

Dekompozice webových stránek

Webové aplikace jsou chytré webové stránky (to jsou stránky, které se dělají na WBA). Tyto stránky jsou dynamické, nestatické a interaktivní. Tedy si můžeme představit e-shopsy, aukce, osobní stránky, portály.

Webové aplikace rozdělujeme velmi často na frontend a backend.

Frontend

Frontend je vše, co vidí uživatel. Je to důležitá část webu, je to jediné, s čím uživatel interaguje. Zaměřujeme se na design stránky (UI) a to, jak se uživateli stránky používají (UX). Používáme HTML, CSS a JS s frameworky. Framework je předpřipravená knihovna pro rychlejší používání webů. V tomto předmětu se zaměříme na Vue, ale existují i další, jako React, Angular či Svelte.

Backend

Backend je vše, co uživatel nevidí. Jedná se o databázi údajů, jejich komunikaci, e-maily, zpracování a třeba kalkulaci. Důležité pro backend je jeho rychlost, která ovlivňuje i rychlost frontendu. Používáme PHP, JS, Java ale i třeba C#. Jako frameworky používáme např. Express, Nette či Symfony.

Teorie uživatelského prostředí

Uživatelské prostředí je vlastně zobrazení, které se používá k ovládání určitého softwaru (případně webu, či cokoliv jiného). Zajímá nás jak se software chová vzhledem k uživateli a jak vypadá.

Uživatelské rozhraní (user interface, UI) je zevnějšek, zejména to, co uživatel vidí. Zajímá nás, jak vypadá toto rozhraní, jak je přehledné a další věci.

Uživatelský zážitek (user experience, UX) je to, jak se stránka chová vůči uživateli, jedná se např. o to, jak uživatel očekává, že se aplikace chová, a jak se opravdu chová. Například si můžeme představit vyskakovací okna, zda má stránka funkcionality a další.

V uživatelském prostředí nás zajímá jak UI, tak i UX.

Zásady

Zásady jsou požadavky (řekl bych checklist), toho, co by měla moderní aplikace splňovat. Důležité je, si uvědomit, že každá aplikace má jiné cíle a jiné publikum, a tedy je důležité se řídit instinktem a případně aplikaci zkoušet na uživateli. Zásad existuje více a dokonce si různé projekty své vlastní vymýšlí, aby zaručili kvalitu aplikace

Konzistence

Rozhraní by mělo být napříč aplikace totožné a nebo změněné velmi málo, mělo by platit, že se stejné tlačítka (např. s názvem či ikonou) chovají podobně. Když tuto zásadu porušíme, uživatel si nebude jistý co v aplikaci má očekávat a bude experimentovat s tím, co prvky dělají.

Přívětivost

Rozhraní by mělo uživateli vždy říct, co dělá špatně a nebo jeho chybu chytrě opravit. Můžeme si představit, že uživatel např. vyplňuje formulář a špatně zadal PSČ, které chceme ve formátu XXX XX a on jej zadal v XX XXX, tato chyba lze samozřejmě programově opravit. V jiných případech však můžeme uživatele o chybě informovat - např. v formuláři políčko označit červeně a přidat hlášku.

Stručnost

Rozhraní by mělo být na první pohled jednoduché a čisté. Uživatel by ihned neměl vědět všechny možnosti, co aplikace nabízí a tím uživatele přetížit uživatelský zážitek. Různé akce se mohou schovávat do menu, otevíracích tlačítek a dalších. V případě, že tyto informace nelze skrývat, je nutné je alespoň umístit stejné funkčnosti k jednomu místu.

Zpětná vazba

Rozhraní by mělo dávat najevo, co právě dělá a po všem, co uživatel způsobí, navrátit zpětnou vazbu (např. s chybou, ale i s úspěchem). Např. po odeslání formuláře obsah tlačítka nahradíme ikonou, tlačítko nepůjde zmáčknout a po zpracování se zobrazí hláška s chybou a nebo uživatel např. bude přesměrován.

Standard

Rozhraní, pokud to není nutné, by mělo vymýšlet nové prvky v rozhraní, které se dají nahradit jinými (jednoduššími) a známými prvky. Aplikace by se měla chovat obecnými a známými standardy, tedy např. při přihlášení se uživatel přesměruje a ne, že bude muset znovu kliknout na tlačítko po přihlášení.

Příjemnost

Rozhraní by mělo být natolik rozvinuté, že uživatel nebude muset dělat pro jednoduché akce desítky kroků. Nejčastěji používané stránky by měli mít různé vylepšení tak, aby čas na nich strávený byl co nejpříjemnější.

Efektivita

Rozhraní by mělo být efektivní, například tedy uživateli dělat hromadné akce či používat zkratky.

Očekávatelnost

Rozhraní by mělo být rozloženo tak, že je na první pohled jasné, jak je aplikace hierarchicky rozložena a jak vypadá tok stránky.

Princip minimálního překvapení

Princip minimálního překvapení nám říká ihned více věcí:

- uživatel je schopen zcela sledovat pouze jednu věc v jeden okamžik, dané věci říkáme focus
- při používání aplikace by mělo být jasné, co se (i bez uživateli akce!) stane

Princip minimálního překvapení zasahuje do různých zásad (a taky z nich vyplívá) a to zejména standardu a očekávatelnost. Před tvorbou aplikace by jsme si měli zjistit, kdo nejspíše aplikaci bude používat, jaké může mít zkušenosti, co očekává od aplikace a s jakým přístupem přistupuje k aplikaci.

Nejčastějším problémem webových stránek jsou různé popup okna, které uživatel nemá šanci očekávat a uživatele to demotivuje aplikaci dále používat. Další problémy jsou různé tlačítka s názvy, od kterých uživatel čeká kompletně něco jiného.

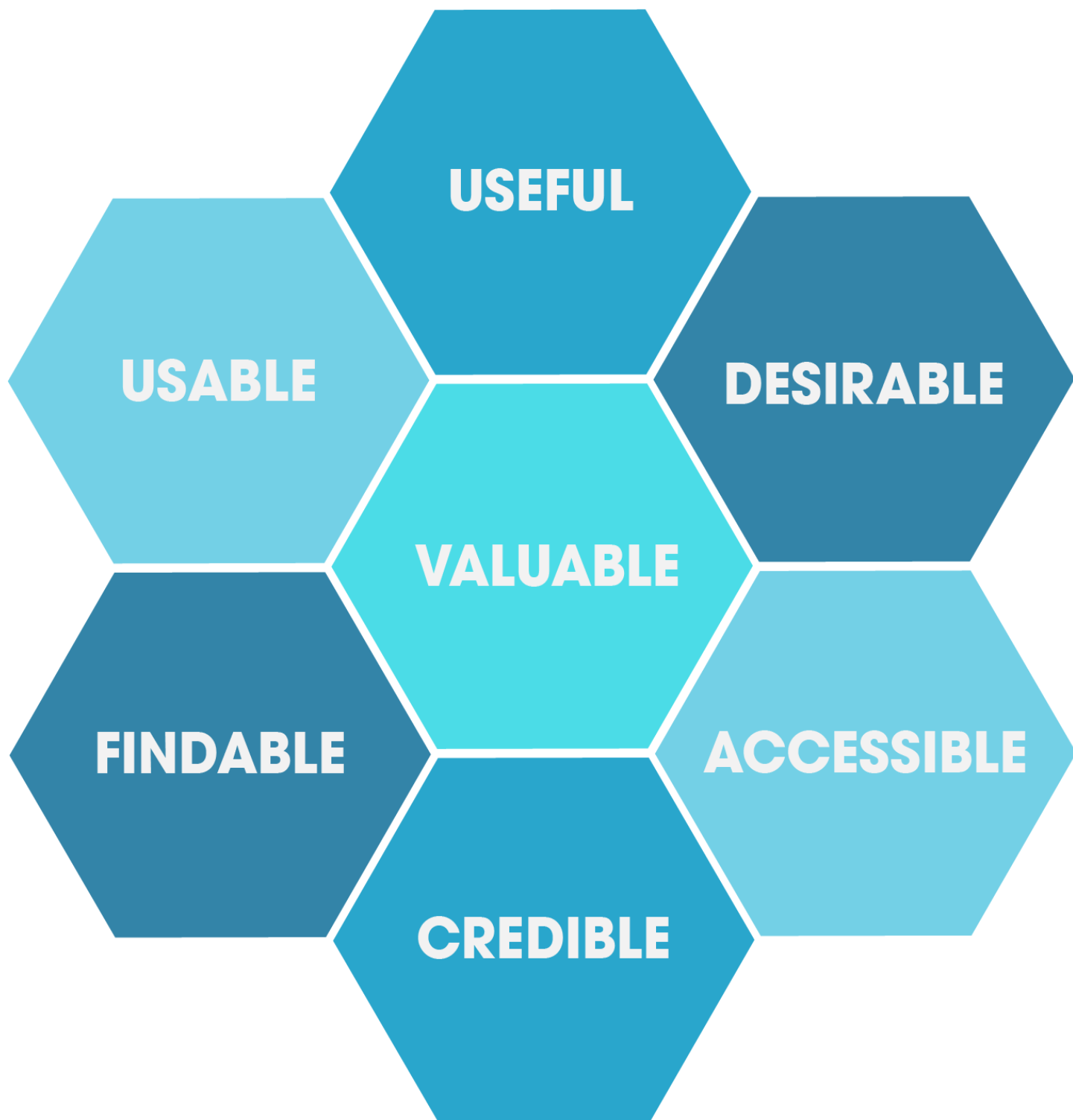
Problémům mezi aplikací a uživatelem **se nelze vyhnout**, ale je důležité je co nejvíce minimalizovat. Do aplikace může vkročit uživatel, se kterým se nepočítalo, nebo novým updatem se přidá něco, co tyto zásady poruší.

UX Honeycomb

UX honeycomb, česky UX pláštěv, je teorie, která nám říká, že všechny části chování aplikace se sebou souvisí a vždy vedou k hodnotné aplikaci.

Plástev nám říká, že aplikace by měla být:

- hodnotná
- užitečná
- žádoucí
- přístupná
- důvěryhodná
- dohledatelná
- použitelná



Problémy tvorby

Při tvorbě UX a UI je důležité si uvědomit, že uživatel určitě najde způsob jak používat aplikaci, o kterém jste při návrhu vůbec nepřemýšleli, a tím obejde aplikaci. Je důležité často, to jak uživatelé používají aplikaci, sledovat a tím vyhodnocovat, jestli je to chtěné chování. Obecně se ale tyto cesty uživatelům přenechávají.

UI a UX lze tvořit velmi špatně, proto doporučuji navštívit tyto stránky:

- <https://how-i-experience-web-today.com/>
- <https://userinyerface.com/>
- <https://twitter.com/ShitUserStory>
- <https://www.reddit.com/r/badUIbattles>

Sledování uživatelů

Při tvorbě webových aplikací často potřebujeme sledovat uživatele - ale ne v tom špatném slova smyslu. Jednoduše potřebujeme vědět, jak se uživatel na stránce chová a nebo na jaké stránky vůbec chodí. Nejčastější sledování je sledování požadavků, kde z různých trendů a grafů můžeme vidět, jak uživatele přicházejí a odcházejí. Pokročilejší sledování cílí na celé monitorování stránky při aktivitě uživatele.

Důležité je, si uvědomit, že sledování požadavků je zcela normální a dělá ho v podstatě každá stránka a to i bez Vašeho souhlasu. Sledování pokročilé, kde sledujeme např. i kam uživatel kliká, musí uživatel vědět a v moderní době je nutné mu povolit to vypnout.

Sledování může však přicházet i na jiné úrovni, a to, že zaplatíte lidem, aby Vaší stránku použili a buď je budete sledovat u Vás v kanceláři (nebo mimo ni) a nebo např. procházení nahraje. Jedná se o tzv. testery.

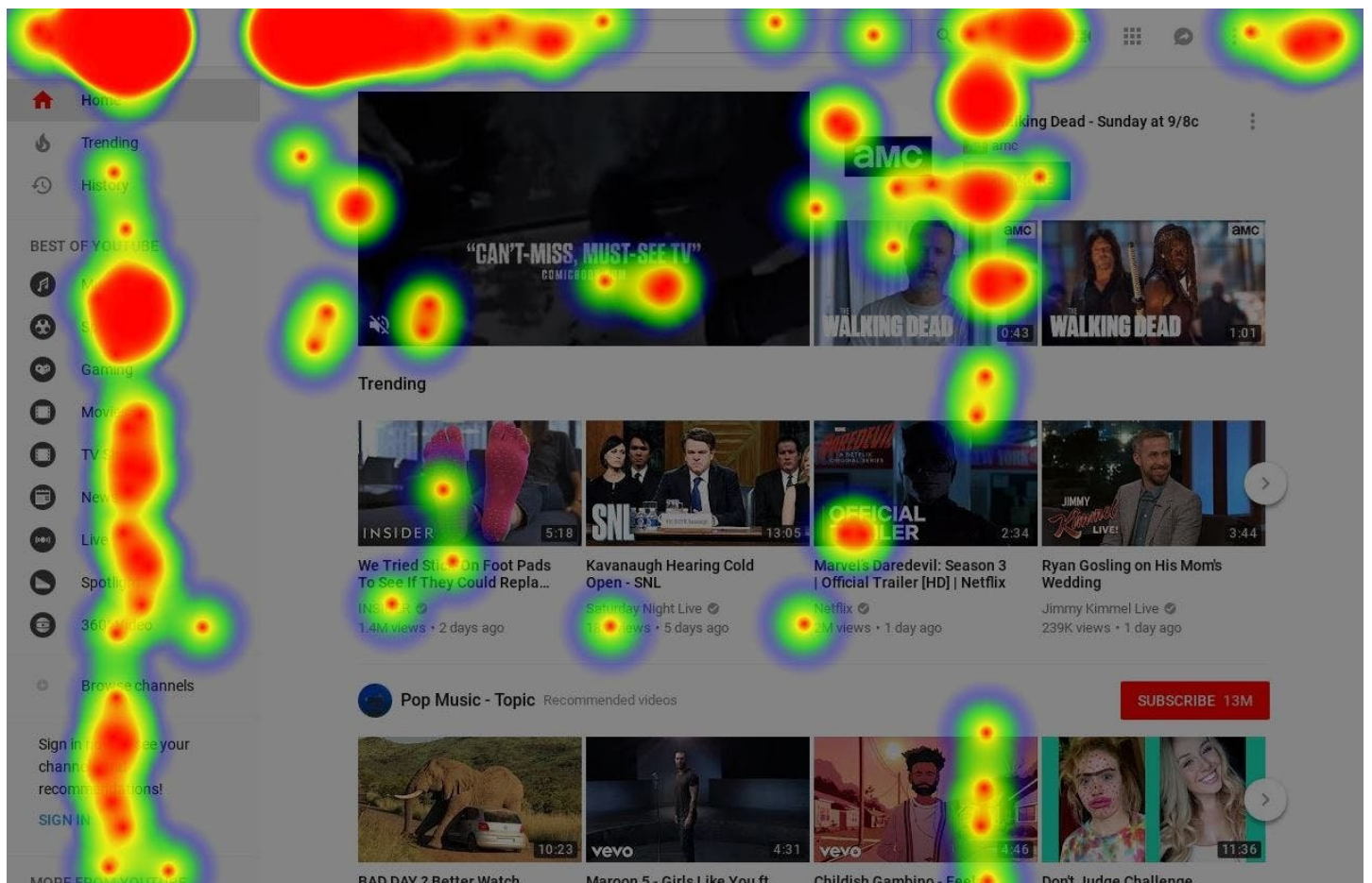
Sledování požadavků

Jedná se o nejjednodušší způsob monitoringu a je (jak jsem již zmínil) používán v podstatě všude. Nenarušujeme práva požadavků, protože nesledujeme to, co uživatel dělá, ale jaké požadavky na server (resp. na web) zasílá pomocí HTTP protokolu. Sledujeme tedy např. cestu, kterou navštívuje, jeho IP (můžeme vědět geolokaci, zemi, ...) a další. Z tohoto sledování můžeme zjistit nejnavštěvovanější stránku, země uživatelů, počet crawlerů a další. Implementace je pomocí jakéhokoliv web serveru či např. pomocí Cloudflare Analytics.

Sledování uživatelů

Sledování uživatelů je více agresivní sledování, které bere informace o přechodu stránek (jak dlouho byl na stránce, kdy odešel, ...), informace o uživateli, typ prohlížeče, zdroj přístupu, různé interakce a další. O tomto sledování je nutné informovat uživatele a dát mu možnost funkcionalitu vypnout. Nejčastěji se používá Google Analytics.

Do sledování uživatelů patří taktéž sledování kurzoru, které se v praxi dělá kvůli soukromí méně často, ale při velkém množství záznamů můžeme vytvořit tzv. heatmapu, které ukazuje např. nejčastější pozici myši, nejčastější místo kliku a další.



Psychologie uživatelského prostředí

Při tvorbě stránek je důležité si uvědomit, že design přímo i nepřímo ovlivňuje chování uživatele. Nepřímo znamená, že ho ovlivnit nechceme a přesto to design dělá, přímo právě tehdy, když ho chceme ovlivnit. Ovlivňovat můžeme špatným i dobrým způsobem a můžeme způsobovat dobrý i špatný dojem. Často dochází k nekalým praktikám, kdy uživatele ovlivňujeme příliš a až neeticky. Ovlivňujeme např. animacemi, barvami, fotkami, klikáním a dalšími věcmi.

Ovlivňujeme pomocí UI a i UX, nejedná se však jen o weby, ale i o aplikace, tisk a i další marketing. Důležité je taktéž si uvědomit, že vlastně není možné neovlivňovat, protože každý design je pro každého člověka jiný a jinak na něj působí.

Barvy na uživatele působí velmi mnoho, protože každá barva vzbuzuje různé emoce. Každý člověk je jiný a z jiné kultury. Každou použitou barvu je nutné rozmyslet. Teorie barev nám říká, jaké barvy by měli k sobě pasovat.

Obecně existuje několik typů písmen, které se hodí na různé věci a vyvolávají jiné myšlenky:

- Serif je typ písmen, které využívají patek (různé dekorace u konce čar znaků). Hojně se používá pro tisk, protože vyvolává důvěryhodnost, autoritu, formálnost a respekt.
- Sans-serif je typ písmen, které patky nevyžívají. Písma jsou moderní a používají se nejvíce na online služby.
- Cursive je typ písmen, které jsou bez-patek a vypadají jako kdyby je někdo napsal (jednotlivé písma se napojují). Vyvolávají pocity elegantnosti, kreativní mysli a tradičnost.
- Monospace je typ písmen, kde všechny znaky zabírají stejně místa. Používají se zejména pro kód (lepší viditelnost syntaxe). Vyvolávají pocity techničnosti, vzdělanosti a úhlednosti.

Obrázky jsou zejména pro webové stránky velmi důležité, protože platí, že obrázek obecně vydá několikrát více informací než text. Proto je vhodné, zejména dlouhé texty, vždy příhodnými obrázky (ukázkami z tématu, ilustracemi, ...) zlepšovat. Obrázky souvisí taktéž s tím, že máme velmi málo

času, abychom oslovili uživatele stránkou. Obrázky a přímým textem uživatele ovlivníme a ne každý bude chtít číst paragrafy informací. Pro čtečky (a vlastně pro všechny) bychom měli obrázkům dávat popisky (jak např. v atributu alt) tak i mimo něj. Obrázky by se měli doplňovat a jeden obrázek by měl popisovat jednu myšlenkou. Obrázky zdarma (např. jen s uvedením zdroje) i pro komerční použití nabízí např. [Unsplash](#), [Pixabay](#) a placené např. [Shutterstock](#).

Pro správnou tvorbu UI je taktéž důležité tzv. bílé místa, které se používají mezi jednotlivými texty a obecně prvky na stránce. Místo by mělo být na stránce dostatečně velké, aby se text a jednotlivé sekce lépe rozdělily. Když jsou ale místa příliš velká, mohli bychom uživatele zmást s chybějícím obsahem.

Typografie

Typografie, neboli typografická pravidla, je soubor různých pravidel pro psaní textu, zejména na internetu. Skoro každé formátování má své různé pravidla a veškeré najdete v internetové jazykové příručce pro jazyk český [ZDE](#). Jako další zdroje doporučuji:

- https://cs.wikipedia.org/wiki/Wikipedie:Typografick%C3%A9_rady#Pozn.C3.A1mka_k_norm.C3.A1m
- https://www.pslib.cz/milan.kerslager/Typografick%C3%A1_pravidla
- https://www.napocitaci.cz/33/typograficka-pravidla-uniqueidgOkE4NvrWuNY54vrLeM676nSxPa8k1rvWcEWLDBR_98/