



DESIGN & DEVELOPMENT WORKFLOW

andrea.crevola@3juice.com

<http://www.webdesign.3juice.com>



4 STRATEGIE PER LO SVILUPPO MOBILE

1. Delegare ai device
2. Versione mobile dedicata
3. App nativa
4. Responsive web design



DELEGARE AI DEVICE

- La prima strategia possibile è “non far nulla”
- I device in commercio possiedono browser che provano ad ottimizzare una pagina web in funzione di una fruizione da mobile
- “Zoom in / Zoom out” può dare problemi di visibilità / orientamento
- L’utente deve confrontarsi con informazioni e strumenti che potrebbero non essere necessari per il particolare contesto d’uso (soprattutto se non compiamo un’attenta analisi dell’utente e delle sue esigenze)
- Anche un “eccesso di codice”: problemi di performance, problemi di stabilità delle applicazioni
- Siti web pensati per altre modalità di input e di output



DELEGARE AL DEVICE: ALCUNI ESEMPI



Nokia Browser - Symbian



iPhone - Safari



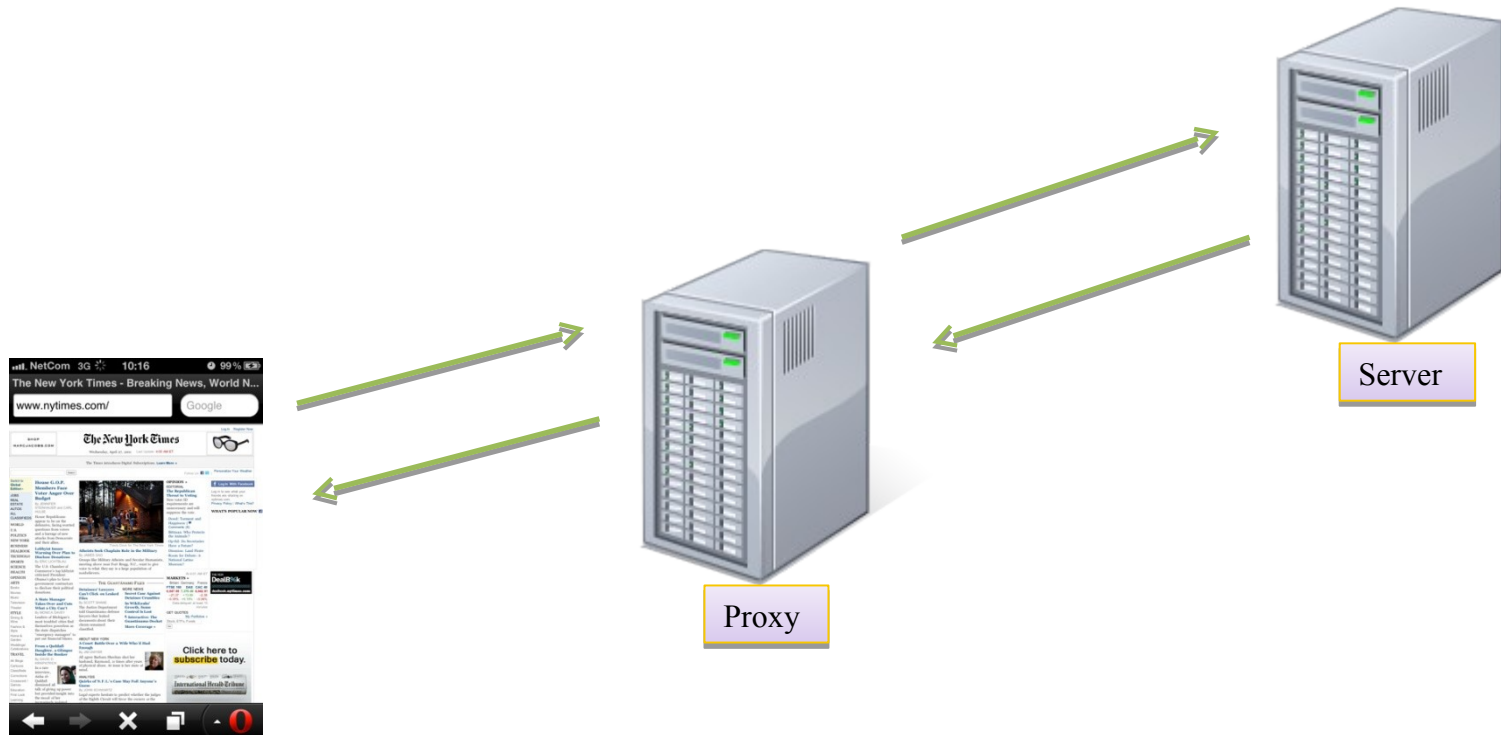
PROXY BROWSERS

- L'utente, quando richiede un URL, invia la richiesta ad un server terzo (proxy) che funge da intermediario
- Il server proxy richiede l'URL e tutte le risorse associate (HTML, CSS, immagini ecc.
- Sempre il server proxy si occupa del rendering complessivo e crea una “immagine” statica e compressa della pagina, dotata di un minimo di interattività (link)
- Questa immagine viene spedita al client in un'unica soluzione: risparmio di banda e di tempo di download (anche del 90%);
- L'utente può visualizzare la pagina ma deve rinunciare a molta interattività
- Es. Opera Mini, UC Browser (Cina), Nokia OVI Browser



OPERA MINI

- Browser multiplatforma: anche per Android e Apple;
- Molto diffuso nei paesi in via di sviluppo, dove spesso è il browser mobile più diffuso in assoluto



VERSIONE MOBILE DEDICATA

- Era (è?) una strategia molto diffusa
 - Inserimento di funzionalità e contenuti pensati appositamente per il mobile
 - Eliminazione di funzionalità che non sono al centro dell'esperienza mobile;
 - Eliminazione di contenuti, brevità, pagine secondarie per approfondimenti
 - Ottimizzazione dell'interfaccia-utente per migliorare un'interazione con sistemi di input mobile (touch, tastiere ridotte ecc.)
- Costruzione di un sito per “desktop”, di un sito per “smartphone” (quale?), di un sito per “tablet” (quale?)...
-



VERSIONE MOBILE DEDICATA: VANTAGGI

- se si parte da un sito già esistente, la selezione dei contenuti più adatti alla fruizione mobile evita di sovraccaricare l'interfaccia con informazione non pertinente, sovrabbondante e – spesso – indesiderata;
- parimenti, è possibile accantonare le criticità presenti sul sito per PC ed evitare di riproporre anche sul canale mobile le medesime imperfezioni (per esempio un codice HTML non standard o l'uso di formati proprietari);
- si può proporre un'architettura dell'informazione ottimizzata e contenuti pensati specificatamente per una consultazione “nomade”, così come si può costruire il sito attorno ad un'interfaccia-utente che sia capace di semplificare l'interazione a chi dispone di sistemi di puntamento e di immissione del testo meno efficienti di



VERSIONE MOBILE DEDICATA: VANTAGGI

- si ha un maggior controllo sulla visualizzazione grafica che si otterrà sui devices finali, per esempio adottando strategie volte ad evitare fastidiosi effetti come lo scorrimento orizzontale;
- si può ottenere un sito più semplice e leggero, rapido da scaricare e da consultare;
- Costringe il progettista a **focalizzare su ciò che è veramente importante** per l'utente;
- Gli utenti affermano di apprezzare la presenza di una versione “dedicata” a loro (**è l'utente a essere mobile, non il sito!**)



VERSIONE MOBILE DEDICATA: SVANTAGGI

- trattandosi di un prodotto che possiede una sua autonomia, sono spesso necessari **costi di progettazione e di sviluppo extra**;
- se il sito viene pubblicato contemporaneamente su più canali (desktop e dispositivi mobili) si può incorrere difficoltà a **mantenere aggiornate e consistenti le varie versioni**: un problema significativo sia per i contenuti, sia – in particolar modo – per le funzionalità più interattive;
- per favorire la consistenza dell'informazione tra le versioni pubblicate sui vari canali potrebbe essere opportuno migrare il proprio **sito su una piattaforma di content management**;
- la grande **variabilità dei dispositivi finali** rende difficile



VERSIONE MOBILE DEDICATA: SVANTAGGI

- la grande **variabilità dei dispositivi finali** rende difficile la produzione di un interfaccia-utente davvero universale, ossia capace di accontentare i requisiti di tutti i browser e le capacità dei vari devices;
- se ci si orienta anche verso la redazione di contenuti pensati appositamente per l'utenza mobile, si deve essere pronti a vedere quasi **raddoppiare il lavoro (e i costi) relativi al content management**: dalla redazione dei testi al montaggio audio-video dei contenuti multimediali
- Sono necessarie tecniche di **user agent detection** (non sempre affidabili)
- Strategia da “**Walled garden**” (vs “**One Web**”:
<http://www.w3.org/TR/mobile-bp/>)



APP NATIVA

- Una possibilità è abbandonare le tecnologie web in favore di una soluzione più compatibile e robusta rispetto alle piattaforme di destinazione
- I principali sistemi operativi (iOS, Android, Windows Phone, Blackberry OS) possono consentire l'installazione di vere e proprie applicazioni dedicate a scopi più o meno specifici;



APP NATIVA: VANTAGGI

- Possibilità di offrire un'esperienza d'uso allineata con quella offerta da altre applicazioni native (soprattutto nel mondo Apple);
 - Un'interfaccia web lascia al progettista maggiori margini di libertà e creatività, non sempre però con risultati all'altezza in termini grafici o di usabilità;
- Possibilità di avere riferimenti più certi in termini di input e output (es. iPhone 3, 4, 4s: sono tutti device touch con 320x480 display);
 - E' esperienza di ogni web designer la difficoltà di controllare “al pixel” l'interfaccia utente (anche se... cfr. “Il Dao del Webdesign”)
- Possibilità di interagire con componenti hardware (fotocamera, sensori GPS, accelerometri, bussola ecc.) e



APP NATIVA: VANTAGGI

- Possibilità di interagire con componenti hardware (fotocamera, sensore GPS, oscilloscopio, bussola ecc.) e software (es. rubrica telefonica, SMS, archivio immagini ecc.), sfruttandone **opportunità** e/o conoscendone le **limitazioni**;
 - Una strategia web based, invece, tende a fornire un minimo comun denominatore che impoverisce l'esperienza d'uso sui device più performanti (cfr. progressive enhancement)
- Offrono un modello di business appetibile per i fornitori di contenuto;
 - Gli utenti sono raramente interessati a comprare i contenuti di un sito web, mentre sono disponibili ad acquistare app.



APP NATIVA: SVANTAGGI

- Richiedono all'utente di possedere un determinato device e di stipulare un contratto con un soggetto terzo (es. Apple);
- Per molte esigenze, gli utenti preferiscono attuare una strategia di ricerca basata su una ricerca web mirata piuttosto che dotarsi di un app;
- Scarsa integrazione con il web
- Occupazione permanente di spazio di memoria sul device dell'utente



APP NATIVA: SVANTAGGI

- Ogni sistema operativo richiede lo sviluppo di un applicazione ad hoc: può comportare tempi e costi di sviluppo significativi;
 - le tecnologie web sono per definizione compatibili e multi-piattaforma
- Problemi di compatibilità tra versioni diverse dello stesso sistema operativo (es. Android): è oneroso intercettare e risolvere tutte le problematiche di questo tipo;
 - Le tecnologie web hanno un livello di standardizzazione maggiore, soprattutto in ottica futura;
-



RESPONSIVE WEB DESIGN

- Tecnica messa a punto da Ethan Marcotte e fatta conoscere tramite un celebre articolo su A List Apart (qui la traduzione:
<http://www.italianalistapart.com/articoli/17-numero-7-8-giugno-2010/71-web-design-reattivo>)
- Media queries + Fluid grids + Flexible images
- È una tecnica molto popolare per i suoi molti pregi anche se ancora in maturazione per diversi aspetti
- Va intesa, più generalmente, come una tecnica di progressive enhancement



MEDIA QUERIES

- Specifica CSS per chiedere al browser di considerare alcune regole di stile in funzione di un test sulle caratteristiche del device / browser
- I punti di “scatto” sono chiamati “breakpoints”

```
/* tutti i device */  
h1 {  
    color: blue;  
}  
/* solo per device con schermo <= 480px */  
@media screen and (max-width: 480px) {  
    h1 {  
        color: red;  
    }  
}
```

Media Queries, W3C Recommendation 19 June 2012

<http://www.w3.org/TR/css3-mediaqueries/>

S. D'Amico, “CSS Media queries: le basi”, HTML.it

<http://www.html.it/pag/19466/css-media-queries-le-basi/>



FLUID GRIDS

- Tecnica CSS per ottenere un'impaginazione in grado di adattarsi al variare della risoluzione dello schermo

E. Marcotte, “Fluid Grids”, A List Apart

- <http://alistapart.com/article/fluidgrids>
- Ogni dimensione è calcolata con la formula:
 - $\text{Target} / \text{Context} = \text{Result}$ (o in percentuale)
- È sufficiente cambiare il valore del contesto (es. via media queries) per modificare l'intera impaginazione



FLEXIBLE IMAGES

- Tecnica CSS
- Le immagini (e i video) sono oggetti a larghezza fissa che mal si conciliano con un approccio “fluido”: tendono a mantenere la loro dimensione naturale senza “reagire” al loro contesto (es. larghezza della colonna in cui sono inserite)

E. Marcotte, “Fluid Images”, A List Apart

- <http://alistapart.com/article/fluid-images>
- **Nota bene:** l’immagine è sempre la stessa, viene solo “stirata” o “compressa” in funzione dell’ampiezza del contenitore: può dar luogo a problemi di definizione dell’immagine oppure di prestazioni nei confronti di un accesso mobile

```
/* Immagini flessibili */  
img {  
    max-width: 100%;  
}
```



RWD: ESEMPIO

```
/* Immagini flessibili */
img {
    max-width: 100%;
}

/* Fisso le dimensioni del context base
(mobile) */
#pagina {
    width: 320px;
}

/* solo per device con schermo >= 600px */
@media screen and (min-width: 600px) {
    #pagina {
        width: 760px;
    }
}
```

```
/* solo per device con schermo >= 900px */
@media screen and (min-width: 900px) {
    #pagina {
        width: 960px;
    }
}

#colonna-sinistra {
    width: 24.26160337552743%;
    float: left;
}

/*
#colonna sinistra sarà pari a:
- 76px su mobile (e si potrebbe...)
- 182px su tablet
- 230px su desktop
*/
```



RWD: ESEMPI REALI

Boston Globe

- <http://www.bostonglobe.com/>

Starbucks

- <http://www.starbucks.com/>

Barack Obama

- <http://www.barackobama.com/>

Smashing Magazine

- <http://www.smashingmagazine.com/>

- ...



RWD: SVANTAGGI

- La gestione dei media dimensione fissa (immagini, video) non è ancora ottimale: si rischia di inviare file o troppo grandi / pesanti o troppo piccoli / leggeri;
- Una stessa base di codice XHTML deve essere utilizzato da layout differenti e per creare esperienze d'uso comunque differenti tra desktop e mobile
- Da solo non sempre è sufficiente: possono occorrere tecniche lato server per offrire contenuti e strumenti di navigazione ottimizzati
- L. Wroblewski, “Which One: Responsive Design, Device Experiences, or RESS?”,
 - <http://www.lukew.com/ff/entry.asp?1509>



RWD: VANTAGGI

- Consente di affrontare un gran numero di varianti nelle dimensioni dello schermo
- Utilizza soluzioni tecniche standard e accettate dalla maggior parte dei browser (laddove non compatibili sono comunque presenti tecniche alternative / polyfills)
- È una tecnica “future-friendly”: è pronta a garantire compatibilità anche nei confronti di display di device ancora non in commercio;
- Pone il progettista nella prospettiva “mobile first” e della “feature detection”
- Design in the browser Vs design in Photoshop



MOBILE FIRST

- Il sito va progettato e organizzato a partire da come potrà e dovrà essere consultato da device mobili;
- Questa strategia consente al progettista / team di lavoro di:
 - **valutare** con attenzione quali informazioni meritino davvero di essere veicolate all'utente e quale debba essere la gerarchia delle informazioni nella pagina: il sito web guadagna in efficacia ed efficienza, ci si concentra su ciò che è realmente importante per gli utenti e si crea così una maggiore usabilità complessiva;
 - **dare ai contenuti la precedenza sulla navigazione:** l'architettura del sito viene mantenuta snella pensando alle difficoltà che gli utenti mobile possono riscontrare



FEATURE DETECTION

- Si contrappone all'idea di un “minimo comun denominatore” da offrire a tutti indiscriminatamente (“graceful degradation”) e rappresenta un'integrazione all'idea di “progressive enhancement”);
- Va predisposto un sito che “si accorge” delle singole funzionalità (feature detection) e potenzialità del dispositivo in uso da parte dell'utente e – quindi – attiva funzionalità più interattive o dinamiche.
- Gli sviluppatori non guardano più all'universo “mobile” come fatto da classi di dispositivo basati su concetti “commerciali” (es. il sistema operativo, il browser, la versione del browser ecc.), bensì come un sistema fluido dove i dispositivi supportano o meno determinate caratteristiche funzionali



FEATURE DETECTION: OPPORTUNITÀ

- **reattività** nei confronti delle evoluzioni tecnologica: per esempio, se una nuova versione di Safari per iOS aggiunge una funzionalità in precedenza assente... il sito sarebbe in grado di accorgersi di tale disponibilità e rendere le funzionalità correlate subito disponibili;
- **compatibilità**: le classi di dispositivo sono sostituite da qualcosa di più simile a una carta d'identità del dispositivo, grazie alla quale le funzionalità del sito sono attivate o meno senza doversi preoccupare del modello / versione installata dall'utente;
- **affidabilità**: altre tecniche di individuazione del device cercano di intercettare nome/versione del browser in uso (*browser sniffing*) e quindi desumere quali opportunità



DESIGN NEL BROWSER

- Fino ad oggi i designer hanno usato strumenti “statici” per rappresentare l’interfaccia grafica di un sito web (con programmi Photoshop, Fireworks);
- Questo approccio produce file di dimensione fissa, “ottimizzati” per una risoluzione base (es. 1024x768);
- Nel momento in cui la possibilità di stabilire un’unica risoluzione base decade, l’uso di questo genere di strumenti non è più adatto
- Non si può usare uno strumento statico per rappresentare un oggetto “fluidi” come un sito web

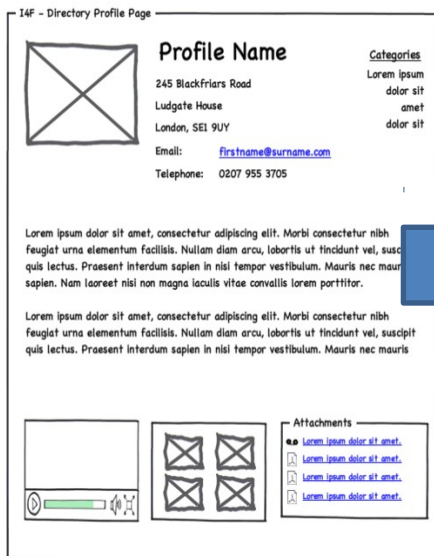


PROCESSO "CLASSICO"

Wireframe

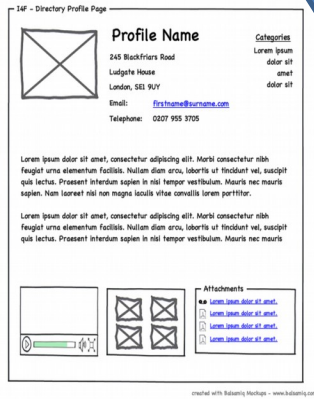
Graphic design

Implementazione



```
<div id="main">
<h1>pure CSS menus</h1>
<ul id="topmenu">
<li class="sub"><a href="http://
<ul>
<li><a href="http://
IE">Changer de navigateur</a
<li class="sub"><a href="http://
communauté Mozilla">Mozilla<
<ul>
<li><a href="http://
Mozilla seul avec des petits
<li><a href="http://
basé sur Mozilla">K-Meleon</
<li><a href="http://
fantôme pour surfer discrète
<li><a href="http://
la version finalisée de la f
</ul>
</li>
<li><a href="http://
l'innovation norvégienne">Op
<li class="sub"><a href="http://
```

PROCESSO “RESPONSIVE”



```
<div id="main">
<h1>pure CSS menus</h1>
<ul id="topmenu">
<li class="sub"><a href="htt
<ul>
<li><a href="http://
IE">Changer de navigateur</a
<li class="sub"><a href="
communauté Mozilla">Mozilla<
<ul>
<li><a href=
Mozilla seul avec des petits
<li><a href=
basé sur Mozilla">K-Meleon</
<li><a href=
fantôme pour surfer discrète
<li><a href=
la version finalisée de la f
</ul>
</li>
<li><a href="http://
l'innovation norvégienne">O
<li class="sub"><a href="
```

