testů	Pass	Fail	Úspěšnost
Homepage	5	5	0	100%
Header	3	3	0	100%
Footer	1	1	0	100%
Itempage	23	23	0	100%
PAC	12	12	0	100%
Cart	5	5	0	100%
Checkout	21	21	0	100%
Historie objednávek	2	2	0	100%

Nicméně je nutno podotknout, že iPad Air vyniká svou rychlostí při načítání stránek i přestože běží na tak starém operačním systému.



Obr. 36 – Testování GUI na tabletu iPad Air

9 ANALÝZA REALIZOVANÉHO TESTU

Celkem bylo otestováno 72 testovacích případů na 14 různých zařízeních.

Jak již bylo zmíněno, jednalo se o regresní testování produkčního prostředí, které je ve své podstatě stabilní a nemůže si dovolit, aby na tomto prostředí byly závažné chyby.

Přestože nebyly nalezeny žádné chyby, není radno regresní testování vynechávat, protože v každém novém update se může vyskytnout něco, co se neobjeví při běžném testování testovacího prostředí.

Nad rámec testovacích případů bylo také provedeno několik nestandardních testovacích postupů jako je rychlé přepínání mezi módy zobrazení na ležato a stojato, rychlo klikání na tlačítka v aplikaci, zmenšování a zvětšování oken v případě prohlížečů na stolním počítači.

Tabulka 17 - Analýza testu

Zařízení	Počet testů	chybovost
Windows 7 - Chrome	72	0
Windows 7 - IE11	72	0
Windows 7 - Firefox	72	0
MacOS X - Safari	72	0
MacOS X - Chrome	72	0
Samsung Galaxy S3 + Android 4.1.2 + Native browser	72	0
Samsung Galaxy S5 + Android 5.0 + Native browser	72	0
Samsung Galaxy S6 Edge + Android 6.0.1 + Native browser	72	0
Samsung Galaxy S8+ + Android 7.0 + Chrome	72	0
Samsung GalaxyTab 4 10" + Android 5.0.2 + Chrome	72	0
Apple iPhone 5C + iOS 9.0 + Safari	72	0
Apple iPhone 6 + iOS 10.0.3 + Safari	72	0
Apple iPad mini + iOS 9.3.2 + Safari	72	0
Windows 7 - Chrome	72	0
Apple iPad Air + iOS 8.1.3 + Safari	72	0

9.1 Stanovení priorit

Jak již bylo zmíněno, regresní testování proběhlo na 100%, nebyly objeveny žádné chyby a tím pádem není nutné stanovovat pevné priority.

Nicméně budou stanoveny priority testování dle daného testovacího plánu a to každých 14 dní, dále bylo navrženo, že před každým vydáním nové verze do produkce bude také proveden regresní test postaven na vytvořeném testovacím plánu.

V případě, že se později vyskytnou chyby je potřeba priority nastavit znovu a zohlednit výsledky testů.

V budoucnu je možné také testovací plán rozšířit o další oblasti anebo přidat nové testovací případy do aktuálních oblastí.

ZÁVĚR

Tato diplomová práce měla jako hlavní cíl seznámit čtenáře srozumitelnou formou s testováním softwaru na obecné úrovni, zařadit testování do širšího kontextu celého procesu vývoje informačního systému a najít rozdíly při testování webové aplikace pro mobilní zařízení a webové aplikace pro klasické počítače.

Pro nalezení rozdílů v testování jsem využil hlavně moje 10-leté zkušenosti s testováním webových aplikací a softwaru obecně.

V první části jsou uvedeny základní charakteristiky testování softwaru. Rozbor kategorií testů obsahuje pouze nejvýraznější rozdělení, která se při testování vyskytují a která jsem pro účely této práce považoval za dostatečná. Část práce je věnována zhodnocení argumentů, které testování podporují nebo jsou naopak proti jeho použití a doporučení, zda při vývoji softwaru testovat či nikoliv. Dospěl jsem k očekávanému závěru, který hovoří ve prospěch testování. Zmiňovaná časová a finanční náročnost je častý argument proti testování a pokud dojde v předaném softwaru k závažné chybě, náklady na její odstranění mohou lehce překročit úspory, vzniklé vynecháním testů.

Hlavním cílem diplomové práce bylo vytvoření testovacího plánu vybrané webové aplikace. Testovací případy jsou zpracovány formou sešitu Microsoft Excel. Sešity jsou rozděleny podle oblastí webové aplikace. Jednotlivé oblasti na sebe úzce navazují a k těmto oblastem jsou vytvořeny reporty z testování.

Sada testovacích scénářů byla napsána ve webové aplikaci TestRail a poté vyexportována do souboru Excel, Celkem bylo vytvořeno 72 testovacích scénářů, které zahrnují celou nejkritičtější oblast dané webové aplikace.

Na závěr bylo celkem provedeno 1008 testovacích případů, což odpovídá 14 různým zařízením a nebyla nalezena žádná chyba v produkčním prostředí. Navíc při testování byly aplikovány nestandardní postupy za účelem nalezení chyb, které by bylo velice těžké odhalit jako obyčejný uživatel, nicméně ani tyto přístupy neodhalily žádné chyby, které by bylo potřeba zdokumentovat.

Po ukončení testování došlo k vyhodnocení těchto provedených testovacích plánů a tento testovací plán je nyní vhodný pro regresní testování dle stanovených priorit.

SEZNAM POUŽITÉ LITERATURY

- [1] KOHL, Jonathan. TapInto Mobile ApplicationTesting [online]. [cit. 2016-11-17]. Dostupné z: <https://leanpub.com/testmobileapps>
- [2] GOUSSET, Mickey. Řízení životního cyklu aplikací ve Visual Studiu 2010. Vyd. 1. Překlad Jan Pokorný. Brno: ZonerPress, 2010, 671 s. ISBN 978-80-7413-102-8.
- [3] PATTON, Ron. Testování softwaru. Vyd. 1. Praha: ComputerPress, 2002. ISBN 80-722-6636-5.
- [4] MÜLLER, Thomas, Debra FRIEDENBERG, Armin BEER a Martin KLONK. Certified Tester: FoundationLevelSyllabus [online]. 2011, , 1-78 [cit. 2016-11-17]. Dostupné z: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html4>
- [5] PAGE, Alan, Ken JOHNSTON a Bj ROLLISON. Jak testuje software Microsoft. Vyd. 1. Brno: ComputerPress, 2009, 384 s. ISBN 978-80-251-2869-5. Dostupné z: <https://books.google.cz/books?id=vQzqCwAAQBAJ&lpg=PA2&ots=fYvfBQtJ8L&dq=Jak%20testuje%20software%20Microsoft.%20Vyd.%201.%20Brno&hl=cs&pg=PA106#v=onepage&q=Jak%20testuje%20software%20Microsoft.%20Vyd.%201.%20Brno&f=false>
- [6] J. P. CemKaner, „WhatIs a Good Test Case?“, Florida Institute of Technology, Department ofComputerSciences, May 2003. [Online]. Dostupné z: <http://www.kaner.com/pdfs/GoodTest.pdf>. [Přístup získán 5 2012].
- [7] TheEssentialGuide to Mobile AppTesting [online]. , 1-48 [cit. 2016-11-17]. Dostupné z: core.ecu.edu/STRG/materials/uTest_eBook_Mobile_Testing.pdf
- [8] A. BUCHALCEVOVÁ a J. KUČERA, „Hodnocení metodik vývoje informačních systémů z pohledu testování“, Katedra informačních technologií VŠE Praha, 2008. [Online]. Available: <http://nb.vse.cz/~buchalc/clanky/testovani.pdf>. [Přístup získán 5 2012].
- [9] MICROSOFT. Testing Performance and Stress UsingVisual Studio Web Performance and LoadTests[online]. © 2015 [cit. 2015-01-30]. Dostupné z: [https://msdn.microsoft.com/en-us/library/dd293540\(v=vs.110\)](https://msdn.microsoft.com/en-us/library/dd293540(v=vs.110))

- [10] RATNAKAR, Nancy. *Beginner's Guide to Mobile Application Testing* [online]. 2016 [cit. 2016-11-17]. Dostupné z: <http://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/>
- [11] *Black box testing and its advantages and disadvantages* [online]. 2013 [cit. 2017-05-11]. Dostupné z: <http://jobsandnewstoday.blogspot.cz/2013/04/what-is-black-box-testing-and-discuss-the-advantages-and-disadvantages-of-black-box-testing.html>
- [12] *Android App Publishing: Reading Android Market Crash Reports* [online]. 2012 [cit. 2017-05-11]. Dostupné z: <https://code.tutsplus.com/tutorials/android-app-publishing-reading-android-market-crash-reports--mobile-9801>
- [12] *Software Testing And ISTQB* [online]. 2015 [cit. 2017-05-11]. Dostupné z: <http://www.softwaretestingandistqb.com/what-is-spiral-development-model/>
- [13] *Tech Hacks: Android: Problem Loading Widget Error – Solved* [online]. 2012 [cit. 2017-05-11]. Dostupné z: <http://tech.chandrasaha.com/2012/06/20/android-fixing-problem-loading-widget-error/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SDLC Software development life cycle

ICT Information and Communication Technologies

SEZNAM OBRÁZKŮ

Obr. 1 – Přístup k black-box testování [11].....	14
Obr. 2 – Ukázky chyby typu	19
Obr. 3 – Spirálový model SDLC [13].....	24
Obr. 4 – Logo android [13].....	29
Obr. 5 – Úvodní stránka testované aplikace Walmart	33
Obr. 6 – Systémové informace telefonu	35
Obr. 7 – Systémové informace	36
Obr. 8 – Ukázka aplikace TestRail s vytvořenými testovacími případy	41
Obr. 9 – Karusely pro posun mezi seznamem produktů.....	42
Obr. 10 – Záhlaví na úvodní straně.....	42
Obr. 11 – Zápatí na úvodní straně.....	43
Obr. 12 – Stránka produktu s typem Multi-Variant.....	44
Obr. 13 – Nákupní košík zobrazující celkovou cenu a předměty v košíku	45
Obr. 14 – Objednávka – přihlášení a vytvoření nového účtu	46
Obr. 15 – Testovací případ v aplikace TestRail.....	47
Obr. 16 – Testování responzivnosti prohlížeče Chrome.....	48
Obr. 17 – Test plán - Výsledek testovacího plánu Windows 7 + Chrome	48
Obr. 18 – Test plán - Windows 7 + IE11	49
Obr. 19 – Responzivnost okna po přidání do košíku v prohlížeči IE11	51
Obr. 20 – Test plán - Windows 7 + Firefox.....	51
Obr. 21 – Test plán - MacOS X + Safari	52
Obr. 22 – Test plán - MacOS X + Chrome.....	53
Obr. 23 – Test plán - Samsung Galaxy S3 + Android 4.1.2 + Native broker.....	54
Obr. 24 – Testování zobrazení obrázku v režimu na ležato.....	56
Obr. 25 - Samsung Galaxy S5 + Android 5.0 + Native broker.....	56
Obr. 26 – Přiblížení stránky a ověření, že vše	57
Obr. 27 - Samsung Galaxy S6 Edge + Android 6.0.1 + Nativní prohlížeč.....	58
Obr. 28 - Samsung Galaxy S8+ + Android 7.0 + Chrome	59
Obr. 29 – Testování objednávky.....	60
Obr. 30 - Samsung GalaxyTab 4 10“ + Android 5.0.2 + Chrome	61
Obr. 31 - Apple iPhone 5C + iOS 9.0 + Safari	62
Obr. 32 – Submap produkt na zařízení	63

Obr. 33 - Apple iPhone 6 + iOS 10.0.3 + Safari.....	64
Obr. 34 - Apple iPad mini + iOS 9.3.2 + Safari	65
Obr. 35 - Apple iPad Air + iOS 8.1.3 + Safari	66
Obr. 36 – Testování GUI na tabletu iPad Air	68

SEZNAM TABULEK

Tabulka 1 – Verze Systému Android.....	29
Tabulka 2 - Windows 7 + Chrome.....	49
Tabulka 3 - Windows 7 + IE11.....	50
Tabulka 4 - Windows 7 + Firefox.....	52
Tabulka 5 - MacOS X + Safari.....	53
Tabulka 6 - MacOS X + Chrome.....	54
Tabulka 7 - Samsung Galaxy S3 + Android 4.1.2 + Native browser.....	55
Tabulka 8 - Samsung Galaxy S5 + Android 5.0 + Native browser.....	58
Tabulka 9 - Samsung Galaxy S6 Edge + Android 6.0.1 + Nativní prohlížeč.....	59
Tabulka 10 - Tabulka 11 - Samsung Galaxy S8+ + Android 7.0 + Chrome.....	60
Tabulka 12 - Samsung GalaxyTab 4 10“ + Android 5.0.2 + Chrome.....	61
Tabulka 13 - Apple iPhone 5C + iOS 9.0 + Safari.....	63
Tabulka 14 - Apple iPhone 6 + iOS 10.0.3 + Safari.....	64
Tabulka 15 - Apple iPad mini + iOS 9.3.2 + Safari.....	66
Tabulka 16 - Apple iPad Air + iOS 8.1.3 + Safari.....	67
Tabulka 17 - Analýza testu.....	69

SEZNAM PŘÍLOH

P I: Seznam příloh přiložených na CD

PŘÍLOHA P I: SEZNAM PŘÍLOH PŘILOŽENÝCH NA CD

Podklady pro testovací scénáře

- Excel soubor s testovacími scénáři