

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Responzivní design a možnosti současného webu

Bakalářská práce

Autor: Martin Štěpánek

Studijní obor: Aplikovaná Informatika

Vedoucí práce: Ing. Aneta Bartůšková

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 29.4.2015

Martin Štěpánek

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Anetě Bartůškové za metodické vedení práce a cenné rady v průběhu psaní.

Anotace

Práce se zabývá problematikou zobrazování webových stránek na různých typech zařízení pomocí responzivního designu a jeho optimalizací. Prvním cílem práce je objasnit, jaké náležitosti musí takový design splňovat, jaké principy se používají pro jeho tvorbu, jakými problémy se v současnosti zabývá a jak se liší od techniky separovaného mobilního webu. Ve druhé části práce jsou podle předem stanovených kritérií srovnány a zhodnoceny existující responzivní či mobilní webové stránky ve zvolené oblasti a nakonec jsou představeny praktické nástroje pro tvorbu responzivního designu.

Annotation

Title: Responsive Design and Possibilities of Contemporary Web

The bachelor thesis deals with displaying web pages on different types of devices using responsive design and its optimization. First goal of this work is to clarify the essentials that should be satisfied for responsive design, principles that are used for its creation, problems that is currently dealing with and differences from the technique of separated mobile site. In the second part there is comparison and evaluation of existing responsive and mobile sites in the selected area using predetermined criteria and at last introduction of practical tools for creating responsive design.

Obsah

1	Úvod.....	1
2	Responzivní design.....	3
2.1	Různorodost zobrazovacích zařízení.....	4
2.1.1	Osobní počítače	4
2.1.2	Tablety	5
2.1.3	Chytré telefony	6
2.1.4	Omezení mobilních zařízení.....	7
2.1.5	Metatag Viewport.....	8
2.2	Definice responzivního designu	9
2.3	Fluidní layout.....	10
2.4	Mediální dotazy.....	12
2.4.1	Media Types.....	12
2.4.2	Media Queries	13
2.4.3	Směr vývoje.....	15
2.5	Fluidní obrázky	15
2.5.1	Nové atributy elementu 	16
2.5.2	Formát SVG	17
2.6	Výkon	18
2.7	Responzivní design vs. Mobilní web	22
2.7.1	Mobilní web	22
2.7.2	Responzivní design	23
3	Příklady implementace a jejich porovnání.....	24
3.1	Stanovení kritérií.....	24
3.2	Testování vybraných stránek	25

3.2.1	Praha.....	26
3.2.2	Plzeň	29
3.2.3	Olomouc	32
3.2.4	Hradec Králové	34
3.2.5	Most.....	36
3.2.6	Opava.....	38
3.2.7	Karlovy Vary	40
3.3	Celkové vyhodnocení testu.....	42
4	Praktické nástroje pro tvorbu responzivního designu.....	43
4.1	Front-end frameworky.....	43
4.2	JavaScriptové pluginy	44
4.3	Nástroje pro testování.....	45
4.4	Vizuální editory responzivního designu.....	46
5	Shrnutí výsledků.....	49
6	Závěry a doporučení	51
7	Seznam zdrojů	53
8	Seznam obrázků.....	56
9	Seznam tabulek.....	58
10	Přílohy.....	59

1 Úvod

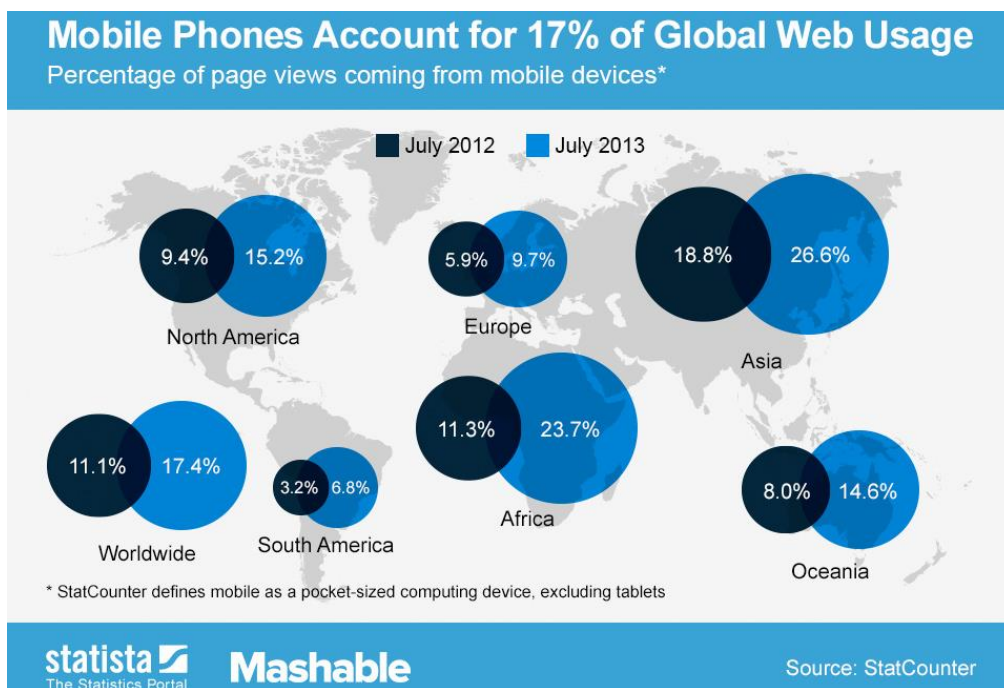
Cílem každého provozovatele webových stránek je nalákat na svůj obsah co nejvíce návštěvníků. To znamená poskytnout informace na své stránce co nejsrozumitelněji, nejatraktivněji, ale také nejpřehledněji; tedy navrhnout stránku tak, aby se co největší skupině uživatelů zobrazovala podle plánu. I když šel vývoj monitorů rychle kupředu, stále zde bylo příliš mnoho uživatelů procházejících web na starších, menších obrazovkách, kteří by mohli mít problémy se stránkami přizpůsobenými velkým šířkám, a když už postupem času doba tyto nejmenší monitory vytěsnila z trhu, průměr začaly snižovat nejrůznější kompaktní notebooky a netbooky. Standard pro šířku internetových stránek se tak v poslední době dlouho pohyboval okolo jednoho tisíce pixelů.

Velkým milníkem byl příchod internetu do mobilních telefonů. Z počátku nebylo nutné stránky nějak přizpůsobovat, prohlížení na malých obrazovkách s tlačítky či joysticky místo myši bylo nepříjemné, takže nezpůsobovalo příliš velkou revoluci. Zároveň majitelé PDA přístrojů nebyli natolik rozšíření. Problém pro vývojáře nastal s poměrně rychlým rozšířením dotykových obrazovek a chytrých telefonů. Internet v telefonu přestal být využíván pouze v nouzi, naopak mu začalo propadat více a více lidí. Mobilní internet a chytré telefony, potažmo tablety, se staly nedílnou součástí lidských životů. Pro udržení co největší návštěvnosti na svých stránkách museli provozovatelé začít přijímat určitá opatření. Nejčastějším řešením návštěvnosti z těchto zařízení se stalo sestavování separátních mobilních webů; jedná se o stránku, která je obsahově, a pokud možno i vzhledově, víceméně identická s desktopovou verzí webu, avšak přizpůsobená menším displejům. Přizpůsobení většinou spočívá v přeskládání obsahu do jednoho dlouhého sloupce. Tento způsob má dvě nevýhody. Zaprvé je nutné udržovat dva samostatné weby stejně aktualizované, aby se všem uživatelům dostalo stejného obsahu, a zadruhé se arzenál těchto menších, mobilních zařízení rozšířil natolik, že by provozovatel stránek musel ve výsledku udržovat ne dva, ale hned několik webů o různém rozlišení.

V roce 2010 byl definován pojem responzivní design. Jedná se o řešení, které nejen nahradí mobilní web a dokáže díky své adaptivitě obsloužit pod jednou doménou nejmenší telefony i největší televize, ale zabývá se i pokročilejšími otázkami prohlížení internetu nejen na příručních zařízeních. Jedná se o velmi dynamickou problematiku, která je probírána hlavně v online médiích, proto i práce vychází převážně z elektronických zdrojů.

2 Responzivní design

S chytrými telefony, tablety nebo třeba už i s elektronickou čtečkou a současným internetovým pokrytím poskytovaným mobilními operátory se dá říci, že si většina lidí v podstatě nosí internet v kapse. Jasně mluví i čísla, podle služby Gemius Ranking bylo na českém internetu k dubnu roku 2015 12,63 % celkového provozu vygenerováno přes mobilní zařízení (Gemius SA, 2015). V celosvětovém měřítku to v roce 2013 bylo celých 17 % (Richter, 2013). Některé statistiky z roku 2011 dokonce odhadovaly, že v roce 2015 bude více amerických uživatelů přistupovat na internet skrz mobilní telefon než přes počítač (Sterling, 2011). Je tedy více než jasné, že tato zařízení již dávno není možné ignorovat. Většina vývojářů začala řešit tuto rozmanitost separovaným mobilním webem. V mnoha případech nemusí jít o špatné řešení, vývojáři je umožněno se při výrobě mobilního webu zaměřit na technické schopnosti mobilních zařízení, jako je GPS modul či různé senzory, a vylepšit tak výsledný prožitek, který by na desktopovém webu nebylo možné nabídnout. Aby se ale tímto přístupem skutečně pokryly všechny typy zařízení, ze kterých se dá k internetu připojit, muselo by takových samostatných webů vzniknout mnohem víc.



Obr. 1: 17 % všech přístupů na internet bylo z mobilních telefonů (Richter, 2013)

2.1 Různorodost zobrazovacích zařízení

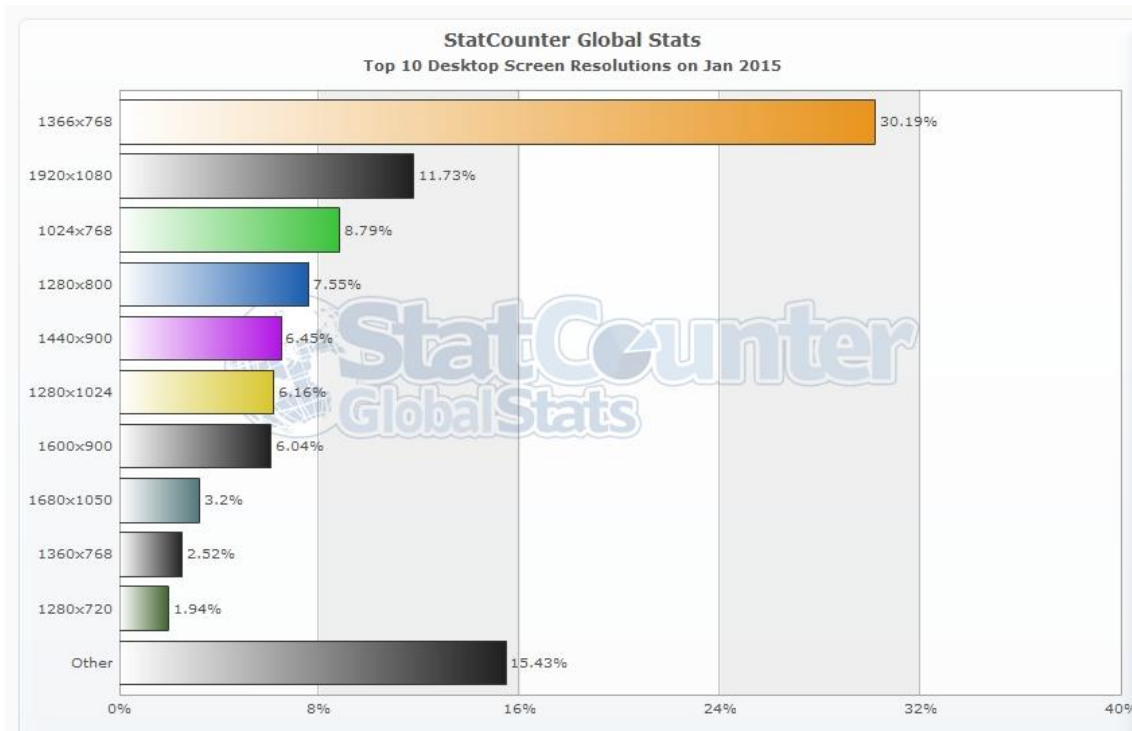
Podle slov v úvodu se v současné době nabízí jednoduché rozdělení zobrazovacích zařízení do tří kategorií. To by znamenalo, že si stačí objednat tři verze webových stránek a mít tak pokryté veškeré uživatele a přístroje, ze kterých na internet přistupují. Bohužel to tak jednoduché není.

Nejsledovanějším parametrem u jednotlivých přístrojů je rozlišení obrazovky udávané v obrazových bodech (pixelech, zkráceně px). Jeden pixel představuje jeden svítící bod na obrazovce zařízení. Rozlišení obrazovky je potom počet pixelů, který se na obrazovce zobrazí.

2.1.1 Osobní počítače

Za osobní počítače se dají považovat jak notebooky, tak stolní počítače. Dále by se nabízelo zařadit do této kategorie ještě malé netbooky, od jejich výroby však v posledních letech vývojářské firmy ustupují a plně je nahrazují oblíbenější tablety.

Externí monitory, pro připojení ke stolním počítačům a případně k notebookům, většinou nabývají větších rozměrů a rozlišení – úhlopříčka se nejčastěji pohybuje mezi devatenácti a třiceti palci (1 palec = 2,54 centimetrů). Notebooky se pro lepší mobilitu osazují obrazovkami nejčastěji o velikosti zhruba třinácti až sedmnácti palců. Nejrozšířenější jsou širokouhlé obrazovky s poměrem 16:9, případně pro práci v různých grafických a jiných specializovaných programech s poměrem 16:10. Zhruba od roku 2004 bylo nejrozšířenější rozlišení 1024×724 pixelů. Až v březnu roku 2012 se s 19 % uživatelů stalo nejrozšířenějším rozlišení 1366×768 pixelů, které si dnes stále drží prvenství se zhruba 30 % uživatelů. Na druhé místo vystoupalo s 12 % uživatelů Full HD rozlišení 1920×1080 pixelů (Statcounter, 2015).



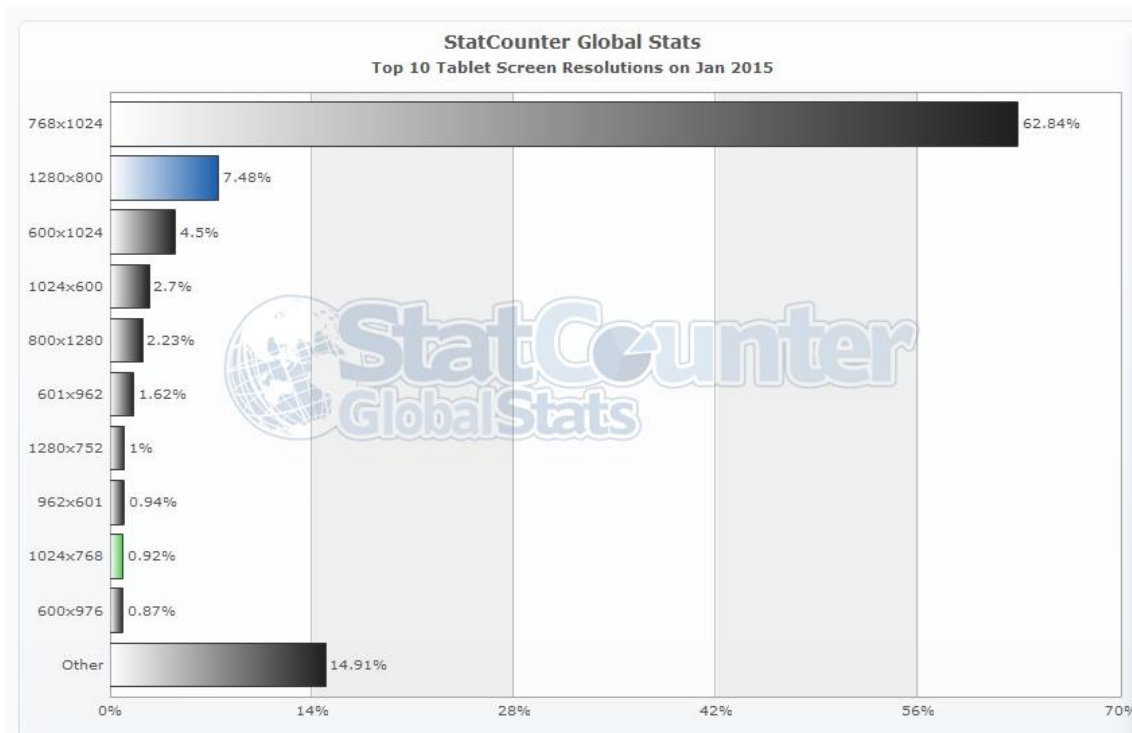
Obr. 2: Top 10 rozlišení osobních počítačů k lednu 2015 (Statcounter, 2015)

2.1.2 Tablety

U tabletů je kladen ještě větší důraz na přenositelnost. S přídatnými periferiemi, jako je třeba klávesnice, dokáží plně nahradit notebook, o čemž svědčí i možnost pořídit si tablet s operačním systémem Windows 8. Společnost Google provedla v roce 2011 průzkum, ve kterém se majitelů tabletů ptala na otázky ohledně jeho využívání. Z výzkumu vyplynulo, že 78 % respondentů na tabletu právě hledá informace na webu. Celých 43 % respondentů tráví na tabletu více času než na jejich stolním PC či notebooku a 82 % respondentů používá tablet převážně doma. 28 % z nich ho dokonce používá jako svůj primární počítač (Shah, 2011).

Tablet tedy vůbec není zařízení pouze na cestování a množství lidí používající je každodenně vypovídá o tom, že uživatelé těchto zařízení očekávají na internetu obsah ve stejné formě, v jaké by se jim dostal na PC.

Velikosti tabletů se pohybují zhruba od sedmi do deseti palců, nejčastěji s rozlišeními 768×1024 a 1280×800 pixelů. Není ale problém sehnat tablet s větším rozlišením, větší displeje disponují běžně rozlišením 1366×768 nebo 1920×1080 pixelů (Statcounter, 2015).

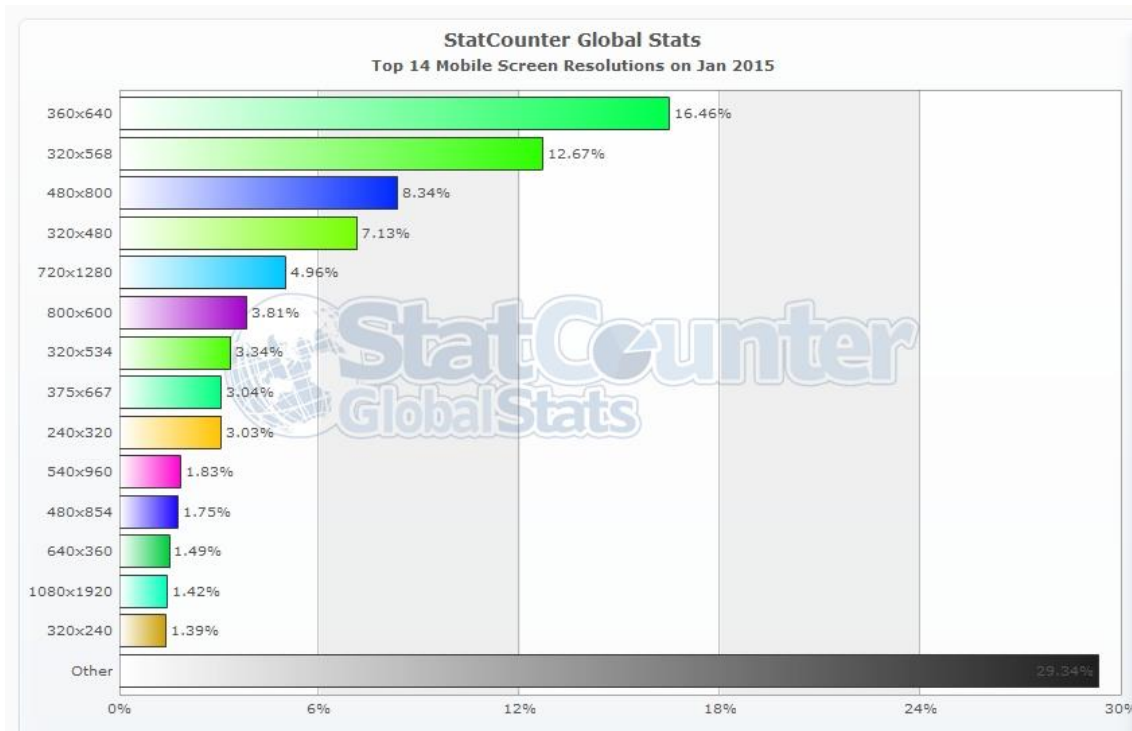


Obr. 3: Top 10 rozlišení tabletů k lednu 2015 (Statcounter, 2015)

2.1.3 Chytré telefony

Jelikož je telefon díky své velikosti pořád po ruce, stává se nejrychlejším zdrojem informací a je mezi lidmi víceméně považován za standard. Přesto výzkum z roku 2012 dokázal, že 68 % respondentů používá chytrý telefon hlavně doma (Harvard Business Review Staff, 2013).

Velikost displeje se pohybuje nejčastěji od tří a půl do pěti palců. Čím dál častěji se objevují telefony s ještě větším displejem a některé se už svou velikostí rovnají nejmenším tabletům. Pro tyto „extra velké“ chytré telefony vzniklo pojmenování phablet. Phablety mají kombinovat vlastnosti chytrých telefonů a tabletů, jak z názvu snadno vyplývá, a jejich velikost se pohybuje právě od pěti do sedmi palců. Rozlišení chytrých telefonů se nejčastěji pohybuje od nejmenších 320×480 přes 720×1280 až po 1080×1920 pixelů (Statcounter, 2015).



Obr. 4: Top 14 rozlišení mobilních telefonů k lednu 2015 (Statcounter, 2015)

Předešlé řádky ukazují, že rozpětí rozlišení v jednotlivých kategoriích jsou obrovská a není tedy možné uspokojivě obsloužit větší množství různých zařízení pod jednou statickou stránkou.

2.1.4 Omezení mobilních zařízení

Při vytváření stránky optimalizované pro mobilní přístroje je důležité mít na paměti omezení v ovládání oproti počítačům, kterým přesné ovládání myši a klávesnicí poskytuje několik výhod.

Několik takových omezení shrnul článek na blogu Baymard Institutu. Prvním omezením je velmi oblíbený stav `hover` využívaný prakticky na každé stránce. Tento stav se může aplikovat na jakýkoliv prvek na stránce; je v něm definováno, jak se daný prvek změní, pokud přes něj uživatel přejeđe kurzorem. Používá se například pro pozměnění vzhledu tlačítka, barvy odkazu apod., pokud se ale pod tento stav v rámci designu schová důležitá informace, mobilní telefony bez kurzoru nemají prakticky šanci se k informaci dostat (Appleseed, 2012).

Očividné je omezení v podobě velikosti obrazovky. S tím souvisí i velikost interaktivních prvků, malá a více nahuštěná tlačítka mohou dobře fungovat pod myší, kde lze přesně určit kam kliknout, ale pod větším prstem může často dojít k překliknutí. Mnohé chytré telefony také disponují slabší hardwarovou výbavou. Pokud se tedy na stránce nachází složité skripty, animace apod., prohlížení na slabším přístroji bude pomalé, nepříjemné a může způsobit odchod uživatele ke konkurenci (Appleseed, 2012).

Také psaní na mobilním telefonu nemusí vždy být příjemné a přesné, často dochází k překlepům a je dobré co nejvíce minimalizovat počet ručně vyplňovaných políček, například ve formulářích (Appleseed, 2012).

Pokud se uživatel připojuje na stránku ze zařízení mimo wifi síť a využívá tedy mobilní data, může se celé spojení značně zpomalit. Ne každý navíc využívá neomezené datové balíčky, a pokud se ze stránky budou stahovat například velké obrázky, datová náročnost může být pro uživatele neúnosná (Appleseed, 2012).

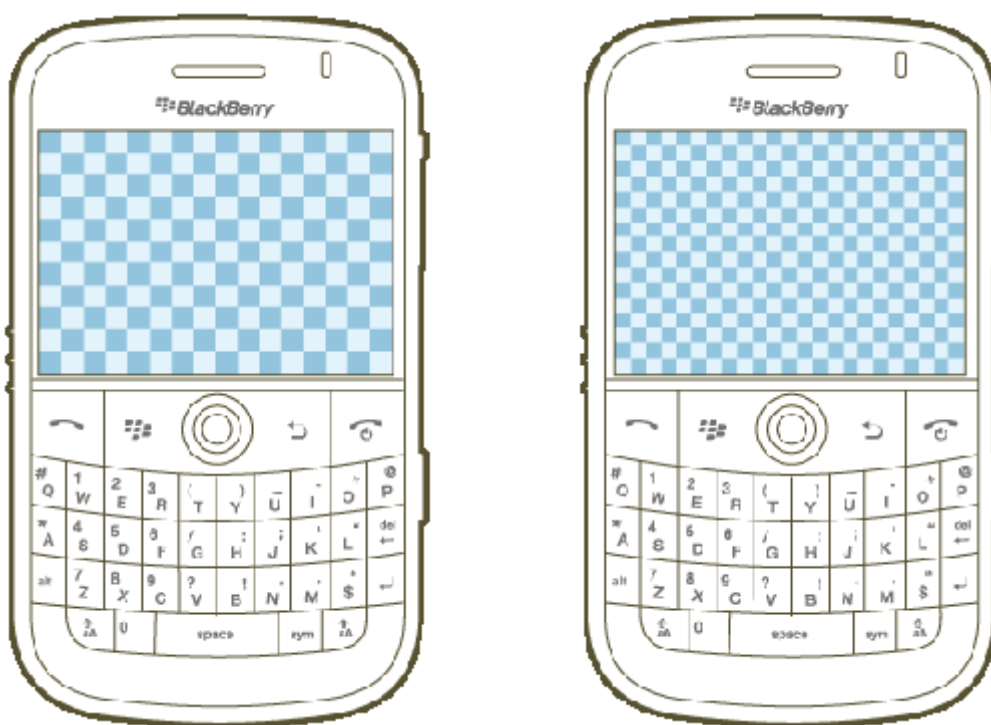
2.1.5 Metatag Viewport

Vedle rozlišení obrazovky je důležitým údajem i PPI (pixels per inch, počet pixelů na palec), tedy hustota obrazových bodů na displeji. Čím více je na obrazovce pixelů, tím ostřejší, ale menší, bude obraz. Problematika je popsána v článku na serveru Interval.cz. Problém nastal, když společnost Apple na trh přivedla tablet iPad Mini. Ten měl totiž stejný počet pixelů jako původní iPad s rozlišením 768×1024 a to i přesto, že obrazovka iPadu Mini je o 40 % menší. Pro vývojáře webů to zprvu znamenalo, že každá grafika, tlačítko či text na jejich stránkách se najednou zdály být velmi drobné. Nakonec vývojáři ze společnosti Apple vytvořili metatag `viewport`. Tag se velmi rychle uchytil a v současné době je podporovaný valnou většinou prohlížečů. Viewport (šířka průhledu) charakterizuje oblast, do které se vykreslí webová stránka, ať už se jedná o displej telefonu nebo okno prohlížeče na PC. Díky němu mohou na začátku HTML dokumentu tvůrci stránek deklarovat, že se mají stránky renderovat ne podle rozlišení přístroje, ale podle jeho fyzické velikosti (Pokorná, 2013).

Takový zápis by pak vypadal následovně:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Hodnota initial-scale charakterizuje měřítko, ve které se stránka zobrazí. Měřítko 1:1 je nejpoužívanějším a značí se právě číslem 1. Tímto atributem je možné nastavit maximální či minimální možné zvětšení obsahu, nebo ruční zvětšování úplně zakázat. Druhou možností, jak použít tag `viewport`, je zadat přímo šířku zobrazovaného obsahu například pomocí hodnoty `content="width=800"`.



Obr. 5: Dvě zařízení se stejně velkým displejem, přesto má displej vpravo čtyřnásobný počet pixelů (Pokorná, 2014)

2.2 Definice responzivního designu

V roce 2010 se objevil článek, ve kterém se webdesigner Ethan Marcotte zamýšlel nad požadavky svých klientů, jako dotvořit k původnímu webu firmy další, samostatný web výhradně pro iPhone. V článku došel k závěru, že nastala doba, kdy je potřeba přijmout změnu a definoval pojem responzivní webdesign: „*Tohle je naše cesta kupředu. Spíše než vytvářet design na míru pro každé z neustále se zvětšujícího počtu webových zařízení, můžeme je chápat jako aspekty stejného prožitku. Můžeme navrhnout optimální prohlížeč prožitek, ale vložit do něj technologie založené na*

standardech a učinit naše designy nejen flexibilními, ale více adaptivními k médiím, která je vykreslují. Ve zkratce, potřebujeme aplikovat responzivní webdesign.“ (přeloženo a citováno z MARCOTTE, 2010). Autor v článku dále na příkladu demonstruje techniky, díky kterým místo statické stránky vytvoří web přizpůsobující se šířce okna prohlížeče. Tyto techniky, principy responzivního designu, definoval tři:

- fluidní (plovoucí) layout
- fluidní obrázky
- CSS3 Media Queries (mediální dotazy)

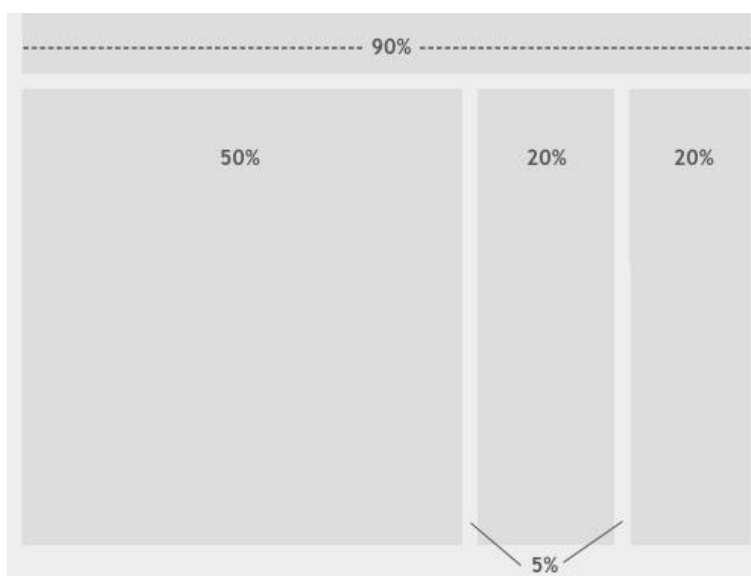
Specialista na webový front-end Martin Michálek na svém blogu zveřejnil článek, ve kterém označuje výše zmíněnou definici za „klasický“ responzivní design, protože „normální“ responzivní design se už v dnešní době musí potýkat s dalšími úkoly, které jeho aplikováním postupně vyplynuly. To znamená, že web se nestává responzivním po nasazení zmíněných tří principů. Být responzivní znamená přijmout nejrůznější nové postupy za účelem zajištění co nejlepšího prožitku z prohlížení jedné stránky na různých zařízeních a které se vždy nemusí týkat jenom šířky obrazovky (Michálek, 2014).

2.3 Fluidní layout

Dlouho bylo běžnou praxí zadávat veškeré rozměry na webových stránkách pomocí statických pixelů. Pokud se taková stránka zobrazí na menším rozlišení, než pro které je optimalizovaná, objeví se neoblíbený horizontální posuvník. Ve fluidním layoutu jsou velikosti všech prvků udávány relativně vzhledem k mateřskému prvku – v procentech. Pro převod fixních prvků na relativní se použije výpočet $\text{cíl/kontext} = \text{výsledek}$, kde cíl je vnitřní prvek, kontext vnější prvek a výsledek je relativní velikost vnitřního prvku vzhledem k vnějšímu v procentech (Svoboda, 2014). Převod v kódu pak vypadá následovně:

Fixní rozměry	Relativní rozměry
<pre>#obal { width:960px; } #levy_sloupec { width:360px; margin-right:40px; } #pravy_sloupec { width:200px; }</pre>	<pre>#obal { width:90%; } #levy_sloupec { width:37,5%;/*360 ÷ 960*/ margin-right:4,16%;/*40 ÷ 960*/} #pravy_sloupec { width:20,83%; }</pre>

Při správné aplikaci nezáleží, v jakém rozlišení si návštěvník stránku prohlíží, obsah se vždy přizpůsobí jeho obrazovce. Plovoucí layout v této své ryzí podobě ale zdaleka není dokonalý – na velmi velkých obrazovkách mohou být řádky s textem příliš dlouhé, na velmi malých zase příliš krátké. V souvislosti s procenty se často používá další relativní jednotka, jednotka em . Ta odpovídá momentální velikosti písma, pokud je tedy velikost písma nastavená na dvacet pixelů, jedna jednotka em se také rovná dvaceti pixelům. Výchozí velikost fontu je většinou prohlížečů nastavená na 16 pixelů.



Obr. 6: Plovoucí layout definovaný procenty (Knight, 2009)

2.4 Mediální dotazy

2.4.1 Media Types

Do CSS2 byl zaveden modul Media Types. Tento modul umožňuje podle několika kategorií identifikovat přístroj, ze kterého uživatel na stránku přistupuje. Seznam těchto typů je v následující tabulce 1.:

Media Type	Odpovídající přístroj
all	Všechna zařízení (výchozí)
braille	Zařízení pracující s Braillovým písmem
embossed	Tiskárny Braillova písma
handheld	Přenosná zařízení (malé obrazovky)
print	Tiskárny nebo náhled před tiskem
projection	Projektory
screen	Barevné počítačové obrazovky
speech	Syntežátory řeči
tty	Média používající neproporcionální znaky.
tv	Televizní zařízení

Tab. 1: Přehled všech mediálních typů (Kadlec, 2014, s. 65)

Ačkoliv jednotlivé typy vcelku pokrývají přístupová zařízení, používání všech možností se nikdy neuchytilo. V době příchodu tohoto modulu nebyla např. handheldová zařízení natolik rozšířená, aby vývojáři stránek definovali kód striktně pro mobilní zařízení. Vývojáři telefonů se tedy rozhodli využít typ `screen` a vydávat tak své přístroje za počítače, aby se jim dostávalo stejného obsahu (Kadlec, 2014, s. 66).

Reálně se tedy používají dva typy; `screen`, pro veškerá zařízení s obrazovkou, a `print`, který definuje styly načtené po iniciování tisku stránky. Tyto styly jsou tradičně ochuzené o reklamy, menu a všechny rušivé grafické prvky, které nejsou pro tisk textu ze stránky důležité. Media Types lze použít dvěma způsoby.

Prvním způsobem je nadefinování stylů přímo v CSS souboru pomocí pravidla

@media:

```
@media screen {
#menu { background:#0000; } /*Následuje zápis vlastností,
které se aplikují pouze na zařízení označené jako screen*/ }
@media print {
#menu { display:none; } /*Následuje zápis vlastností, které
se aplikují, pokud uživatel iniciuje tisk stránky*/ }
```

Druhý způsob se použije v případě, že má každý typ nadefinované své styly ve vlastním souboru. Do hlavičky HTML dokumentu se pak připojí následující notace:

```
<link rel="stylesheet" href="default.css" media="screen" />
<link rel="stylesheet" href="print.css" media="print" />
```

Defaultně se, pro všechna zařízení označená jako `screen`, použijí styly definované v souboru `default.css`. Pokud návštěvník iniciuje tisk stránky, načtou se styly ze souboru `print.css`.

2.4.2 Media Queries

S příchodem CSS3 se tento modul zdokonalil a vznikly Media Queries. „*Mediální dotazy nám umožňují zaměřit se ne pouze na určité třídy zařízení, ale na konkrétní fyzické charakteristiky přístroje, který zrovna vykresluje naši práci*“ (přeloženo a citováno z Marcotte, 2010). Mediální dotazy se skládají z mediálního typu, mediálního výrazu, logického klíčového slova a pravidla (Kadlec, 2014, s. 64).

Mediální výraz je jádro dotazu. Výrazy umožňují zeptat se právě používaného přístroje, zda splňuje nějakou konkrétní podmínku. Výsledek se vyhodnocuje jako `true` a `false`. Logická klíčová slova jsou nepovinnou součástí dotazů a dokáží je ještě více specifikovat. Například pomocí `and` je možné testovat současně více výrazů a pomocí `not` negovat výsledek celého výrazu. Dalším slovem je `or`, které je v zápisu reprezentováno čárkou mezi dvěma výrazy a které říká, že pro provedení pravidla stačí splnění alespoň jedné ze dvou uvedených podmínek. Seznam všech charakteristik, na které je možné se v rámci mediálních výrazů ptát, je uveden v příloze 1. Sloupeček Min/Max v tabulce charakterizuje, zda je možné ve spojení s daným výrazem použít prefix `min-` a `max-`.

Výsledek dotazu rozhodne o tom, zda se provede nebo neprovede napsané pravidlo. Jedná se jednoduše o klasická stylová CSS pravidla, která se aplikují pouze pro ta zařízení, která vyhoví definovaným výrazům. Uvedený příklad níže nejprve pomocí pravidla `@media` zkontroluje, zda se jedná o zařízení s barevnou obrazovkou a výraz `width` s prefixem `min-` testuje, zda je šířka viewportu zařízení alespoň 600 pixelů. Logické klíčové slovo `and` určuje, že se musí splnit obě podmínky, aby se aplikovalo pravidlo mezi složenými závorkami. Zde se konkrétně po splnění podmínek obarví všechny odkazy modrou barvou.

```
@media screen and (min-width: 600px) { a { color: blue; } }
```

Pomocí mediálních dotazů lze dovést flexibilní layouty téměř k dokonalosti. Zde se nabízí vytvářet pravidla pro známé a konkrétní hraniční rozměry displejů různých zařízení – v takovém místě se zavede mediální dotaz s pravidly, která pro všechny přístroje se stejnou nebo menší, respektive větší, šířkou nějakým způsobem upraví layout stránky. Bod, ve kterém dojde k takové podmíněné změně, se nazývá breakpoint. Tento přístup svádí k opětovnému rozdělení zařízení do tří kategorií, které, jak bylo vysvětleno v kapitole 2.1, není dostačující. Martin Michálek upozorňuje, že ač se na první pohled zdá možné identifikovat pomocí mediálních dotazů konkrétní typ zařízení, nemusí to být vždy spolehlivé. Jako příklad je v článku uvedený iPad, který při položení na výšku má stejnou šířku viewportu jako Nokia Lumia 800 položená na šířku (Michálek, 2013a).

Stejnou filozofii ve své knize vyznává i Tim Kadlec, který navíc čtenáře navádí k určování breakpointů ne podle známých zařízení, tedy nedoporučuje používat jakékoliv globální breakpointy, ale podle potřeby každého webu: *„Mnohem lepším přístupem je nechat samotný obsah, ať on sám nadiktuje, kde má být další breakpoint a kolik breakpointů celkem potřebuje. Začněte se zvětšováním okna prohlížeče a sledujte, kde se objeví nějaký prostor pro vylepšování. Když dovolíte, aby vás obsah vedl za ruku, ještě více uvolníte layout z těsných pout specifického rozlišení“* (citováno z Kadlec, 2013, s. 77). V tomto úseku je možné nalézt ještě další důležité autorovo doporučení, tedy vyvíjet responzivní stránku „směrem nahoru“.

2.4.3 Směr vývoje

Teoreticky je možné chopit se vytváření responzivního designu ze dvou stran. Oblíbeným způsobem se stalo vytvářet nejprve dokonalou verzi pro velká, desktopová rozlišení a poté pomocí postupné degradace ubírat a přerovnávat prvky pro menší zařízení a viewporty. Tento postup se nazývá Desktop First, tedy vývoj směrem dolů.

Již zmíněný druhý způsob se nazývá Mobile First, tedy vývoj směrem nahoru. Zde se nejprve vytvoří nejzákladnější podoba webu pro nejmenší rozlišení a pak se pomocí postupného vylepšování stránka testuje na větších a větších viewportech a podle potřeby se přidávají různé prvky a elementy. Podle Tima Kadlece je nejdůležitějším z důvodů, proč si vybrat budování stránky postupným vylepšováním skutečnost, že existují přístroje a prohlížeče, které zatím nepodporují mediální dotazy. Pokud uživatel skrze takový prohlížeč navštíví responzivní stránku budovanou postupnou degradací, nerozezná žádné dotazy a načte původně definovaný web pro velké desktopy, který pravděpodobně nebude schopný rozumně zobrazit. Na druhou stranu, pokud takový uživatel navštíví stránku budovanou technikou Mobile First, zobrazí se mu nejzákladnější sada prožitků sestavená pro nejmenší displeje, kterou bez problémů zvládne zpracovat každé zařízení (Kadlec, 2013, s. 73).

2.5 Fluidní obrázky

Poslední přísadou do klasického responzivního webdesignu jsou podle prvotní definice plovoucí obrázky. Ethan Marcotte ve svém původním článku demonstroval přizpůsobení obrázků plovoucím layoutům tím, že obrázek umístil do kontejneru, pro který nastavil požadovanou velikost v relativních jednotkách. Samotnému obrázku uvnitř kontejneru pak nadefinoval šířku na 100% (Marcotte, 2010). Problém takto nastavených obrázků je ve výkonu stránky; každé zařízení totiž stahuje původní velký soubor, který se pak na daném zařízení pouze rozměrově zmenší. Pro počítače nebo zařízení připojené na wifi nebo vysokorychlostní internet to není žádný problém, uživatelé s pomalejším mobilním připojením však pocítí delší odezvu a ztracená data (Kadlec, 2013, s. 93). Není problém vytvořit takový kód, ve kterém by se pro malé displeje jednoduše nezobrazoval. Zde láká v pravidlech

pro detekci malého displeje pod obrázkem nadefinovat `display: none`. Tento příkaz však obrázek pouze schová, prohlížeč ho i tak stáhne. Řešením může být nadefinovat základní sadu prožiteků, ve kterých se obrázek nevyskytuje a poté v pravidlech pro větší displeje obrázek teprve zavést. (Kadlec, 2013, s. 93)

2.5.1 Nové atributy elementu ``

Problém však dokáže vyřešit novinka v HTML5 v podobě nových atributů elementu ``, které ve svém článku pro server Zdroják podrobně popsal Martin Michálek. Jedná se o atributy `srcset`, který slouží k zadání všech variant obrázku a `sizes`, v němž se určí jejich velikosti v jednotlivých breakpointech. Zápis by vypadal například následovně:

```

```

Prohlížeči tím vývojář dává vědět, že má připravený obrázek `maly.png` pro viewporty o šířce alespoň 600 pixelů, `stredni.png` pro viewporty se šířkou alespoň 1024 pixelů a `velky.png` pro šířku alespoň 1600 pixelů. Do atributu `src` se pak vkládá nejmenší verze obrázku jako fallback pro prohlížeče, které by si s atributem `srcset` neporadily. (Michálek, 2015).

Veškeré rozhodování zůstane na prohlížeči, autor stránek pouze vygeneruje dostatek variant a správně je popíše pomocí deskriptorů. V příkladu výše je uveden deskriptor `w`; ten říká, jakou šířku v pixelech obrázek reálně má. Druhý deskriptor `x` určuje připravenost souboru s obrázkem pro různé `device-pixel-ratio` poměry, tedy pro zařízení s větší hustotou obrazových bodů. Například zápis níže říká, že soubor `image@2.jpg` má prohlížeč použít při `device-pixel-ratio` alespoň 2, pro menší hodnoty použije `image.jpg`.

```
<img ... srcset="obr.jpg, obr@2x.jpg 2x" >
```

Díky atributu `sizes` není nutné volit obrázek jen podle šířky okna, ale podle šířky elementu v rámci layoutu.

```

```

Pravidlo (min-width: 768px) 300px v zápisu výše říká, že pro rozlišení nad 768 pixelů má mít obrázek šířku 300 pixelů, hodnota "100vw" znamená, že pro všechna nevyhovující, tedy nižší, rozlišení bude mít obrázek 100 procent šířky viewportu (Michálek, 2015).

Tato novinka v HTML5 se rychle ujala a všechny prohlížeče oznámily, že tento standard naimplementují. Tato implementace však chvíli potrvá, prozatím je tedy pro správné fungování potřeba používat nějaký polyfill či fallback. Polyfill je speciální JavaScriptový kód, který dokáže nahradit chybějící funkci u starších verzí prohlížečů (Michálek, 2015).

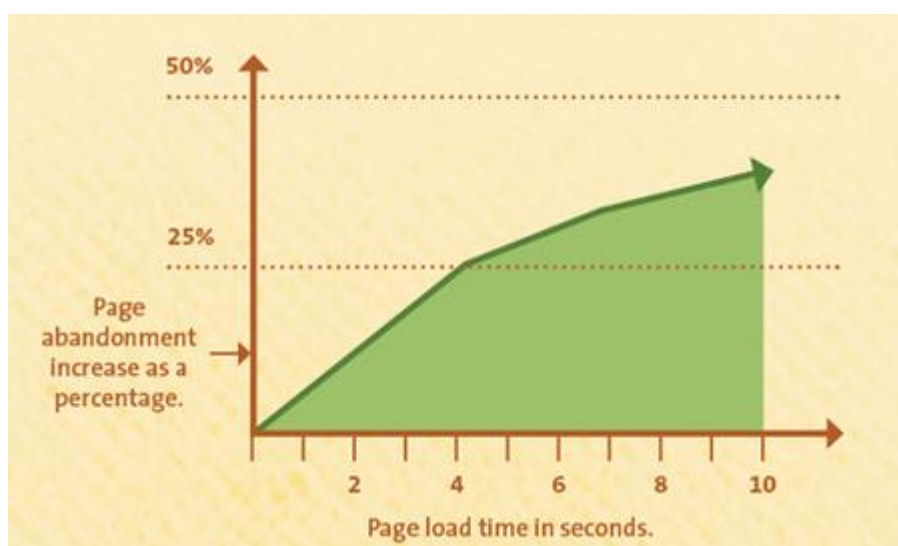
2.5.2 Formát SVG

Řešení pro bitmapovou grafiku na webu tedy konečně existuje, pokrývá většinu případů užití a není důvod ho nevyužívat. Za zmínku však také stojí formát SVG, neboli Scalable Vector Graphics. Chování obrázků v SVG se definuje v XML. To znamená, že mohou dobře měnit velikost, aniž by se nějak zvyšovala velikost samostatného souboru, a jsou programovatelné. Díky tomu je také umí vyhledávač Google indexovat (Stanescu, 2010).

Jedná se tedy o vektorový formát a je užitečný právě díky vlastnostem vektorové grafiky, která je tvořená čarami a plochami, a dá se tedy opakovaně a bez ztráty kvality zmenšovat a zvětšovat. Taková vlastnost se hodí například při problémech s displeji o větší hustotě obrazových bodů – když takový displej vykreslí vektor detailněji, bude pořád stejně kvalitní. Vadou na kráse jsou stále používané starší verze prohlížeče Internet Explorer a prohlížeče na Androidu verze 2.3 a nižší, které formát SVG nepodporují (Kadlec, 2013, s. 110). Pro tyto verze je nutné zavádět fallbacky a náskok ve výkonu, který SVG formát získal svou výraznou datovou nenáročností, se zde vrátí v podobě zbytečného stažení obrázku, který nakonec starší prohlížeč nezvládne zpracovat.

2.6 Výkon

Jak bylo na začátku práce řečeno, v dnešní době řeší responzivní design i další problémy kromě přizpůsobování rozměrů prvků na stránce. Jedním velmi důležitým je často opomíjený výkon webové stránky. Při prohlížení na stolním počítači se většinou taková věc příliš řešit nemusí, internetové připojení v domácnostech je v dnešní době velmi rychlé a většina ne příliš složitěho obsahu se načte vmžiku. Tato skutečnost ale pouze zvyšuje očekávání uživatelů mobilních zařízení, která mnohdy využívají omezená mobilní data a na načtení většiny stránek musí čekat. V současné době tedy optimalizovaná stránka, která dokáže zobrazit svůj obsah rychle a v malé velikosti, má oproti ostatním významnou výhodu. Na toto téma proběhl výzkum a jeho autoři vysledovali, že 56 % oslovených uživatelů očekává, že načítání stránek na jejich telefonu bude zhruba stejně rychlé, jako na PC. Celých 73 % respondentů uvedlo, že jsou schopni stránku opustit kvůli pomalejšímu načítání. Pokud se jedná o internetový obchod, 79 % respondentů uvedlo, že pokud by byli jakkoliv nespokojeni s jeho výkonem, už zde znovu nenakoupí. 47 % pak očekává, že se eshop načte do dvou sekund. Autoři výzkumu nakonec ze získaných dat vypočítali, že sekundové zpoždění při načítání stránky může znamenat sedmiprocentní redukci konverze. Pokud by byla uvažována internetová služba vydávající 100 000 dolarů za den, sekundové zpoždění může společnost stát 2,5 milionu dolarů za rok (Work, 2011).



Obr. 7: Graf zobrazující stoupající šanci na zavrnutí stránky z důvodů dlouhého načítání (Work, 2011)

Metod pro zrychlení webových stránek je mnoho. Jedním z nejefektivnějších způsobů je Minimalizace HTTP požadavků. Spočívá ve snížení stahovaných dat na co nejmenší počet. Je možné například sjednotit externí soubory; ať už se jedná o JavaScript nebo CSS, je vhodné umístit je vždy do jednoho souboru. Každý separovaný soubor totiž znamená pro prohlížeč další zbytečný požadavek. Velikost těchto externích souborů je ještě možné snížit pomocí takzvané minifikace. Ta spočívá v tom, že odstraní mezery a tabulátory ze zdrojových souborů, tedy všechna volná místa. Taková úprava dokáže soubor zmenšit o několik desítek procent. Ještě většího snížení lze dosáhnout pomocí obfuskace, která převede soubor na jednoduchý zápis se zapojením jednoduchých identifikátorů. Tato metoda je však vcelku chybová a ve výsledné změti nicneřikajících znaků je chyby složité opravit. Pro tyto metody se používají různé externí nástroje (Maňák, 2010).

Dalším způsobem je využívání ukládání do mezipaměti, tedy cachování. Běžně se stává, že po prvním načtení stránky se při dalších návštěvách načítání zrychlí. Je tomu tak proto, že se data při první návštěvě uložila do mezipaměti prohlížeče a při opětovné návštěvě se nemusela znovu stahovat. Pro správné ukládání a aktualizování mezipaměti je nutné správně nastavit hlavičku požadavků. Ty se poté využívají při komunikaci pro ověření platnosti dat přes http. Pokud nejsou hlavičky nastavené, uložení dat bude pouze v rámci jednoho sezení. Po úpravě hlaviček je potřeba správně nastavit webový server s využitím specializovaných modulů (Maňák, 2010).

Protokol HTTP 1.1 dovoluje přenášet data v komprimované podobě. Na webovém serveru se nejprve obsah zkomprimuje a až poté je odeslán prohlížeči ke zpracování. Tato metoda dokáže snížit velikosti souborů až o desítky KB, což je výhodné především pro mobilní weby. Na obrázky se tato komprese nevztahuje (Maňák, 2010).

Pro testování výkonu stránek existuje několik zdarma dostupných nástrojů, které dokáží podrobně ohodnotit jejich výkon. Velmi komplexním pomocníkem je stránka webpagetest.org. Tento projekt disponuje několika stroji po celém světě, na kterých jsou nainstalované různé prohlížeče různých verzí. Pro test pak tedy stačí zadat URL

adresu požadovaného webu, vybrat lokaci a prohlížeč, ve kterém má test proběhnout. K dispozici jsou i další nastavení, kde lze zvolit rychlost připojení počítače, ze kterého se má test provést, kolik testů se má provést apod. Speciálními funkcemi je i možnost provést test v mobilním prohlížeči a z mobilní sítě nebo otestovat několik stránek najednou a porovnat jejich rychlost načtení. Po veškerém nastavení stačí stisknout tlačítko a za krátkou dobu se objeví výsledky testu. Základním hodnocením je šest známek, kde každá hodnotí jedno z (podle provozovatelů) majoritních hledisek úspěšně optimalizovaného webu.

První, pojmenovaná jako „First Byte Time“, hodnotí dobu mezi prvním http požadavkem od uživatele a doručení prvního bytu stránky prohlížeči (WebPageTest Documentation).

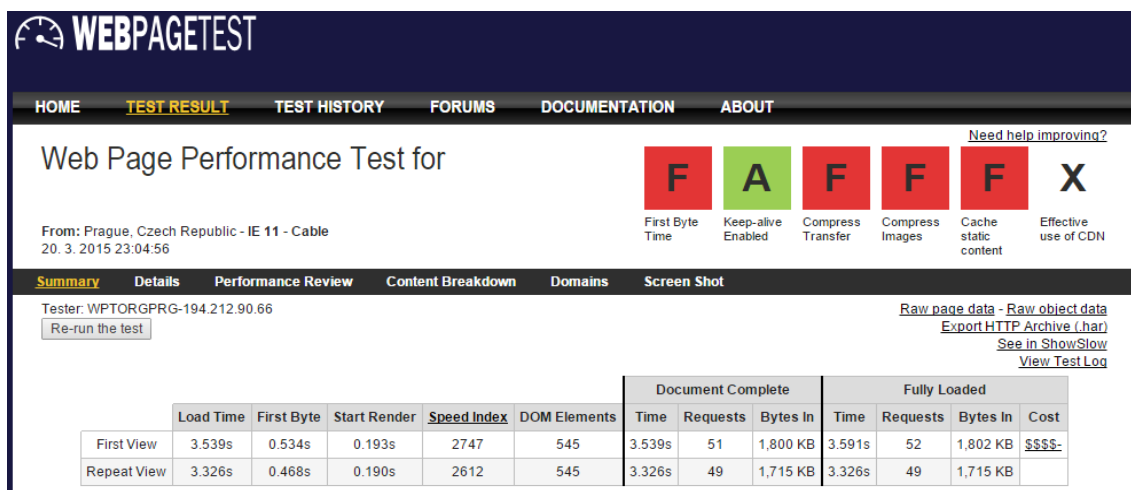
Druhá kontroluje, zda je na stránce aktivní „Keep-alive“ funkce. Pro každý kus obsahu na stránce, ať už se jedná o obrázek, JavaScript nebo CSS, který musí prohlížeč návštěvníka stáhnout, je potřeba požadavek na spojení s web serverem. Vytváření nového spojení může zabrat nějaký čas a stránku zpomalovat, funkce Keep-alive pak zajišťuje, že se místo vytváření nových znovu používají spojení již vytvořená. Funkce bývá ve výchozím stavu zapnutá, v nějakých situacích ale nemusí fungovat správně. Její konfigurace se provádí na straně serveru a nevyžaduje žádné změny v kódu stránky (WebPageTest Documentation).

Další dvě známky hodnotí míru komprese využívané na stránce. Kontroluje, zda je všechn text na stránce zasílán prohlížeči v komprimované podobě a zda nejsou obrázky pro zobrazení na webu zbytečně kvalitní nebo neobsahují-li nějaká přídavná data, například informace o nastavení fotoaparátu kterým byla fotka pořízená (WebPageTest Documentation).

Pátá známka hodnotí míru dat na stránce, která se nemění a mohou se tak uložit do paměti cache. Návštěvník pak při další návštěvě, pokud před ní tuto paměť nevymaže, nemusí stahovat všechny soubory znovu a stránka se mu načte rychleji (WebPageTest Documentation).

Poslední známka představuje efektivnost využití CDN (Content Distribution Network). CDN je síť serverů rozmístěných různě po světě a každý z nich je připravený dodat uživateli stejný obsah. Smysl pak spočívá v tom, že se aktivuje vždy server nejbližší uživateli, požadavky na přístup k obsahu tak nemusí putovat tak dlouho, jako kdyby byl používán pouze jeden statický server. Tato služba se vyplatí hlavně pokud provozovatel ví, že jeho návštěvníci přistupují z mnoha různě vzdálených míst (WebPageTest Documentation).

Kromě těchto známek služba podrobně zobrazuje například trvání kompletního načtení stránky, velikosti stahovaných souborů a jejich vliv na rychlost nebo počet uskutečněných požadavků. Rozlišuje také trvání načtení při první návštěvě s čistou cache pamětí a bez cookies a druhé načtení, kde jsou tyto paměti využity.



Obr. 8: Znamky a hodnocení udílené službou WebPageTest (WebPageTest Documentation)

Speciální nástroj pojmenovaný PageSpeed Insights vznikl i pod záštitou Google Developers. Služba se zaměřuje na měření rychlosti načítání stránek, a to zvláště na mobilním telefonu a počítači. U testu na mobilním zařízení zobrazí po proběhnutí analýze její ohodnocení ve dvou kategoriích; v první se zaměřuje na rychlost a ve druhé na uživatelský dojem, kde zohledňuje například velikost prvků pro pohodlné klepnutí u zobrazení na mobilních telefonech. U obou kategorií ukáže problémy i s doporučeným řešením. V testu pro počítač pak zobrazí souhrn návrhů pro zlepšení výkonu stránky, taktéž s navrženým postupem opravy.

Obr. 9: Problémy a jejich doporučená řešení nalezené službou PageSpeed Insights (Google Developers, 2015)

2.7 Responzivní design vs. Mobilní web

Z celé práce tedy vyplývá, že mobilní zařízení není možné ignorovat a jim doručovaný obsah je potřeba přizpůsobit. Otázka může být, jestli vážně využít responzivní design, nebo se uchýlit k separovanému mobilnímu webu. Třetí možností by mohlo být vytvoření mobilní aplikace. Největší výhodou tohoto řešení je možnost poskytovat uživatelům informace i v offline režimu, náklady na takovou aplikaci však bývají obrovské a veškerá údržba by se musela provádět skrze aktualizace celé aplikace, kterou si však uživatel nemusí stáhnout. Navíc by pro nejširší distribuci bylo nutné aplikaci upravit, aby vyhovovala všem zařízením napříč operačními systémy. Tato možnost je tedy jen pro velmi specifické projekty.

2.7.1 Mobilní web

Vytvořit samostatný web může být pro nějaké charakteristické projekty stále výhodnější. Tím, že se vývojář zaměří na tvorbu čistě mobilního webu, může využít všech technických vymožeností, které mobilní zařízení nabízejí, a definovat tak velmi specifické prožitky. Takto postavený web pak může například pracovat s aktuální polohou čtenáře. Pro takové weby většinou odpadá i problém s optimalizací, jelikož je web definován pouze pro mobilní zařízení, všechen

mediální obsah je patřičně zmenšený a celý web tak dosahuje malého datového objemu. To znamená, že pracuje rychle i na velmi pomalých sítích, se kterými se uživatel lehce setká mimo větší města (Ghazarian, 2014). Na druhou stranu jsou nevýhodou dvě domény – jedna mobilní a jedna pro desktopovou verzi. To znamená udržovat obsah na dvou stránkách, zajišťovat SEO pro dvě stránky, apod.

Hlavním mínusem je však již zmíněný problém s diverzitou mobilních zařízení, pro mobilní web by se tak jen těžko stanovovaly univerzální statické rozměry. Tato rozmanitost by se dala vyřešit poměrně jednoduše – mobilní web a responzivní přístup se totiž nemusí vylučovat - je možné obě metody zkombinovat a zavést responzivní mobilní web.

2.7.2 Responzivní design

Výhody responzivního designu podle (Ghazarian, 2014) spočívají v jednodušší údržbě. Opět je vše pod jednou doménou a obsah stačí psát pouze jednou, i SEO a vůbec marketing je v tomto podání jednodušší. Jeden web pro všechna zařízení schopná prohlížet internet však nedokáže využít technický potenciál mobilních telefonů či tabletů.

Velkou nevýhodou responzivního designu je jeho závislost na nejnovějších webových technologiích, které nemusejí podporovat stále velmi používané starší verze prohlížečů. Velmi početná základna příznivců této techniky však pro většinu takových problémů vytvořila nejrůznější fallbacky a pollyfily.

Oba přístupy mají své výhody a nevýhody. Vždy budou projekty, pro které bude výhodnější vytvořit separovaný web, responzivní webdesign má však velký potenciál k tomu, aby takovou stránku ve většině bodů předčil. Rozhodnutí tedy musí padnout na základě charakteru vytvářené stránky.

3 Příklady implementace a jejich porovnání

V této kapitole proběhne srovnání několika vybraných webových stránek v jedné kategorii.

Reprezentace jednotlivých měst na internetu je určitě důležitá; prostřednictvím svého webu tak svým občanům nebo návštěvníkům mohou jednoduše doručovat nejnovější informace, novinky, vyhlášky či jiné různé články o dění ve městě, zajímavá turistická místa apod. Takový zdroj informací by měl být co nejlépe dostupný všem čtenářům bez ohledu na přístroj, z něhož k němu přistupují, a to, jak už bylo v práci několikrát uvedeno, znamená stránky nějakým způsobem optimalizovat pro mobilní zařízení.

Pro tento test bylo vybráno 20 největších měst v České republice a proběhl průzkum jejich webových prezentací. V tabulce na začátku kapitoly 3.2 je zobrazeno, která města investovala do svých stránek a nějakým způsobem je optimalizovala pro mobilní zařízení, a která nikoliv. Tyto optimalizované stránky byly dále podrobeny podrobnějšímu testu podle kritérií stanovených v kapitole 3.1.

3.1 Stanovení kritérií

Základní rozdělení testovaných stránek poskytne již tabulka v kapitole 3.2, ve které je zaznamenáno, zda město využívá responzivní design nebo separátní mobilní web. U mobilních webů bude dále testována hlavně jejich rychlost, ve své podstatě by taková stránka měla dosahovat mnohem lepších výsledků než ta responzivní.

U webů aplikujících responzivní design bude také nejprve proveden test v obou zmíněných nástrojích, ve WebPageTest.org a v PageSpeed Insights, jejichž výsledky budou podrobněji popsány. Každá stránka bude podrobena dvěma testům, jedním se bude testovat přístup z desktopu za použití kabelového připojení a prohlížeče Google Chrome, druhý bude emulovat přístup přes mobilní prohlížeč a ze 3G sítě. Dále budou ohodnocovány jednotlivé principy responzivního designu (počet breakpointů, použité jednotky,...) a použité speciální techniky a nástroje (SVG formát, frameworky, JavaScriptové knihovny,...).

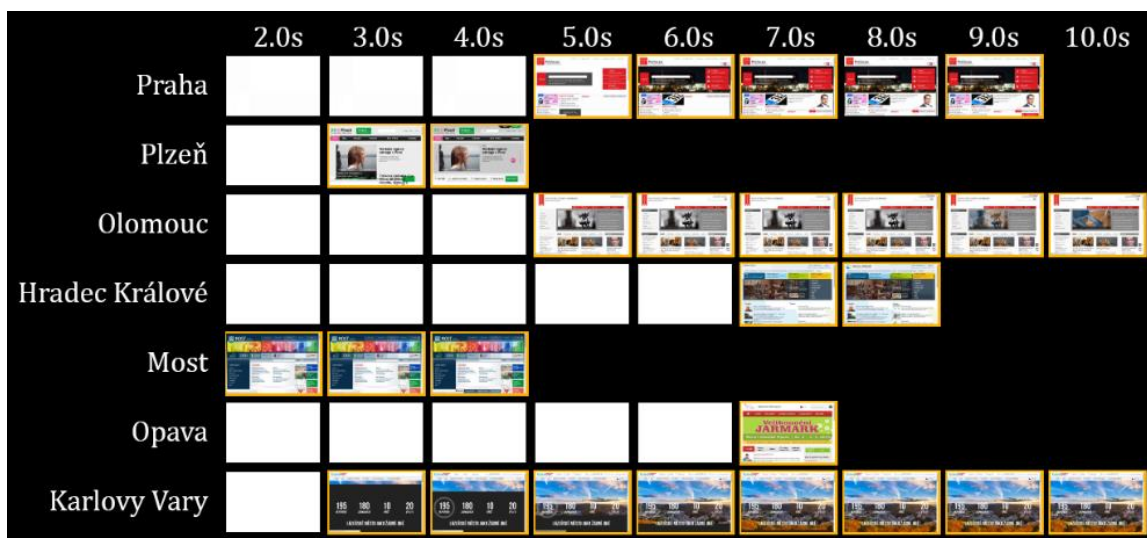
3.2 Testování vybraných stránek

U každého města v tabulce níže je vyznačeno, jakým způsobem je jeho web optimalizovaný pro mobilní zařízení. Pokud disponuje separátním webem na speciální doméně, je označen ve sloupci MW. Pokud nějakým způsobem obsluhuje všechna zařízení pod jednou doménou, je označen ve sloupci RD. Města bez označení nijak neřeší přizpůsobování stránek pro mobilní zařízení.

Město	MW	RD	Město	MW	RD
Praha		●	Pardubice		
Brno			Havířov		
Ostrava			Zlín		
Plzeň		●	Kladno		
Liberec			Most	●	
Olomouc		●	Karviná		
Ústí nad Labem			Opava		●
Hradec Králové	●		Frýdek-Místek		
České Budějovice			Karlovy Vary		●

Tab. 2: Seznam vybraných měst k testování (Vlastní zpracování)

Jako první byl proveden hromadný test rychlosti načtení desktopových verzí webu. Jak je vidět na obrázku níže, nejlépe v testu dopadlo město Most. Městský web se začal vykreslovat už dvě sekundy od odeslání požadavku a zobrazilo se hned 98 % obsahu. O dvě sekundy později byl načtený celý. Stránky Hradce Králové a Opavy začaly odpovídat nejpozději, až 2 sekundy po kompletním načtení stránek Mostu a Plzně, která obsadila druhé místo. Stránky Olomouce a Karlových Varů svůj obsah načítaly nejdéle, Karlovy Vary byly kompletně načtené ve třinácté sekundě, zatímco Olomouc na sebe nechal čekat 15 sekund. Všechny údaje v testu byly pro snazší vzájemné porovnání zaokrouhlené na jednotky sekund. Přesné časy budou probrány u jednotlivých rozborů.



Obr. 10: Přibližná rychlost načtení stránek jednotlivých měst (Vlastní zpracování)

3.2.1 Praha

Desktopové verzi webu www.praha.eu trvalo 0,274 sekund odpovědět na požadavek a poskytnout ke stažení první bajt; od služby webpagetest.org si tak za toto kritérium vysloužil známku B. Veškerý obsah se stáhl za 3,660 sekund, úplného načtení ale stránka dosáhla až o sekundu později.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 5)	3.660s	0.274s	0.000s	1074	3.660s	63	1,198 KB	4.503s	66	1,291 KB	\$\$\$--
Repeat View (Run 1)	1.381s	0.203s	0.000s	1074	1.381s	5	140 KB	1.464s	6	144 KB	

Obr. 11: Jednotlivé časy načítání webu www.praha.eu (Vlastní zpracování)

Pro načtení stránky bylo potřeba učinit 66 requestů. 63,5 % z nich si vyžádaly obrázky, 31,7 % pak JavaScript. Velikost stránky je celkem 1 291 KB z čehož naopak nejvíce, přesně 51,7 %, patří JavaScriptu a 37,2 % obrázkům. Po opětovném načtení, s využitím cookies a paměti cache, bylo potřeba pouze šesti requestů na 144 KB dat. Stránka se pak načítala 1,464 sekundy, rozdíl je tedy asi ve dvou sekundách.



Obr. 12: Podíl obsahu na www.praha.eu pro desktop (Vlastní zpracování)

Podle testovací služby stránka dostatečně nevyužívá komprimovaný přenos textových dat a hodnotí ho tak známkou F. Na rozdíl od toho kompresi obrázků známkou C, až na několik výjimek se tedy dá říci, že obrázky na stránce nejsou pro zobrazení na webu zbytečně veliké, nedá se tedy počítat s tím, že by obrázky byly optimalizované pro Retina displeje. Využití paměti cache je opět oznámkováno jako F, hlavní problém vidí služba v tom, že pro zhruba půlku obsahu je nastavená expirace obsahu jenom na 24 hodin.

Mobilní verze webu poskytla první bajt ke stažení za 3,5 sekundy, stránka se ale plně načetla za 10,659 sekund. Oproti desktopové verzi se zmenšil počet requestů na 52, stahovat se ale muselo 1 418 KB, což je ještě více než při desktopové verzi webu. Celých 73,8 % celkové velikosti náleží JavaScriptu. Obrázků je na mobilní verzi dokonce tak málo, že dosahují menší velikosti než CSS soubory. I služba PageSpeed Insights doporučuje pro obě verze webu hlavně omezit JavaScriptu, využít načítání do mezipaměti a aktivovat kompresi.



Obr. 13: Podíl obsahu na www.praha.eu pro handheld (Vlastní zpracování)

Desktopový web prakticky vůbec neaplikuje zásady responzivního designu, ze samotného prohlížení je patrné, že veškeré rozměry jsou definovány statickými jednotkami, a tak se při zmenšování viewportu uživatel dočká horizontálního posuvníku. Při rapidnějším zmenšení viewportu dokonce ani nedojde k přepnutí na mobilní web, na ten vede až odkaz v patičce stránky. Web tedy provede pouze počítačnickou detekci zařízení a po celou dobu zobrazuje jednu jeho verzi. Tuto svou volbu ukládá do mezipaměti a tak se při opětovné návštěvě detekce neprovede, ale zobrazí se ta verze webu, ze které uživatel při poslední návštěvě odešel.

Po přepnutí na mobilní verzi se objeví radikálně pozměněná stránka, ovšem pod stejnou doménou. Tato verze je vytvářena v relativních jednotkách a tak se do jisté míry prvky na obrazovce přeskupují a zmenšují či zvětšují. Web však nemá nastavené žádné breakpoints, pouze se přizpůsobuje. Zbytečným prvkem na této stránce je na první pohled stav hover, tedy chování prvku po najetí myši. Všechna tlačítka a fotky či novinky mají tento stav definovaný, i když účelem webu je obsloužit většinou dotyková zařízení bez kurzoru.



Obr. 14: Obě verze stránek www.praha.eu (Vlastní zpracování)

Po nahlédnutí do zdrojového kódu stránky je vidět, že přepínání mobilního a desktopového webu je řešeno pomocí dvou souborů index. Jednou je to index.html pro desktopovou verzi a jednou index.xhtml pro mobilní verzi psanou v XHTML 1.0

Transitional. Každý index má samozřejmě napojený svůj vlastní soubor se styly, takže nejsou využívány žádné mediální dotazy.

Celé řešení je na půlce cesty mezi separovanou mobilní stránkou a responzivní designem. Jedná se o dva rozdílné weby, které jsou však násilím přivedeny pod jednu doménu. V tomto stavu by bylo výhodnější a rychlejší, kdyby mobilní verze byla pod svou vlastní doménou. Optimalizace výkonu zde neproběhla téměř žádná, například i tento mobilní web má ve zdrojovém kódu nalinkované všechny JavaScriptové soubory desktopové verze a i když je vůbec nevyužívá, uživatel je pokaždé stahuje. A ve výše provedeném testu to byl právě JavaScript, který nejvíce brzdil načítání stránky.

3.2.2 Plzeň

V hromadném testu se stránky města Plzeň umístily oproti ostatním velmi dobře. Při prohlížení na desktopu se stránka načetla za 3,867 sekundy, první bajt ke stažení ale poskytla už za 0,284 sekund od odeslání požadavku, vykreslování stránky tedy začalo prakticky okamžitě. Od služby webpagetest.org si tak web za tuto kategorii vysloužil známku B.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 4)	3.867s	0.284s	0.000s	1307	3.867s	78	1.600 KB	3.909s	79	1.601 KB	\$\$\$\$-
Repeat View (Run 4)	0.669s	0.190s	0.000s	1307	0.669s	4	70 KB	0.669s	4	70 KB	

Obr. 15: Jednotlivé časy načítání webu www.plzen.eu (Vlastní zpracování)

Pro zobrazení stránky bylo zapotřebí vznést 79 requestů, 80,8 % se jich týká obrázků a pouze 9 % JavaScriptu. Velikost stránek je 1 601 KB, nejvíce, přesně 77 %, opět zabírají obrázky, na druhém místě jsou JavaScriptové soubory s 11,5 %. Při opětovném navštívení stránky se zapnutými cookies a pamětí cache se stránka načte za 0,669 sekund a je pro její zobrazení potřeba poslat pouze 4 requesty a stáhnout 70 KB dat.



Obr. 16: Podíl obsahu na www.plzen.eu pro desktop (Vlastní zpracování)

Podle testovací služby posílá web komprimovaná textová data pouze ze 45 %, stejně tak míra komprese obrázků v testu příliš neobstála, tam prošlo pouze 21% obrázků. Další špatně hodnocenou kategorií je využívání paměti cache, většina obsahu je podle služby ukládána špatně, nebo vůbec.

Při použití mobilního emulátoru jsou výsledky horší. Čas, za který se stránky na mobilním zařízení zcela zobrazily, se vyšplhal na 11 sekund, první bajt byl poskytnutý ke stažení za 1,157 sekundy a bylo potřeba odeslat 71 requestů. Mobilní telefon musí k zobrazení stáhnout 1 835 KB dat, z čehož 76,9 % jsou obrázky a 11,5 % JavaScriptové soubory. Služba PageSpeed Insights také jako největší problémy vyhodnotila nevyužití načítání do mezipaměti, špatně optimalizované obrázky a neaktivovanou kompresi.



Obr. 17: Podíl obsahu na www.plzen.eu pro handheld (Vlastní zpracování)

V případě stránek www.plzen.eu se podle úvodní tabulky skutečně jedná o responzivní webdesign. Při změně okna prohlížeče se jednotlivé prvky mění a je patrné, že je design postavený na relativních jednotkách. Celkově je na titulní stránce definováno pět mediálních dotazů; ten největší breakpoint pak přichází při

šířce 768 pixelů, kdy se hlavní menu omezí na čtyři položky a zbytek se schová pod jednu z nich. Stejně tak sekundární menu, které při větších rozlišeních funguje jako rychlé přepínání karet nad jedním blokem textu, při tomto rozměru úplně zmizí a jednotlivé karty zobrazí pod sebou. Při menších a menších rozlišeních se jednotlivé prvky ještě zmenšují, některá tlačítka opustí svůj vzhled a přepnou se pouze do textové, méně prostorově náročné, podoby. Celkově však web žádné informace na titulní straně nezapouzdruje a dodává uživateli mobilních zařízení stejný obsah, i když to znamená zobrazit nepěkné sloupce o dvou slovech na řádku. Z připojených CSS stylů lze vyčíst, že web je stavěný technikou Desktop First, dochází tedy k postupné degradaci.



Obr. 18: Zobrazení stránek www.plzen.eu na různých rozlišeních (Vlastní zpracování)

Dá se říci, že tato implementace responzivního designu splňuje všechny hlavní principy. Bohužel ale podrobněji neřeší otázku výkonu, obrázky se tedy sice správně přizpůsobují, ale i na nejmenším displeji se stáhne stejně velký soubor jako na desktopu. Stejně tak pro různé ikony jsou použity obrázky ve formátu JPEG, které jsou přizpůsobené tak, aby se kvalitně zobrazily na větších rozlišeních. Použití SVG formátu by zde bylo lepší volbou. Pro mobilní rozměry je tak web opravdu pouze stlačený; nejde o žádnou velkou redukci, což je ve výsledku poznat na množství stahovaných dat a rychlosti načítání.

3.2.3 Olomouc

Stránky www.olomouc.eu dopadly v prvním hromadném testu nejhůře. Celkový čas potřebný ke stažení a načtení celé stránky je 16,255 sekund. Přitom první bajt je serverem poskytnutý už jednu sekundu od odeslání požadavku.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 4)	15.725s	1.018s	0.000s	1894	15.725s	288	2,156 KB	16.255s	289	2,161 KB	\$\$\$\$\$
Repeat View (Run 1)	2.055s	0.898s	0.000s	1895	2.055s	9	97 KB	2.625s	10	97 KB	

Obr. 19: Jednotlivé časy načítání webu www.olomouc.eu (Vlastní zpracování)

Pro stažení celé stránky je potřeba odeslat 289 requestů. To je oproti ostatním testovaným webům obří číslo. 85,4 % je jich potřeba pro obrázky, na druhé pozici s osmi procenty překvapivě nejsou JavaScriptové soubory ale CSS dokumenty. Celkově je ke stažení 2 161 KB dat, z toho 79,3 % jsou obrázky a 14 % JavaScript. Při opětovné návštěvě s aktivovanou mezipamětí je potřeba pouze 10 requestů a 97 KB dat. Stránka se pak načte za 2,625 sekundy.



Obr. 20: Podíl obsahu na www.olomouc.eu pro desktop (Vlastní zpracování)

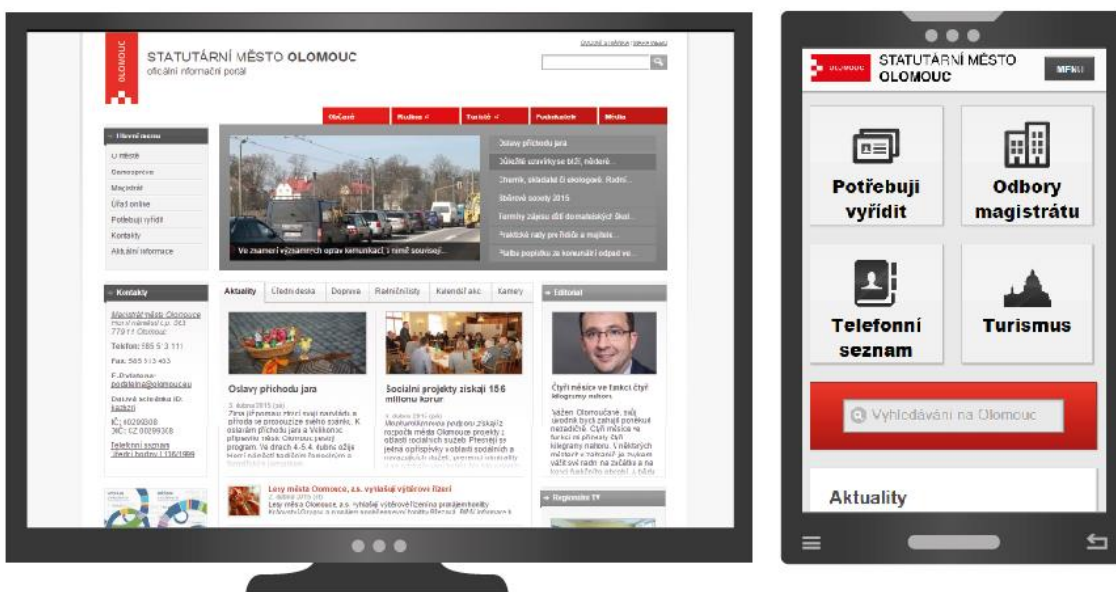
Naopak v ostatních kategoriích stránky Olomouce bodují; podle testovací služby je využití komprimovaného přenosu dat a komprese obrázků zvládnuté na jedničku a, ostatně jak lze poznat z rozdílu rychlostí načtení při prvním a druhém příchodu na stránku, i mezipaměť je na stránce dobře využívána.

Emulátor mobilního zařízení musel na poskytnutí prvního bajtu čekat jenom lehce přes jednu sekundu; načtení celé stránky o velikosti 645 KB pak trvalo 6,980 sekund. V počtu requestů je velký rozdíl, pro mobilní verzi webu je jich potřeba pouze 53. Podle PageSpeed Insights jsou největším problémem některé JavaScriptové kódy, které brání rychlejšímu vykreslení stránky.



Obr. 21: Podíl obsahu na www.olomouc.eu pro handheld (Vlastní zpracování)

Řešení stránek je podobné tomu pražskému; desktopový web je definovaný staticky a nijak nereaguje na změnu viewportu. V patičce webu pak vede odkaz na přepnutí na mobilní verzi, která je však opět pod stejnou doménou. Všechny prvky mobilní verze jsou definovány relativně a design je velmi odlehčený. Při prohlížení mobilní verze na velké obrazovce je na první pohled vidět, že pro krásně ostře zobrazené ikony v menu je použitý formát SVG.



Obr. 22: Obě verze stránek www.olomouc.eu (Vlastní zpracování)

Řešení je opět na půli cesty, mobilní web je separovaný od desktopového, ale pořád existuje pod stejnou doménou. V tomto případě ale funguje lépe a rychleji než v případě Prahy, mobilní web je optimalizovanější a i když pořád používá mnoho JavaScriptu, který celé fungování brzdí, není zde jeho množství tak markantní.

3.2.4 Hradec Králové

Stránky Hradce Králové jsou první, které využívají mobilní web na separované doméně. Co se týče desktopového webu, načítal se za 6,278 sekund, první bajt byl poskytnutý ve druhé sekundě a bylo potřeba stáhnout 1 813 KB dat přes 83 requestů. Obrázkům patří 63,4 % requestů a 71,6% bajtů, na druhém místě JavaScript s 26,6 % requestů a 21,3 % bajtů. Při opakovaném načtení s mezipamětí se stránka načte za 2,525 sekundy a je potřeba stáhnout 87 KB dat pomocí pěti requestů.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 5)	6.194s	1.830s	0.000s	438	6.194s	82	1,811 KB	6.278s	83	1,813 KB	\$\$\$\$-
Repeat View (Run 1)	2.475s	1.961s	0.000s	438	2.475s	4	87 KB	2.525s	5	87 KB	

Obr. 23: Jednotlivé časy načítání webu www.hradeckralove.org (Vlastní zpracování)

Ostatní kategorie hodnotí testovací služba známkou F, stránka tedy dostatečně nevyužívá komprimovaného přenosu textových dat ani kompresi obrázků, stejně tak využití mezipaměti je podle testu nedostatečné.



Obr. 24: Podíl obsahu na www.hradeckralove.org pro desktop (Vlastní zpracování)

Mobilní web na adrese m.hradeckralove.org se načítal až za 9,504 sekund, příčinou může být dlouhé čekání na první bajt, který byl poskytnut ke stažení až 2 sekundy od požadavku. K načtení bylo zapotřebí 43 requestů a velikost stránky je 1 433 KB, to není moc velký rozdíl od desktopové verze. Obrázkům patří 64,9 % z celkové velikosti a JavaScriptu 28,5 %. Kromě čekání na první bajt tedy bude načtení nejvíce brzdit stahování velkého množství tohoto datově náročného obsahu. Druhá testovací služba od Google Developers doporučuje hlavně eliminaci nadbytečného JavaScript a CSS kódu, optimalizaci obrázků a lepší využití komprese a mezipaměti prohlížeče. Zajímavé je doporučení vyhnout se přesměrování ze vstupní stránky;

test totiž zaznamenal, že při zadání adresy hradeckralove.org na mobilním zařízení proběhne přesměrování na www.hradeckralove.org, pak na m.hradeckralove.org/index.php a až potom na m.hradeckralove.org, kde začne vykreslování.



Obr. 25: Podíl obsahu na www.hradeckralove.org pro handheld (Vlastní zpracování)

Při nahlédnutí do zdrojového kódu mobilní verze stránek je vidět velké množství připojených JavaScriptů, rychlejšího načítání by se tedy dalo docílit omezením nějakých pokročilých funkcí, které zprostředkovávají. Více ale stránku skutečně brzdí obrázky, které jsou dodávány ve stejné kvalitě, jako na desktopové verzi webu.



Obr. 26: Obě verze stránek www.hradeckralove.org (Vlastní zpracování)

Separovaný mobilní web by měl disponovat vysokou optimalizací a rychlostí, na rozdíl od responzivního webdesignu například nemusí řešit otázku zobrazování více verzí jednoho obrázku pro různá rozlišení. Web by měl být jednodušší nejen

z hlediska vzhledu, ale také obsahu; je potřeba zamyslet se, zda mobilní návštěvníci skutečně využijí veškeré vymoženosti moderního webu. Takto špatně optimalizovaná stránka potom nevyužívá svého potenciálu a například olomoucké polovičaté řešení nakonec dosahuje lepších výsledků.

3.2.5 Most

Webové stránky města Most se v prvním srovnání umístily, spolu s Plzní, nejlépe. Kompletní načtení trvá 1,725 sekund, to je zatím bezkonkurenčně nejlepší čas mezi testovacími subjekty. Stránka má pouhých 412 KB a pro její stažení je potřeba 95 requestů. Nejvíce, přesně 81,9 %, requestů patří opět obrázkům, na druhém místě je ale s 12,8 % CSS, JavaScript je až daleko za ním se čtyřmi procenty. 66% podíl na velikosti stránky mají také obrázky, na druhém místě je tentokrát HTML s 19,4 %, JavaScript je zde až poslední. Při návratu se zapnutou mezipamětí se stránka načte za 0,897 sekund a je potřeba odeslat 37 requestů na 275 KB dat.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 4)	1.655s	0.227s	0.000s	1105	1.655s	92	410 KB	1.725s	95	412 KB	\$---
Repeat View (Run 2)	0.829s	0.202s	0.000s	1105	0.829s	36	275 KB	0.897s	37	275 KB	

Obr. 27: Jednotlivé časy načítání webu www.mesto-most.cz (Vlastní zpracování)

Podle webpagetest.org je na stránce využitý komprimovaný přenos textových dat pouze na 46 %. Lépe jsou na tom obrázky, jejichž kvalitu služba hodnotí známkou C. Jako problém test vyhodnotil opět špatně nastavené cachování do mezipaměti.



Obr. 28: Podíl obsahu na www.mesto-most.cz pro desktop (Vlastní zpracování)

Při testování mobilního webu se stránka o velikosti pouhých 243 KB načetla za 2,078 sekundy, na první bajt prohlížeč čekal půl sekundy a musel odeslat 15 requestů. 53,9 % z celkové velikosti patří tentokrát HTML souboru, na druhém místě

jsou pak obrázky s 35,4 %. Posledních 10,7 % připadá CSS souborům, na mobilní stránce se tak nenachází žádný JavaScript. Stránky města Most dosahují dobrých výsledků i ve službě PageSpeed Insights, pro mobilní verzi pouze upozorňuje na CSS zdroje, které blokují vykreslení a způsobují prodlevu ve vykreslení stránky.



Obr. 29: Podíl obsahu na www.mesto-most.cz pro handheld (Vlastní zpracování)

Separátní mobilní web je v tomto případě velmi odlišný od toho desktopového. Uživatel se zde nedočká prakticky žádných grafických prvků kromě barevných pruhů a pár malých obrázků u článků. JavaScript se na stránce také prakticky vůbec nenachází, maximálně v podobě krátkého skriptu uvnitř HTML dokumentu.



Obr. 30: Obě verze stránek www.mesto-most.cz (Vlastní zpracování)

Mobilní web v této podobě by se možná dal nazvat extrémním, došlo zde vážně k velkým úpravám a na první pohled jde poznat, že se skutečně jedná o webové stránky města Most jenom podle nadpisu a domény. Uživatele zvyklého na

desktopovou verzi tato podoba může zmást. Ve výsledku je však moudřejší řešení mobilní web takto zjednodušit, než se snažit za každou cenu udržet vzhled a funkčnost desktopové verze a docílit tak příliš pomalého načítání. Jediný problém tohoto konkrétního řešení je ten, že některé odkazy na články vedou na nemobilní verzi, která je samozřejmě statická a donutí uživatele bojovat s neoptimalizovaným obsahem.

3.2.6 Opava

Desktopový web města Opava se v testu plně načetl za 4,840 sekundy, bylo k tomu zapotřebí učinit 122 requestů a stáhnout 1 634 KB dat. První bajt poskytl server ke stažení za 1,636 sekund. 69,4 % requestů bylo potřeba pro stažení obrázků a 16,5 % pro JavaScriptové soubory. V grafu se objevila i nová položka patřící fontům. Stránka tedy využívá písmo mimo základní výbavu uživatelů a prohlížeč ho musí pro správně zobrazení stáhnout. Při opakovaném příchodu se stránka načetla za 2,5 sekundy a bylo potřeba 8 requestů na 93 KB dat.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 4)	4.057s	1.636s	0.000s	1346	4.057s	97	1,104 KB	4.840s	122	1,634 KB	\$\$\$\$-
Repeat View (Run 1)	2.414s	1.447s	0.000s	1346	2.414s	3	38 KB	2.842s	8	93 KB	

Obr. 31: Jednotlivé časy načítání webu www.opava-city.cz (Vlastní zpracování)

V testu přenosu komprimovaných textových dat stránka obstála na jedničku, horší je to s kvalitou obrázků a využitím mezipaměti, za něž v testu dostala známky D a F.

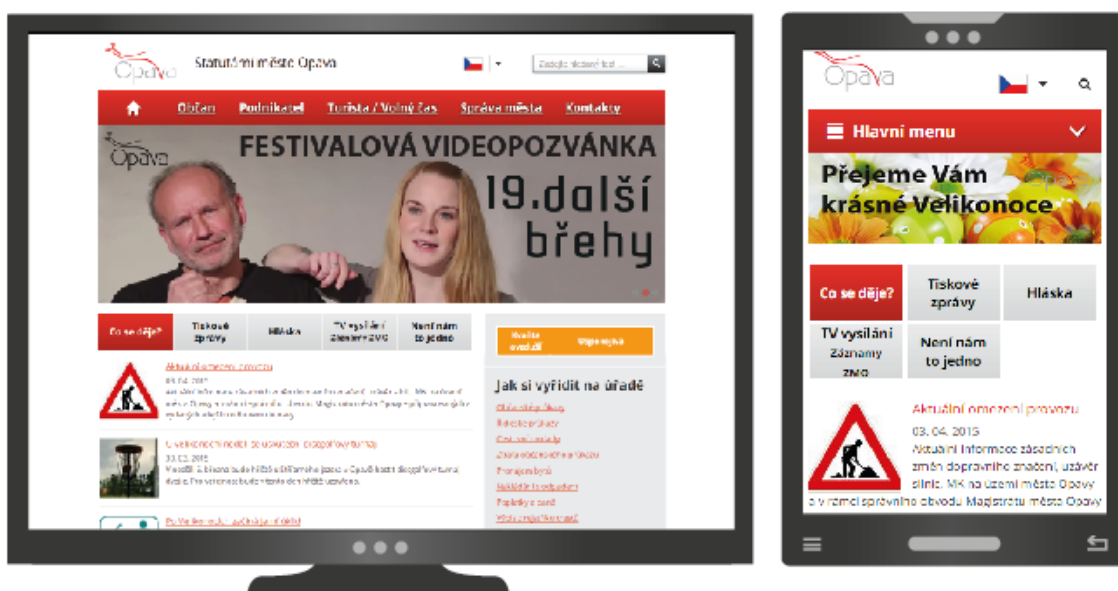


Obr. 32: Podíl obsahu na www.opava-city.cz pro desktop (Vlastní zpracování)

Na mobilním emulátoru se web načetl za 12,276 sekund, bylo potřeba 122 requestů na 1 773 KB dat a na první bajt prohlížeč čekal necelé dvě sekundy. Datově nejnáročnější jsou obrázky, které zabírají 61 % celkové velikosti, na druhém místě

jsou JavaScriptové soubory s 28,4 %. Podle služby PageSpeed Insights jsou nejzávažnější problémy nevyužívání mezipaměti, neoptimalizované obrázky a dlouhá doba odezvy serveru.

Stránky www.opava-city.cz také aplikují plnohodnotný responzivní design, všechny prvky reagují na změnu šířky okna a ani při rapidním zmenšení se uživatel nedočká horizontálního posuvníku. Při prohlížení na malém rozlišení se objekty pěkně přeskládají do jednoho sloupce, tato funkce je zde aplikována o něco lépe než v případě Plzně. Po nahlédnutí do zdrojového kódu je opět vidět velké množství nalinkovaných JavaScriptových souborů. Za zmínku stojí skript `css3-mediaqueries.js`, který funguje jako polyfill pro media queries a obchází tak absenci jejich podpory ve starších verzích prohlížeče. Dále je ze souborů `bootstrap.js` a `bootstrap.css` vidět, že stránka z nějaké části využívá kód front-end frameworku Bootstrap, kterému se věnuje i tato práce v kapitole 4. 1. Další CSS soubory jsou možná zbytečně roztroušené mezi několik souborů. Mediálních dotazů je vytvořených opravdu mnoho, emulátor mobilního rozlišení v prohlížeči Google Chrome jich napočítal 25. Maximální šířka stránky dosahuje zhruba jenom na 980 pixelů, na větších obrazovkách zanechává po stranách spoustu volného místa, určitě by tedy neuškodilo zaměřit se v mediálních dotazech i na větší rozlišení.



Obr. 33: Zobrazení stránek www.opava-city.cz na různých rozlišeních (Vlastní zpracování)

Největší slabina celé prezentace je v době načítání. Pro zobrazení webu je potřeba stahovat velké množství různých JavaScriptových nebo CSS dokumentů, výkonu stránky by pomohlo jejich protřídění a seskupení do menšího počtu souborů. Dalším náročným obsahem jsou obrázky, které nejsou nijak optimalizované a stahují se stejně na všech zařízeních.

3.2.7 Karlovy Vary

Při hromadném testu se stránky Karlových Varů sice začaly načítat mezi prvními, plně se ale načetly až v patnácté sekundě. Při individuálním testu se web načetl za 7,964 sekund, prohlížeč k tomu potřeboval vznést 93 requestů a stáhnout 4 519 KB dat. První bajt server předal prakticky okamžitě. Celých 81,2 % celkové velikosti zabírají obrázky, na druhém místě jsou JavaScripty se třinácti procenty. Opět se zde objevila i položka fontů. CSS soubory dosahují rekordních 0,7 %. Při druhé návštěvě je díky mezipaměti potřeba pouze 17 requestů na 78 KB dat a stránka se tak načte za 2,434 sekundy.

	Load Time	First Byte	Start Render	DOM Elements	Document Complete			Fully Loaded			
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 4)	7.870s	0.146s	0.000s	721	7.870s	92	1,724 KB	7.964s	93	4,519 KB	\$\$\$\$\$
Repeat View (Run 5)	1.862s	0.165s	0.000s	721	1.862s	9	32 KB	2.434s	17	78 KB	

Obr. 34: Jednotlivé časy načítání webu www.karlovyvary.cz (Vlastní zpracování)

Za přenos komprimovaných dat a obrázků dostala stránka od testovací služby známku F, zato využití paměti cache zvládá web bez mála na výbornou.



Obr. 35: Podíl obsahu na www.karlovyvary.cz pro desktop (Vlastní zpracování)

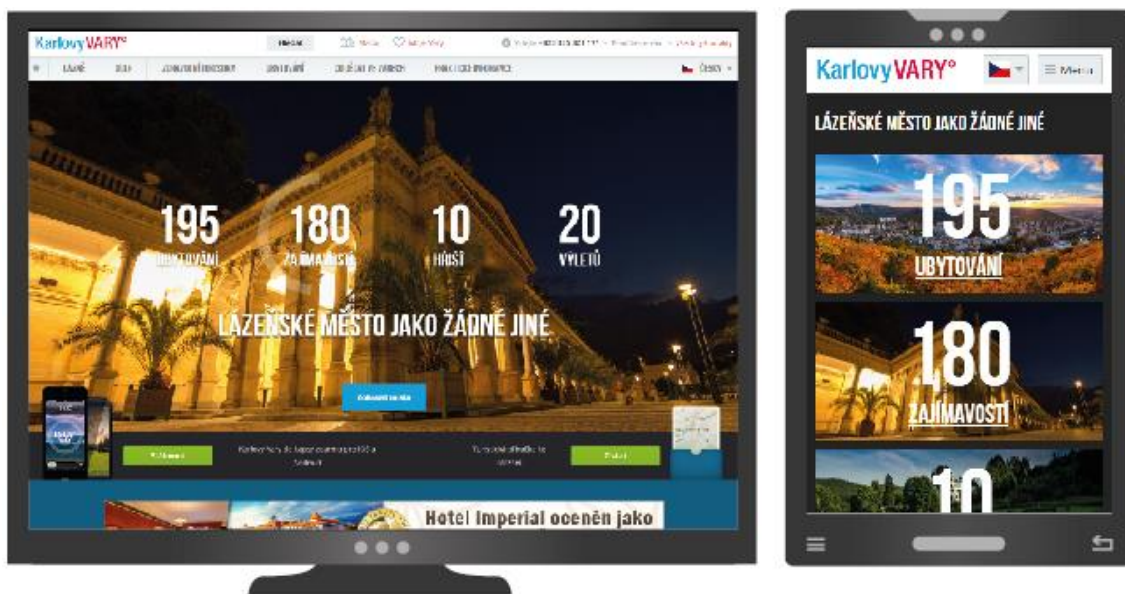
Při emulování mobilního prohlížeče a 3G připojení se stránka o velikosti 1 911 KB načítala 11,264 sekund a potřebovala 89 requestů, první bajt opět dorazil prakticky okamžitě. Podíl obsahu je vyrovnanější, 57,4 % celkové velikosti patří obrázkům

a 32,1 % JavaScriptu. Testovací služba od Google Developers pak doporučuje hlavně aktivovat kompresi a eliminovat JavaScriptový a CSS kód blokující vykreslení.



Obr. 36: Podíl obsahu na www.karlovyvary.cz pro handheld (Vlastní zpracování)

Stránky www.karlovyvary.cz jsou plnohodnotně responzivní opravdu na první pohled. Na rozdíl od Opavy nebo Plzně konečně využívají obrazovku na sto procent a i přechod na menší rozměry funguje výborně. Mediálních dotazů detekoval prohlížeč Google Chrome 19. Při zobrazení zdrojového kódu jsou vidět pouze tři nalinkované soubory s kaskádovými styly, ty jsou navíc ještě pro větší optimalizaci minifikované. JavaScriptový kód se na stránkách také párkrát ukáže, například knihovna Modernizr.js zajišťuje podporu SVG a detekuje, zda prohlížeč podporuje touch eventy, nebo knihovna Respond.js, která při detekci Internet Exploreru verze 9 nahrazuje chybějící podporu funkcí HTML5 a CSS3. Těmto knihovnám se věnuje kapitola 4.2.



Obr. 37: Zobrazení stránek www.karlovyvary.cz na různých rozlišeních (Vlastní zpracování)

I když se stránka při přístupu z pomalejší sítě načítá poměrně dlouho, v porovnání s ostatními testovanými stránkami využívá potenciál responzivního designu nejvíce. Ani zde ale nejsou obrázky optimalizované pro malá rozlišení.

3.3 Celkové vyhodnocení testu

Nejrychlejšími stránkami nakonec disponuje město Most, zde použitý velmi zjednodušený mobilní web dostává svému jménu a svou optimalizací poskytuje mobilním uživatelům nejrychlejší přístup. Weby využívající responzivní design se bohužel kvůli horší optimalizaci na mobilní přístroje načítají až přes 10 sekund, stále to jsou ale oprávněnější výsledky než v případě polovičatých řešení Prahy a Olomouce, nebo špatně optimalizovaného mobilního webu Hradce Králové.

Zmíněné polovičaté řešení pražských stránek si nakonec svým zpracováním vysloužilo poslední místo. Při prohlížení na mobilním zařízení se stránka stahuje jen asi o půl sekundy rychleji než nejrychlejší responzivní design, což je u mobilního webu velmi špatný výsledek.

Karlovarský web pak sice patří při prohlížení z mobilního zařízení mezi nejpomalejší, ale vzhledem k zpracovanosti vůči ostatním responzivním příkladům v testu na to má největší právo a vyplatí se na něj uživatelům počkat.

Hlavním kamenem úrazu jsou u všech příkladů neoptimalizované obrázky. Je to celkem pochopitelné; jednoduše použitelné řešení v podobě nových atributů elementu `img` je poměrně mladé a v době vytváření těchto stránek se tato funkce musela nahrazovat složitými JS kódy. To ale neomlouvá příklady využívající mobilní web.

Dalším faktorem je velké množství roztroušeného CSS nebo JS kódu do mnoha souborů, případně jeho nadměrné používání pro mobilní verze. Každý takový soubor znamená další požadavek a jejich seskupení nebo zredukování může ušetřit čas.

4 Praktické nástroje pro tvorbu responzivního designu

V průběhu vývoje responzivního designu vzniklo mnoho nástrojů umožňujících a usnadňujících jeho tvorbu, například v práci již několikrát zmíněné JS polyfily, jež zprostředkovávají moderní funkce starším prohlížečům, nebo různé testovací nástroje, které dokáží emulovat mobilní prostředí v pohodlí pracovního počítače. Speciálními nástroji jsou pak front-end frameworky. Ty dokážou poskytnout již hotovou responzivní kostru webu a uživateli stačí vybrat si z velkého množství šablon. Dále jsou to vizuální editory umožňující v prostředí prohlížeče naskládat jednotlivé elementy na stránku a vygenerovat jejich kód.

4.1 Front-end frameworky

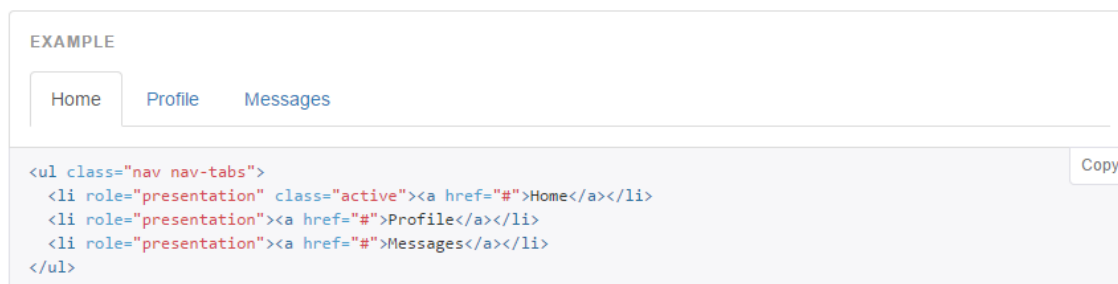
Front-end framework je sada nástrojů usnadňující tvorbu a manipulaci s typografií, elementy uživatelského rozhraní nebo rovnou s celým responzivním layoutem. Jejich základem jsou dynamická mřížka, přednastavené styly pro typografii a formulářové elementy a nejrůznější komponenty uživatelského rozhraní od malého tlačítka s ikonkou přes komplexní kód tvořící rotující karusel až po různá modální okna nebo navigace (Michálek, 2013b).

Nejnámějším a nejpoužívanějším frameworkem je Bootstrap dostupný zdarma z adresy www.getbootstrap.com. Nově přichodzí uživatel si na stránkách může vybrat z několika způsobů instalace. Tím nejjednodušším je stáhnout zabalený archiv s používanými fonty a minifikovanými CSS a JS soubory, všechny optimalizované pro responzivní design. Uživatel pak tyto soubory pouze umístí do svého projektu a připojí je k HTML dokumentu. Na stránce je pak k dispozici obrovské množství zmíněných komponent, jejichž deklarace využívá právě styly definované v předem stažených kompletních dokumentech, takže pro jejich instalaci stačí pouze zkopírovat vždy přiložený HTML kód. Na základě těchto komponent lze tedy buď vylepšit existující design, nebo sáhnout po kompletních layoutech a jednoduše sestavit úplně nový web. Za Bootstrapem stojí široká komunita příznivců, a tak je na internetu k dispozici ke stažení mnoho placených i neplacených šablon.

Pro pokročilejší uživatele nabízí Bootstrap propojení s preprocesory, konkrétně hlavně s preprocesorem Less. Pro tyto možnosti nabízí upravené zdrojové kódy, které se přes něj zkompilují a nainstalují.

Tabs

Note the `.nav-tabs` class requires the `.nav` base class.



The screenshot shows a Bootstrap navigation tab component with three tabs: 'Home', 'Profile', and 'Messages'. The 'Home' tab is active. Below the component is the HTML code used to create it, with a 'Copy' button on the right.

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Obr. 38: Příklad stažitelného komponentu a jeho kód (Bootstrap, 2015)

I když frameworky slibují kompletní responzivní řešení, nejsou dokonalé. Styly jsou sice definované pomocí mediálních dotazů, relativních jednotek a dokonce i technikou Mobile First, stále jsou ale stavěny univerzálně, takže uživatel nemůže čekat breakpointy šité na míru nebo nějakou větší optimalizaci kódu z hlediska výkonu. Další alternativou je například framework Foundation, který úzce spolupracuje s preprocesorem Sass, jinak funguje na podobném principu.

4.2 JavaScriptové pluginy

V práci bylo několikrát zmíněno, že pro responzivní design je potřeba několika moderních funkcí současného webu, ne vždy plně podporovaných staršími prohlížeči. Aby tato skutečnost nebránila rychlému rozvoji oblíbené techniky, začalo vznikat mnoho volně dostupných JavaScriptových pluginů, které svým kódem chybějící podporu dokáží vyrovnat.

Takovým případem je například knihovna Respond.js, která se specializuje na zpřístupnění responzivního designu prohlížečům bez podpory mediálních dotazů, konkrétně pro Internet Explorer verze 8 a nižší, samotní autoři ale uvádí, že je skript psaný tak, aby v případě potřeby dokázal obsloužit i jiné prohlížeče. Skript je psaný velmi minimalisticky, zabírá tedy pouze 3 KB dat v minifikovaném stavu a 1 KB v případě odeslání v zazipované podobě. Pro správnou instalaci se doporučuje psát mediální dotazy korektně a za použití techniky Mobile First. V HTML dokumentu se

pak skript připojí v kódu hned za CSS soubory. Když pak responzivní stránku navštíví například Internet Explorer 8, je zvyklý mediální dotazy ignorovat a mazat. V tu chvíli zafunguje skript, který CSS soubor sám znovu načte a každý blok s mediálním dotazem ze souboru vyjme a připojí do elementu head. Tyto jednotlivé bloky se pak aktivují nebo deaktivují na základě poměření zadané šířky s aktuálním viewportem prohlížeče. Knihovnu Respond.js nativně používá i framework Bootstrap.

Další, o mnoho mohutnější knihovnou, je plugin Modernizr.js. Na stránce modernizr.com si uživatel může stáhnout buď vývojářskou verzi o velikosti 42 KB, nebo si přes speciální rozhraní přímo vybrat, které součásti knihovny využije a stránka pak na základě preferencí vygeneruje o dost kratší kód. V nabídce jsou k nalezení polyfilly pro nejrůznější CSS3 elementy, např. pro vržený stín boxu či textu, zakulacené rohy, průhlednost barev atd., HTML5 elementy, např. HTML5 audio a video, Canvas, Drag'n Drop, atd., nebo pro nezařazené a extra možnosti, jako např. podpora SVG formátu apod. Stránka dokáže vygenerovaný kód pro větší ušetření místa i rovnou minifikovat. Modernizr funguje tak, že se JS soubor připojí do elementu head, skript se tak spustí hned po načtení stránky, detekuje problematické funkce a nabídne prohlížeči vlastní kód, který dokáže zpracovat.

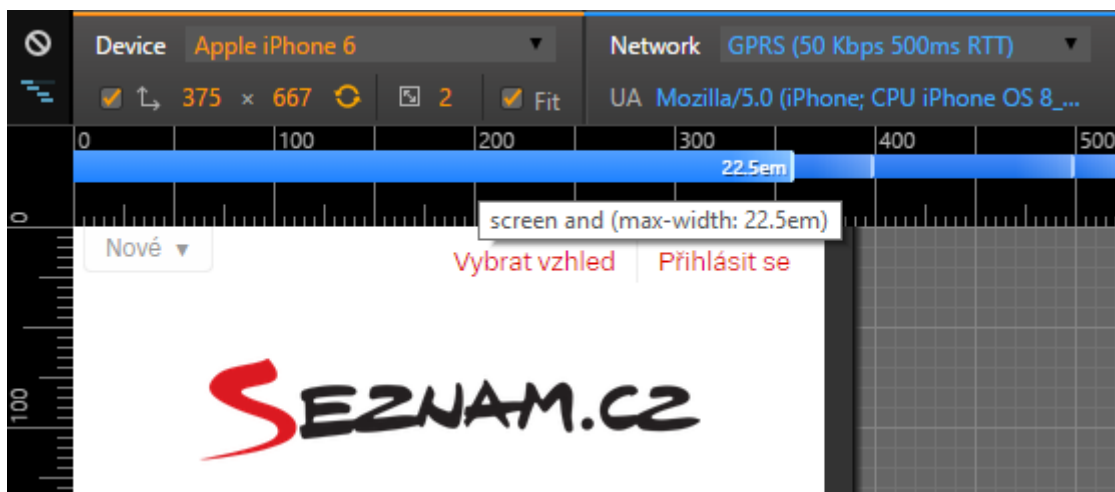
4.3 Nástroje pro testování

Chování responzivní stránky může být velmi nevyzpytatelné, testování je tedy jednou z nejdůležitějších součástí vývoje. Nic samozřejmě nenahradí konkrétní fyzické přístroje, pokud ale vývojář nedisponuje širokou škálou mobilních zařízení, z části ho mohou zachránit nástroje dostupné online.

Nejkratší cesta vede ke speciálním nástrojům novějších verzí prohlížečů Google Chrome nebo Mozilla Firefox. Oba dokáží, po speciální kombinaci kláves, emulovat mobilní obrazovku, u Mozilly Firefox je to kombinace Ctrl + Shift + M.

Konkurence od společnosti Google nabízí o něco sofistikovanější nástroj, po stisku Ctrl + Shift + J se uživateli naskytne pohled na prostředí ohraničené pravítky s ovládacím panelem, kde je možné vybrat ze široké škály přednastavených

přístrojů, nebo zadat vlastní rozměry. Při zadávání rozměrů nějakého přístroje je potřeba dávat pozor na již zmíněné Device pixel ratio, pokud například Apple iPhone 6 disponuje displejem s rozlišením 750×1334 a jeho Device pixel ratio je rovné dvěma, reálné CSS rozměry jsou poloviční, tedy 375×667. Kromě nastavování rozměrů dokáže prohlížeč emulovat i různé rychlosti mobilní sítě nebo graficky zobrazit na stránce definované mediální dotazy.



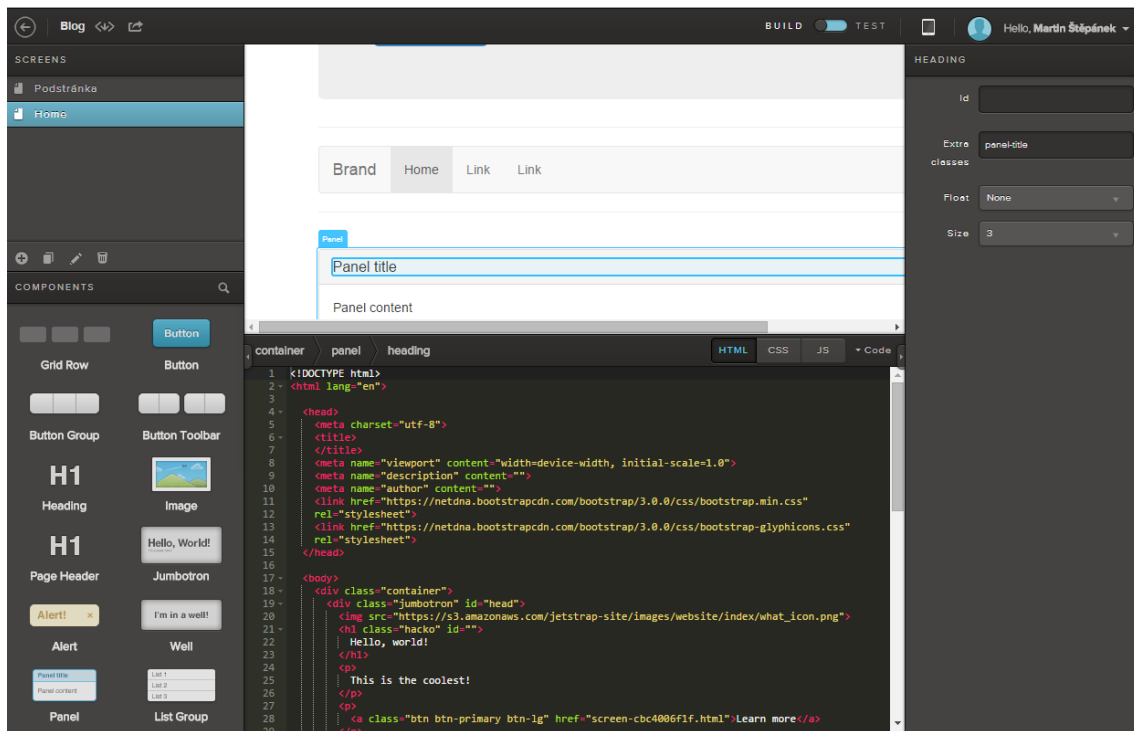
Obr. 39: Emulace mobilního zařízení v prohlížeči Google Chrome (Vlastní zpracování)

Podobné funkce nabízí například služba www.responsivepx.com, jejíž hlavní výhodou je možnost zvětšovat viewport po jednotlivých pixelech, nebo nástroj na stránce www.mattkersley.com/responsive, který zaujme možností zobrazit více náhledů různých rozlišení vedle sebe.

4.4 Vizualní editory responzivního designu

V souvislosti se zmíněným Bootstrapem je možné použít službu www.jetstrap.com. Ta za určitý měsíční poplatek nabídne v prohlížeči prostředí, ve kterém je možné skládat na sebe právě známé komponenty frameworku Bootstrap. V průběhu tvoření je možné nahlédnout do kódu, který se sám generuje podle naskládáních elementů a vloženého textu, a upravit ho. K dispozici jsou úpravy na úrovni HTML, CSS a JavaScriptu. Úpravy se okamžitě projeví i na již naskládáních prvcích, kterým lze pro tyto příležitosti přidat speciální identifikátory, služba jinak používá pouze předdefinované styly ze zdrojových kódů Bootstrapu. Po dokončení je možné celý projekt uložit do počítače, v základu se stáhne archiv s HTML, v případě zásahu do

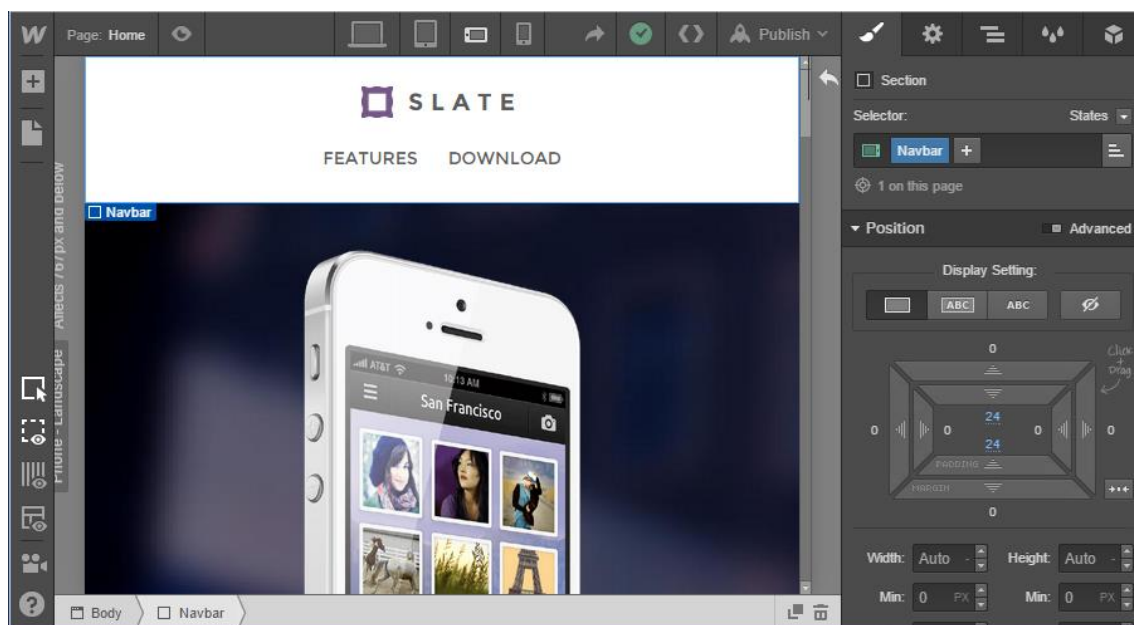
kódu i CSS či JS, soubory, ve kterých jsou předem nalinkované CSS a JS dokumenty jako odkazy na server Bootstrapu, nejsou tedy součástí archivu a fyzicky je zakladatel stránky nemá pod kontrolou.



Obr. 40: Prostředí nástroje Jetstrap (Vlastní zpracování)

Konkurencí je například služba www.webflow.com. Tu je v základu možné používat zdarma, ale takový uživatel může spravovat omezený počet stránek a všechny musí publikovat pod doménou webflow, až třetí úroveň platebního plánu umožňuje například export zdrojových kódů. Samotná služba pak není omezená žádným frameworkem, nabízí velké množství vlastních prvků, u kterých se však počítá, že si je uživatel sám nastyluje. Při označení elementu se tedy zobrazí panel s různými CSS vlastnostmi a intuitivně je umožňuje změnit. Webflow se chlubí, že jsou všechny výtvořené responzivní, ve skutečnosti to ale znamená, že nabízí čtyři pevné breakpointy bez možnosti jejich změny nebo přidání dalších. Jednodušším stránkám a uživatelům takové nastavení samozřejmě může stačit, ti náročnější si musí připlatit, vyexportovat kód a následně ho upravit ručně; služba totiž na rozdíl od Jetstrapu zásah přímo do kódu neumožňuje, maximálně v podobě vložení embed widgetu, což je prázdný element, který dokáže přijmout uživatelův HTML kód a přizpůsobit se mu.

Dalším konkurenčním editorem je například Adobe Edge Reflow jako součást balíčku Creative Cloud.



Obr. 41: Prostředí nástroje Webflow (Vlastní zpracování)

5 Shrnutí výsledků

Součástí práce byl průzkum dnes používaných mobilních zařízení a četnosti, s jakou jejich prostřednictvím uživatelé přistupují k internetu. Výsledky nebyly překvapivé, diverzita rozměrů mobilních přístrojů je obrovská a uživatelé s nimi prohlížejí internet již několik let častěji a častěji. Z toho vyplývá, že v dnešní době je optimalizace pro taková zařízení velmi důležitá a její absence může vážně uškodit návštěvnosti. Zároveň je z práce zřejmé, že v současnosti je už responzivní design, minimálně pro nové verze prohlížečů, bez větších problémů implementovatelný, jediným faktorem, který může brzdit jeho rozlet, jsou prohlížeče starší, u kterých je potřeba zavádět fallbacky a polyfills.

V průběhu následného testování stránek byl tedy kladen velký důraz na to, zda česká scéna vůbec stíhá reagovat na nejnovější trendy v oblasti tvorby webových stránek a zvládá svůj obsah optimalizovat pro mobilní přístroje. Pro testování byly zvoleny stránky několika největších měst České republiky, propagace měst na internetu je důležitá a města často pro jejich tvorbu vyhražují nemalý objem financí, je zde tedy prostor pro aplikování nejnovějších technologií. Tato oblast tak dokáže sloužit jako dostatečný vzorek reprezentující českou připravenost na prohlížení z mobilních zařízení. Hned první část testu však ukázala, že více než polovina vybraných stránek optimalizaci vůbec neřeší.

Zbylé stránky byly podrobeny detailnějšímu testu, ve kterém byly zkoumány zmíněné principy a náležitosti jednotlivých metod. Většina testovaných separovaných mobilních webů používala tuto techniku převážně nesprávně; výkonnostně v testu nedopadly o moc lépe než v základu pomalejší responzivní stránky a většinou obsluhují pouze malé procento mobilních zařízení. Responzivní vzorky pak mají největší problém v rychlosti načítání, zároveň také často nevyužívají úplný potenciál responzivního designu a pouze se příliš soustředí na co nejjednodušší doručení obsahu malým displejům. Následující tabulka shrnuje všechny testované stránky, které jsou nějakým způsobem optimalizované, a zaznamenává rychlost načítání na mobilních zařízeních a množství dat, která musí uživatel pro připojení skrze takový přístroj stahovat.

Město	Typ optimalizace	Doba načítání/objem dat
Praha	Mobilní web pod stejnou doménou	10,659 sekund / 1 418 KB
Plzeň	Responzivní design	11 sekund/1 835 KB
Olomouc	Mobilní web pod stejnou doménou	6,980 sekund/645 KB
Hradec Králové	Mobilní web	9,504 sekund/1 433 KB
Most	Mobilní web	2,078 sekund/243 KB
Opava	Responzivní design	12,276 sekund/1 773 KB
Karlovy Vary	Responzivní design	11,264 sekund/1 911 KB

Tab. 3: Shrnutí testovaných stránek a jejich výkon při prohlížení z mobilního telefonu
(Vlastní zpracování)

6 Závěry a doporučení

Prvním cílem práce bylo objasnit, v čem responzivní design spočívá, jaké principy a praktické nástroje se pro jeho tvorbu využívají a zdůraznit odlišnosti od metody separovaného mobilního webu. S výsledky této teoretické části se pracovalo v praktických kapitolách.

Teoreticky bylo dále vymezeno, že množství přístupů z mobilních telefonů, tabletů nebo jiných alternativních prohlížečích zařízení je tak obrovské, že nemůže být ignorováno. Separované mobilní weby pak, se svým většinou úzkým zacílením, nedokáží všechna tato různá zařízení obsluhovat. Responzivní design je tak bezpochyby technikou budoucnosti.

Následně proběhl detailnější průzkum této techniky, byly vysvětleny a přiblíženy všechny její náležitosti a problémy, se kterými se vývojář responzivních stránek potýká, spolu s návrhy na jejich řešení. Získané znalosti byly použity při určování testovacích kritérií pro hodnocení vybraných stránek. Nakonec proběhlo představení praktických nástrojů, které tvorbu takového designu usnadňují.

Hlavním výsledkem práce je tedy ucelený přehled o technice responzivního designu, výsledek testu mobilních a responzivních stránek pak dokazuje, že velké množství webů není pro mobilní zařízení vůbec optimalizováno, a pokud ano, často ne příliš efektivně. Správná aplikace responzivního designu může být tedy v současnosti velkou konkurenční výhodou.

Největším úskalím je mnohokrát zmíněný výkon stránky, na jehož optimalizaci si vývojáři musí zvyknout, například responzivní obrázky, které byly výkonnostním problémem velmi dlouho, mají nyní spolehlivé a funkční řešení, které je však velmi mladé a mnoho webových stránek jej neaplikuje. Tato konkrétní situace podtrhuje dynamičnost problematiky responzivního designu. Ve jménu této techniky ožívají i starší praktiky, jakou je například SVG formát, který je ideálním nástrojem pro zobrazení ikon, log nebo tlačítek díky vlastnostem vektorové grafiky a datové nenáročnosti.

Dále je potřeba uvědomit si, že responzivní design nemá za úkol pouze doručit obsah menším rozlišením, ale celkově se oprostit od pevných rozměrů a využít celou šířku obrazovky, bez ohledu na typ a velikost zařízení, a to v co nejmenší datové velikosti.

Téma nabízí i další pokračování v navazující práci, kde by bylo možné podrobněji probrat technickou stránku věci a demonstrovat zmíněné principy a nástroje na komplexním praktickém příkladu s cílem vytvořit perfektně optimalizovanou a ukázkovou responzivní webovou prezentaci.

7 Seznam zdrojů

- APPLESEED**, Jamie. 8 Limitations When Designing For Mobile. In: *Baymard Institute* [online]. March 21, 2012 [cit. 2015-01-18]. Dostupné z: <http://baymard.com/blog/mobile-design-limitations>
- BOOTSTRAP**. *Components* [online]. 16. 3. 2015 [cit. 2015-04-27]. Dostupné z: <http://getbootstrap.com/components/>
- GEMIUS SA**. *PC vs. Non-PC generated traffic* [online]. 6. 4. 2015 - 12. 4. 2015 [cit. 2015-04-14]. Dostupné z: <http://rankings.cz/en/rankings/pc-vs-nonpc.html>
- GHAZARIAN**, Armen. The Pros and Cons of Responsive Web Design vs. Mobile Website vs. Native App. In: *Designmodo* [online]. February 10, 2014 [cit. 2015-01-20]. Dostupné z: <http://designmodo.com/responsive-design-vs-mobile-website-vs-app/>
- GOOGLE DEVELOPERS**. *PageSpeed Insights* [online]. 8. 4. 2015 [cit. 2015-04-27]. Dostupné z: <https://developers.google.com/speed/pagespeed/insights/>
- HARVARD BUSINESS REVIEW STAFF**. How People Really Use Mobile. In: *Harvard Business Review* [online]. January 2013 [cit. 2015-01-18]. Dostupné z: <https://hbr.org/2013/01/how-people-really-use-mobile>
- KADLEC**, Tim. *Responzivní design profesionálně*. Vyd. 1. Brno: Zoner Press, 2014, 246 s. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.
- MAŇÁK**, Michal. Jak můžeme zrychlit své webové aplikace?. In: *Blog Michala Maňáka* [online]. 14. 10. 2010 [cit. 2015-01-21]. Dostupné z: http://www.manakmichal.cz/blog/optimalizace/jak-muzeme-zrychlit-sve-webove-aplikacBP_e/
- MARCOTTE**, Ethan. Responsive Web Design. In: *A List Apart* [online]. May 25, 2010 [cit. 2014-08-22]. Dostupné z: <http://alistapart.com/article/responsive-web-design>

MICHÁLEK, Martin. Jaké breakpointy zvolit v responzivním webdesignu?. In: *Vzhůru dolů* [online]. 27. 3. 2013a [cit. 2014-08-22]. Dostupné z: <http://kratce.vzhurudolu.cz/post/46416507703/jake-breakpointy-zvolit-v-responzivnim-webdesignu>

MICHÁLEK, Martin. K čemu je dobrý Bootstrap a frontend frameworky?. In: *Zdroják* [online]. 6.12.2013b [cit. 2015-04-17]. Dostupné z: <http://www.zdrojak.cz/clanky/k-cemu-je-dobry-bootstrap-frontend-frameworky/>

MICHÁLEK, Martin. Proč adaptivní, ne responzivní. A proč to responzivní zůstane. In: *Vzhůru dolů: Webový front-end ze všech stran* [online]. 4. 8. 2014 [cit. 2014-08-22]. Dostupné z: <http://www.vzhurudolu.cz/blog/23-adaptivni-responzivni>

MICHÁLEK, Martin. (Snad už) Definitivní responzivní obrázky - srcset a sizes. In: *Zdroják.cz* [online]. 16. 1. 2015 [cit. 2015-01-21]. Dostupné z: <http://www.zdrojak.cz/clanky/snad-uz-definitivni-responzivni-obrazky-srcset-sizes/>

POKORNÁ, Miluše, Stephanie RIEGER, Luke WROBLEWSKI, Peter-Paul KOCH a Lyza DANGER GARDNER. Trápení webdesigneru s viewporty. In: *Interval.cz* [online]. 17. 09. 2013 [cit. 2015-01-19]. Dostupné z: <http://interval.cz/clanky/trapeni-webdesigneru-s-viewporty/>

RICHTER, Felix. Mobile Phones Account for 17% of Global Web Usage. In: *Statista* [online]. August 20th, 2013 [cit. 2014-08-22]. Dostupné z: <http://www.statista.com/chart/1380/mobile-web-usage/>

SHAH, Harsh. Tablets are changing the way consumers engage with content. In: *Http://googlemobileads.blogspot.cz/* [online]. April 7, 2011 [cit. 2015-01-18]. Dostupné z: <http://services.google.com/fh/files/blogs/AdMob%20-%20Tablet%20Survey.pdf>

STATCOUNTER. *StatCounter Global Stats* [online]. 2015 [cit. 2015-04-17]. Dostupné z: <http://gs.statcounter.com/>

STANESCU, Bogdan a John SARAPATA. Google now indexes SVG. In: *Google Webmaster Central Blog* [online]. August 31, 2010 [cit. 2015-04-17]. Dostupné z: <http://googlewebmastercentral.blogspot.cz/2010/08/google-now-indexes-svg.html>

STERLING, Greg. Forecast: More US Mobile Web Users Than PC By 2015. In: *Search Engine Land* [online]. September 12, 2011 [cit. 2014-08-22]. Dostupné z: <http://searchengineland.com/forecast-more-us-mobile-web-users-than-pc-by-2015-92516>

SVOBODA, Martin. Použití responzivního designu při návrhu webových rozhraní. Brno, 2014. Dostupné z: http://is.muni.cz/th/395653/fi_m/thesis.pdf. Diplomová práce. Masarykova Univerzita. Vedoucí práce doc. Ing. Jiří Sochor, CSc.

WEBPAGETEST DOCUMENTATION. *Quick Start Guide* [online] [cit. 2015-04-17]. Dostupné z: <https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/quick-start-guide>

WORK, Sean. How Loading Time Affects Your Bottom Line. In: *KISSmetrics: A blog about analytics, marketing and testing* [online]. 2011 [cit. 2015-01-21]. Dostupné z: <https://blog.kissmetrics.com/loading-time/>

8 Seznam obrázků

Obr. 1: 17 % všech přístupů na internet bylo z mobilních telefonů (Richter, 2013).....	3
Obr. 2: Top 10 rozlišení osobních počítačů k lednu 2015 (Statcounter, 2015)	5
Obr. 3: Top 10 rozlišení tabletů k lednu 2015 (Statcounter, 2015)	6
Obr. 4: Top 14 rozlišení mobilních telefonů k lednu 2015 (Statcounter, 2015).....	7
Obr. 5: Dvě zařízení se stejně velkým displejem, přesto má displej vpravo čtyřnásobný počet pixelů (Pokorná, 2014)	9
Obr. 6: Plovoucí layout definovaný procenty (Knight, 2009)	11
Obr. 7: Graf zobrazující stoupající šanci na zavrnutí stránky z důvodů dlouhého načítání (Work, 2011).....	18
Obr. 8: Znamky a hodnocení udílené službou WebPageTest (WebPageTest Documentation)	21
Obr. 9: Problémy a jejich doporučená řešení nalezené službou PageSpeed Insights (Google Developers, 2015).....	22
Obr. 10: Přibližná rychlost načtení stránek jednotlivých měst (Vlastní zpracování).....	26
Obr. 11: Jednotlivé časy načítání webu www.praha.eu (Vlastní zpracování).....	26
Obr. 12: Podíl obsahu na www.praha.eu pro desktop (Vlastní zpracování)	27
Obr. 13: Podíl obsahu na www.praha.eu pro handheld (Vlastní zpracování)	27
Obr. 14: Obě verze stránek www.praha.eu (Vlastní zpracování).....	28
Obr. 15: Jednotlivé časy načítání webu www.plzen.eu (Vlastní zpracování).....	29
Obr. 16: Podíl obsahu na www.plzen.eu pro desktop (Vlastní zpracování)	30
Obr. 17: Podíl obsahu na www.plzen.eu pro handheld (Vlastní zpracování)	30
Obr. 18: Zobrazení stránek www.plzen.eu na různých rozlišeních (Vlastní zpracování)	31
Obr. 19: Jednotlivé časy načítání webu www.olomouc.eu (Vlastní zpracování)	32
Obr. 20: Podíl obsahu na www.olomouc.eu pro desktop (Vlastní zpracování)	32
Obr. 21: Podíl obsahu na www.olomouc.eu pro handheld (Vlastní zpracování)	33

Obr. 22: Obě verze stránek www.olomouc.eu (Vlastní zpracování)	33
Obr. 23: Jednotlivé časy načítání webu www.hradeckralove.org (Vlastní zpracování)	34
Obr. 24: Podíl obsahu na www.hradeckralove.org pro desktop (Vlastní zpracování)	34
Obr. 25: Podíl obsahu na www.hradeckralove.org pro handheld (Vlastní zpracování)	35
Obr. 26: Obě verze stránek www.hradeckralove.org (Vlastní zpracování)	35
Obr. 27: Jednotlivé časy načítání webu www.mesto-most.cz (Vlastní zpracování)	36
Obr. 28: Podíl obsahu na www.mesto-most.cz pro desktop (Vlastní zpracování)	36
Obr. 29: Podíl obsahu na www.mesto-most.cz pro handheld (Vlastní zpracování)	37
Obr. 30: Obě verze stránek www.mesto-most.cz (Vlastní zpracování)	37
Obr. 31: Jednotlivé časy načítání webu www.opava-city.cz (Vlastní zpracování)	38
Obr. 32: Podíl obsahu na www.opava-city.cz pro desktop (Vlastní zpracování)	38
Obr. 33: Zobrazení stránek www.opava-city.cz na různých rozlišeních (Vlastní zpracování)	39
Obr. 34: Jednotlivé časy načítání webu www.karlovyvary.cz (Vlastní zpracování)	40
Obr. 35: Podíl obsahu na www.karlovyvary.cz pro desktop (Vlastní zpracování)	40
Obr. 36: Podíl obsahu na www.karlovyvary.cz pro handheld (Vlastní zpracování)	41
Obr. 37: Zobrazení stránek www.karlovyvary.cz na různých rozlišeních (Vlastní zpracování)	41
Obr. 38: Příklad stažitelného komponentu a jeho kód (Bootstrap, 2015)	44
Obr. 39: Emulace mobilního zařízení v prohlížeči Google Chrome (Vlastní zpracování)	46
Obr. 40: Prostředí nástroje Jetstrap (Vlastní zpracování)	47
Obr. 41: Prostředí nástroje Webflow (Vlastní zpracování)	48

9 Seznam tabulek

Tab. 1: Přehled všech mediálních typů (Kadlec, 2014, s. 65)	12
Tab. 2: Seznam vybraných měst k testování (Vlastní zpracování)	25
Tab. 3: Shrnutí testovaných stránek a jejich výkon při prohlížení z mobilního telefonu (Vlastní zpracování)	50
Tab. 4: Seznam všech charakteristik, na které se může mediální dotaz zeptat (Kadlec, 2014, s. 66-67)	59

10 Přílohy

Charakteristika	Definice	Hodnota	Min/Max
width	Popisuje šířku zobrazovací oblasti zařízení	<lenght>	ano
height	Popisuje výšku zobrazovací oblasti zařízení	<lenght>	ano
device-width	Popisuje šířku renderovacího povrchu zařízení	<lenght>	ano
device-height	Výška renderovacího povrchu	<lenght>	ano
orientation	Indikuje, že orientace zařízení je na výšku nebo na šířku	portrait landscape	ne
aspect-ratio	poměr hodnot charakteristiky width/height	<ratio>	ano
device-aspect-ratio	Poměr hodnot charakteristik device-width/device-height	<ratio>	ano
color	Počet barevných bitů zařízení	<integer>	ano
color-index	Počet položek ve vyhledávací tabulce zařízení	<integer>	ano
monochrome	Počet bitů na pixel u monochromního zařízení	<integer>	ano
resolution	Rozlišení (hustota pixelů) zařízení.	<resolution>	ano
scan	Skenovací proces „televizních“ zařízení	progressive interlace	ne
grid	Vrací, zda je zařízení mřížkové (1) nebo bitmapové (0)	<integer>	ne

Tab. 4: Seznam všech charakteristik, na které se může mediální dotaz zeptat (Kadlec, 2014, s. 66-67)



UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta:

Martin Štěpánek

Obor studia:

Aplikovaná informatika

Jméno a příjmení vedoucího práce:

Aneta Bartůšková

Název práce:

Responzivní design a možnosti současného webu

Název práce v AJ:

Responsive Design and Possibilities of Contemporary Web

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Cílem práce je objasnit v čem responzivní design spočívá, jaké nástroje a principy se k jeho tvorbě používají, zdůraznit odlišnosti od mobilního webu a zhodnotit existující responzivní stránky.

Osnova práce:

1. Úvod
2. Responzivní design
3. Responzivní design vs. Mobilní web
4. Příklady implementace a jejich porovnání
5. Praktické nástroje pro tvorbu responzivního designu
6. Shrnutí výsledků
7. Závěry a doporučení

Projednáno dne: *8.10.14*

Podpis studenta

Podpis vedoucího práce