

# Designing Usable Pocket Rockets: A Three Step Guide to Usability on the Mobile Web

---

# Designing Usable Pocket Rockets

---

Designing mobile sites is a different kind of web design. Much like your first experience of designing for the desktop web, it can be both exhilarating and daunting in equal measures. So many possibilities, yet so many usability restrictions.

Don't panic, we've been there too. As the guys responsible for maintaining the .mobi top level domain, we work with experts and beginners on a daily basis to help them get the best out of their mobile web strategies. Over the past few years we've come to appreciate just how different a beast the mobile web is to the desktop web, and what it means to deliver a truly usable and functional mobile site.

This paper is our '101' guide to getting your design and usability principles right. Our aim is not to help you to cut corners. Like any good design, mobile design is a craft and we encourage you to experiment and make mistakes. Instead, we're providing a set of best practice conventions that will help to get your mobile sites working and off the ground in quick time.

We'll start by setting some mobile web design rules to live and die by...

## Five Rules for Designing Usable Mobile Web Sites

- 1 **The mobile web is mobile.**
- 2 **Context is king.**
- 3 **The devices are (very) different.**
- 4 **Forget your dotcom thinking. You need unique content and design.**
- 5 **Never forget rules one to four.**

---

# The Rules

---

## **Rule 1 The Mobile Web is Mobile.**

Never overlook the obvious. Forgetting that the mobile web is mobile is the single most common factor in delivering a poor mobile user experience. The mobile web is used by people on the move. For this reason, you need to create user experiences that fit with this dynamic. Think trains, planes and automobiles, and the type of content and services that people will need in these environments in relation to your content and services. As such, the mobile web is a great place to help people find restaurants and grab sports scores and news headlines; but it's the wrong place to encourage people to sit down and browse a lengthy back catalogue of research articles. This basic point leads us to the fact that

## **Rule 2 Context is King.**

When it comes to usability, context is everything. Your users may be enjoying a packed commute, a scramble to a flight terminal or a cab ride home. Their time and attention is likely to be diminished, so you need to give them an experience that's more transactional than browsing-based.

This point is critical to the success of your mobile site: you can no longer account for where your users are accessing your site from. Think about this for a moment. The desktop web has definite contextual and physical limits. You need a desk (or table) or a lap and two hands free to use it, and this fact tends to dictate how and why it gets used. As such, the desktop web is great for research, casual browsing, and writing important emails and documents. The mobile web, on the other hand, has none of these contextual / spatial limits. You could be traveling, snowboarding, or loafing on the couch - all of which assume a different user mentality and suggest different user requirements. In short, the mobile web needs to be a lot simpler than its desktop counterpart and a lot more goal or task-based to get the job done.

## **Rule 3 The devices are (very) different.**

Mobile devices are, of course, very different to desktop and laptop computers. They have small screens and small (most often non-QWERTY) keyboards. Further, as a category, they also provide a wealth of form factor permutations: clamshell phones, candy bars, smart phones, PDAs, etc. These all present different technical requirements, browser formats and functional possibilities and restrictions.

Our accompanying paper, the 'Mobile Web Developer's Guide', provides you with a wealth of technical implementation best practices to help you reach out to the broadest possible spectrum of devices. You can find it at <http://dev.mobi/content/dotmobi-mobile-web-developers-guide>.

---

# The rules

---

## **Rule 4 Forget your dotcom thinking. You need unique content and design.**

Faced with the above rules there's really only one thing to do: forget the bulk of your dotcom thinking and approach the mobile web as the unique medium that it is.

If you can make this leap, then we guarantee you'll create a better mobile user experience. But be warned: it's not easy. You need to cast away much of what you hold sacred today. For example...

- Forget about three-column designs
- Forget about persistent primary and secondary navigation schemes
- Forget about any reliance you may have on Flash
- Rethink how your content is written and rendered
- Rethink how and where you use images and video
- Rethink how you render navigational aids like links and buttons

But most importantly, forget about the mobile web as a 'desktop lite' experience. If your objective is just to repurpose your desktop content for the mobile web, then you'll most likely fail your users: their location, their contextual requirements and their devices demand that they be treated in a different fashion.

## **Rule 5 Never forget rules one to four**

In short, as a mobile web designer, your first job is to think of your work as unique to the mobile medium. Your second job is to think through the points outlined above and ask this question, "Why should it be mobile?"

Often the answer may be that it doesn't, in which case you may only need to offer a simplified version of your desktop site and encourage users to visit it while sitting down. But if you can find compelling angles to present your content and services for mobile users, then you're probably onto a winning site.

With this in mind, use the following Three Step Guide to making your new mobile web sites as usable as possible:

- Step One: Good Mobile Information Architecture
- Step Two: Solid Prototyping
- Step Three: How to Design for Mobile Constraints

Let's look at each one in detail...

# Usability step one: Information Architecture. Make it transactional. Keep it simple and clickable.

Context is king. Because your users are likely to be snowboarding or boarding a plane, there are two key latency issues to consider when designing your information architecture:

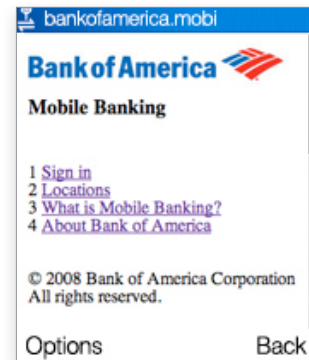
- i) bandwidth and speed
  - ii) 'click investment'
- ...and these two things are mutually exclusive.

Even with 3G transfer speeds, browsing the mobile web can still be painfully slow; you have to accept that users will be frustrated by poor navigation schemes or unnecessary click paths. That's why you need to be mindful of user goals from the outset and be sure to get them to what they need in a fast, logical fashion. In practice this means creating a rock-solid mobile information architecture (IA).

## Creating a Simple Mobile IA

The best advice for creating a strong mobile information architecture is to keep it as simple as possible. The following approaches work well when structuring your mobile IA:

- Limit choices. Take the content that's relevant to a mobile user and discard the rest. This results in a simple and focused IA that cuts down the risk of the user getting lost. This approach works well with small, tightly-focused sites.



BankofAmerica .com and .mobi sites show how limiting choices benefits the mobile web.

# Usability step one: Information Architecture. Make it transactional. Keep it simple and clickable.

- Create a simple site drill-down architecture that nests content into well-labeled categories. While this sounds straightforward, it's necessary to plan carefully before taking this approach. A typical Web site has sub-pages: users follow a link or 'drill down' to reach the next page.



FoxNews.mobi uses nested categories, that link to a specific article / page.

---

Usability step one: Information Architecture.

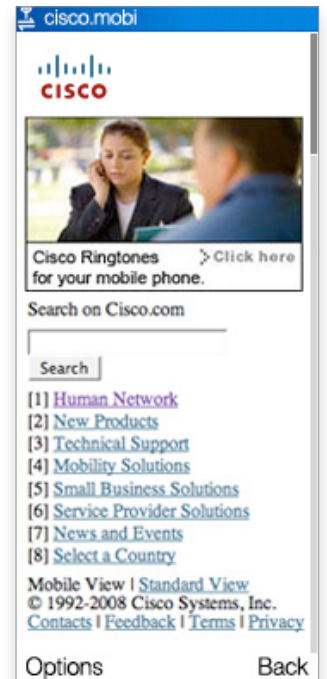
Make it transactional. Keep it simple and clickable.

---

### Drill-Down Recommendations

Here's some suggestions for creating an accessible drill-down architecture for your site:

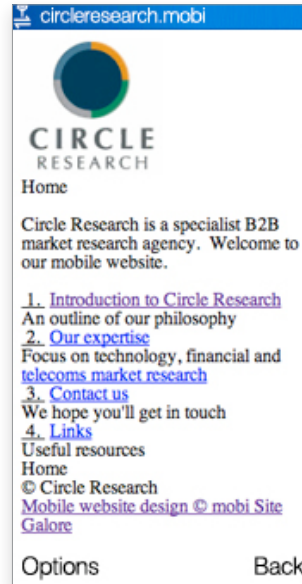
- Limit your content categories. Users become increasingly disoriented as they navigate deeper into a web site. With a Mobile Web site, their tolerance often stops at about five levels when using a low-end mobile browser. You should adjust this number upwards when targeting more capable devices that allow for more elaborate content presentation.
- Try to limit links to ten per page unless you know that the target device is able to present a greater variety of content. With this in mind, at a technical level, you should code your links with 'access keys', so that the user can use the phone's keypad to navigate your links. We recommend assigning up to ten access keys to any page (0-9) to ensure compatibility with older devices. You should adjust this number upwards when targeting more capable devices.



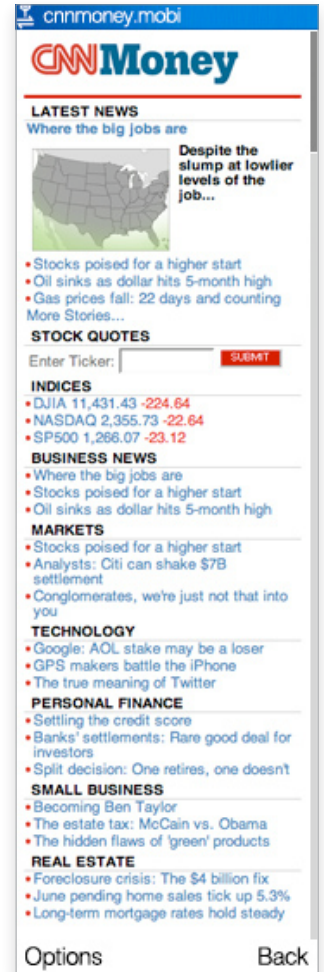
Cisco.mobi utilizes a zero to eight access key navigation.

## Usability step one: Information Architecture. Make it transactional. Keep it simple and clickable.

- Provide at least one content item with each category page (i.e., avoid aimless links). Giving users at least one sample of the content they're likely to find within a category is a good way to make sure that users find their way to the right place. Consider placing a link to the featured content with a one or two sentence overview (the shorter, the better).
- Prioritize links by activity, theme or popularity. This is often referred to as 'deck placement'. Sorting links in order of popularity ensures that the most sought-after content appears first on the list. And that, statistically, will improve a user's chances of getting to the right destination. The drawback, however, is that lower listed and newer items won't perform as well. You can work around this by maintaining some editorial control and featuring specific categories or content within the category.



CircleResearch.mobi showcasing a short explanation for each link.



CNNMoney.mobi uses themed categories and then nested links.

---

Usability step one: Information Architecture.

Make it transactional. Keep it simple and clickable.

---

### Thinking about Click Streams

An important difference between developing for the mobile web and developing for the desktop is creating the right flow of information for the user. Due to screen size limitations, the mobile designer needs to spread out information into multiple pages rather than present it on one page. Therefore, a well-designed click stream is an important concept in mobile design and content development.

In practice, this means - unlike a typical desktop web page - you need to paginate your content in the most effective way to propel the user through an article or a sequence of ideas. In turn, this means you need to bring out the salesperson in your copywriters or editors, because each break in pagination will be an opportunity for your users to tune out. As mentioned above, as with section or category-level navigation, it's critical that you load your content navigation points (or links) with 'pre-sales' pieces that hint of the 'payoff' to follow.



CNNMoney.mobi uses click streams to limit the length of articles.

---

## Usability step two: Prototyping.

# Test it. Unearth your usability issues before it's too late.

---

While thinking about click streams will give you a way to visualize how your mobile IA will work, it doesn't show what you could do better. If you plan to test the usability of your IA with potential users - as you always should - creating a prototype is a cheap and easy method of finding potentially expensive problems early on in the design and development process.

There are two main types of prototypes: paper prototypes and HTML prototypes. Here's how to do them effectively...

### **Paper Prototypes**

A paper prototype provides a simple way to gather feedback on how people will interact with your mobile site. You can create a paper prototype in a couple of hours and present it almost anywhere. The best way of doing this is to create a 'flip chart' of paper or cards that walks your testers through a series of interface wireframes. It can be tricky to find people to participate, but you can usually muster a few volunteers from your colleagues, friends and family. And remember, a testing sample of one person is better than zero!

Paper prototypes are valuable because they force you into a dialogue with your potential users. You get the opportunity to ask them how they use their mobile phone, plus where and how they use it. With this fundamental feedback in hand, you can get a high-level grasp of what types of content and services would work best for your new mobile service.

For the actual testing, you need to have a prepared script of tasks for your users to perform, and then let them loose on your paper prototype. Start the activity by explaining what goal you want them to reach. For example, 'Find today's movie times'. Use this time to interject conceptual asides like, 'Is this something you have done, or considered doing on a mobile phone?' Their experiences or perceptions about mobile technology will offer valuable insight as to how they will perform the task.

In general, since your purpose is to get high-level functional feedback, you should keep the tasks simple and avoid confusing your testers. Most importantly, leave room in your script to improvise. If they get stuck on a particular screen, draw a new script and / or screen on the spot and see if it works better. Don't rush from participant to participant, but perform each exercise on a one-to-one basis. Each study will deliver new insights and new answers to key problems. Keep modifying your script and screens for each participant until you feel you've enhanced your original prototype enough.

---

Usability step two: Prototyping.

Test it. Unearth your usability issues before it's too late.

---

### **HTML Prototypes**

The HTML prototype might take a little more time and effort to create than a paper prototype, but ultimately it will provide you with more flexibility and - if done well - a greater level of 'real world' interactivity and realism. To build this kind of prototype, you simply need to create each of your primary interface wireframes in HTML and give them some degree of interoperability. This lets you to link each of the screens together for realistic working example of your designs.

You can remotely test a simple HTML prototype through any desktop web browser or on a mobile device. While this testing method works in an observed environment like a paper prototype, it's harder to make changes on the spot. The real benefit comes from distribution and scale: you can get more testing and more feedback because you can widely distribute the prototype, and ask testers to perform one or two general tasks. You can also have them provide their feedback by email or an online survey.

We always recommend that you do HTML prototyping because it offers a handy way to do iterative development and avoiding coding and / or design mistakes before it's too late - and that is especially valuable in a medium that's difficult to test. Getting feedback from real people while developing your content and service ideas will steer your project towards a more valuable user experience.

---

## Usability step three: Designing for mobile constraints.

# Exploring the outer limits of browser and device capabilities

---

Regardless of how simple or how rich the desired outcome is, care needs to be taken to match your mobile design to the capabilities of your audience's devices: a site is only ever as good as the browser that displays it. Devices and browsers are already very good and are improving rapidly, but you can still expect to encounter many unique hurdles like slow load times, reduced device compatibility and inconsistent stylesheet support.

Here are some design conventions and restrictions to bear in mind...

### Designing for Different Screen Sizes

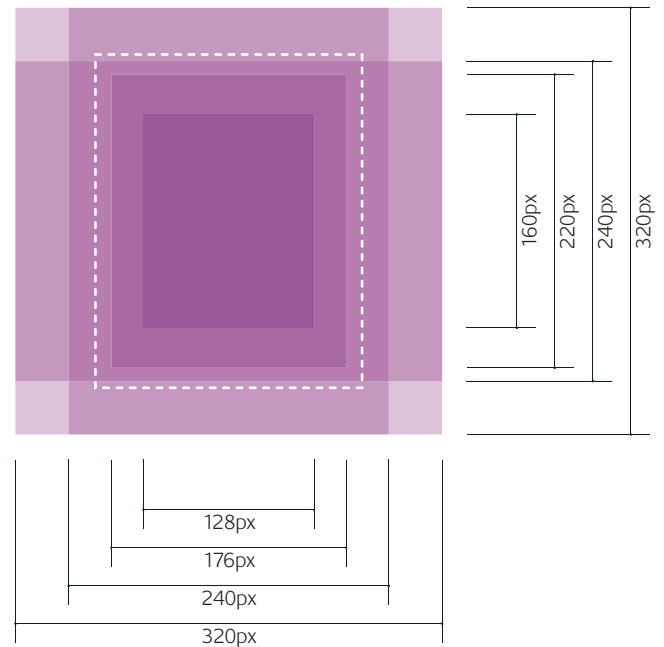
On the desktop web, we're lucky to have a manageable number of different screen resolutions – most PCs support resolutions of approximately 1024 x 768 pixels, and have a full keyboard and mouse. Conversely, with the mobile web, there's a lot more diversity in the physical attributes of the devices – screen sizes and keyboard layouts vary hugely across the range of devices currently in use.

Thankfully, it's not too hard to separate these screens into different classes of devices. This way, you can reduce the number of screen sizes you need to worry about and limit the complexity of your design process.

At the same time, you should also bear in mind that it's the screen width more than screen length which defines the usability and attractiveness of a mobile site. Images that look good in a low-end phone with a low resolution screen may only half or a third of that of a high-end 'smart phone' like the Apple iPhone or RIM Blackberry, and be completely unusable without requiring the user to zoom in to view the page contents.

This problem serves to underscore the importance of knowing your target audience and the devices they're likely to have. Once you know this, you're better equipped to decide what screen size(s) to target during the design stage of your project.

Side note: See section two of the Mobile Web Developer's Guide for more detail on the tools and methods for dynamically adapting content for the requesting device.



Recommended Max Size:  
200 x 250px

Mobile phone screen size variations and recommended max size.

---

## Usability step three: Designing for mobile constraints.

# Exploring the outer limits of browser and device capabilities

---

### Designing for the Right Device

When designing for the mobile web, you need to think about the different kinds of mobile devices that your customers will be using. Broadly speaking, they can be broken down into the following categories:

**Feature Phones.** These are the most common device type. Feature phones usually come in candy bar, clamshell or slider form. They have a 12-key layout and typically come with voice, messaging and data capabilities. Most feature phones sold in the past three years also come with built-in digital cameras and media players. They're typically used by and marketed to general consumers.

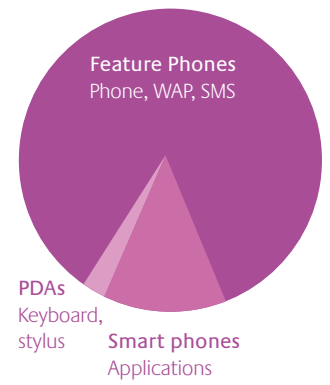
**Smart Phones.** Smart phones share many of the same properties as feature phones, but have two primary differences: they're able to run additional third-party applications and they have a slightly larger screen. Smart phones usually have a more fully featured operating system and they're typically marketed as advanced multimedia devices to consumers or as productivity devices to the business sector.

**PDA's.** These devices - having evolved from the personal address books and contact managers of the '90s - now often include voice, messaging, and data capabilities. PDA's have much in common with smart phones but differ in that much of their functionality is oriented towards organizational tasks rather than voice communications. Another difference is that PDA's often possess QWERTY keyboard layouts and a stylus in place of the 12-key layout found on normal phones. They also feature a larger screen that can often switch between portrait and landscape mode.

**Voice-Only Phones.** This class of devices represents the low-end, low-cost phones that are targeted at developing markets or low-usage consumers and, as of now, are not particularly relevant in the context of the mobile web.

In terms of market share, feature phones lead the way by a large, large margin, but bear in mind that the borderline between the feature phones and smart phones is constantly shifting towards the smart phone category. The latest feature phones often are equal in functionality to yesterday's smart phones.

These different features and screen sizes will ultimately effect how you design your mobile site. Is your service a pocket productivity application targeted at business users with PDA's or is it a gaming experience for the mass market? Either of these choices would bring unique interface design challenges and opportunities.



Share of different phone types

## Usability step three: Designing for mobile constraints.

# Exploring the outer limits of browser and device capabilities

### Mobile Web Navigation Paradigms

We're accustomed to a variety of standard navigation conventions on the desktop web such as tabbed primary navigation schemes and secondary navigational menus located on the sides of the main page content.

These schemes give us useful visual clues about where we're located within a desktop web site and will provide reference points on how to navigate within it. Obviously, this is more difficult to do in a mobile context because of the mobile devices' limited screen size and navigational input constraints. While it's possible to use desktop style navigation scheme like tabs on a mobile device, they generally don't work as well as they do on the desktop due to limited screen sizes, and pointing and clicking capabilities.

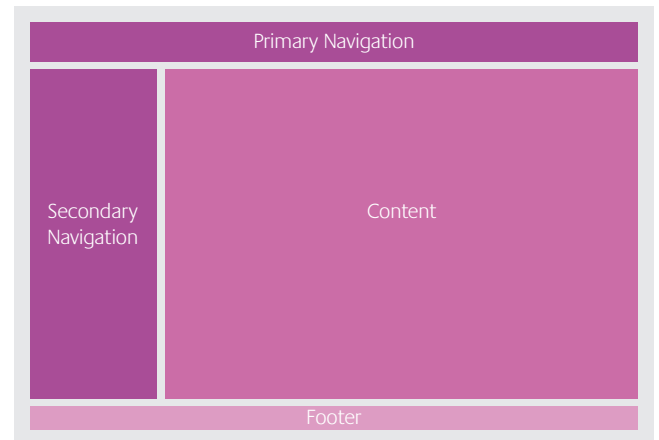
### Standard desktop website layout

The most common navigation schemes use a simple vertical list of options, often assigning and listing the corresponding numbers (0–9) to the access keys for keypad navigation. You can design this list in many ways using stylesheets or images. At the same time, it may be worth considering the support of more advanced navigation methods for higher-end phones to ensure a rich user experience on these devices.

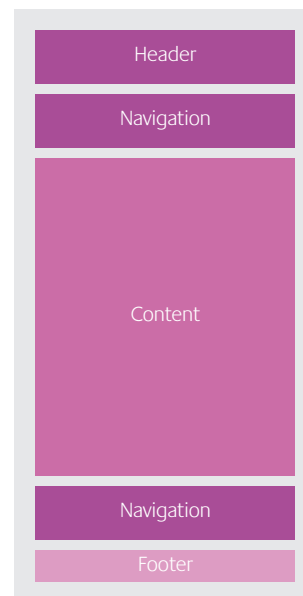
### Standard mobile website layout

Showing multiple levels within your navigational listings doesn't tend to work well because it gives your users too many options to consider and consumes valuable screen real estate. A better way to present navigation is to show only the options related to the page that your users are viewing.

Having said this, you should always also provide escape points, either as links to the next section, to the parent section, to the home page or all of the above. These links usually work best at the bottom of the page and allow the user to move on, up or out without scrolling back to the top of the screen.



Standard desktop website layout



Standard mobile website layout



- 1 Select previous link or scroll up
- 2 Select next link or scroll down
- 3 Back or page up
- 4 Forward or page down

Basic navigational set-up for mobile web usage

---

## Usability step three: Designing for mobile constraints.

# Exploring the outer limits of browser and device capabilities

---

### Layout and Information Design

The navigation schemes that we inherit from the desktop web may lead you down a blind alley in the mobile domain because they've been predominantly designed in a landscape format, where pages are wider than they are tall.

Designing for the mobile requires switching your thinking to portrait mode where the content is typically taller than it is wide. Most landscape layouts and navigation schemes - like horizontal tabs and columns of text - don't work well on the mobile. Instead, think of the mobile like a page in a book with a portrait orientation. So use a single column with text that's left justified.

At the same time you need to totally re-think your notions of 'above and below the fold' interfaces. As mentioned, the mobile web demands that you rip up the layout and presentation of your desktop content assets and start again with a mobile mindset. To this end, interfaces and content presentation need to be short, sharp, to the point and - as far as possible - within the screen.

### Technical Design Considerations

Alongside these 'look and feel' conventions sit a wealth of technical standards and constraints that need to be factored into your design process as you move from paper-based design to technical implementation of your mobile site.

While recognizing that the mobile device landscape is more technically varied than its desktop counterpart, you should note that much of this diversity has been addressed by device manufacturers and the mobile industry at large and there is a wealth of supporting tools, services and documentation to help the mobile designer and developer negotiate this fast-moving arena.

For further information about best practice technical implementation of mobile sites, we recommend you (and your development teams) review our Mobile Web Developer's Guide (see: <http://dev.mobi/content/dotmobi-mobile-web-developers-guide>) and also our free development resource, <http://dev.mobi>. At dev.mobi you'll find an array of tools such as our leading-edge device compatibility database, DeviceAtlas™, as well as ready.mobi, our site-compliance testing tool as well as documentation, blogs and discussion forums.

---

## Conclusion: Feel liberated

---

The upside of writing about mobile usability today is that the market is awash with new opportunities to deliver experiences to users in new and unexpected ways. Conversely, this also frames the downside: so much of what needs to happen to create definitive best practice guidelines is yet to be discovered. New applications and interfaces are springing up every month, many of which improve upon the recommendations of the recent past.

That said, the principles outlined in this document are tried and tested, and ensure that you avoid some common design and usability mistakes.

And what of the future? Our advice is to go back to our introductory rules and remember that the mobile web is fundamentally different. This ought to be liberating. Somewhere out there there's a totally niche, totally mobile-dependent application that's waiting to happen. It may never have been thought of before in the desktop domain, and it may not have an 'off the shelf' interface waiting for it... but planning it, testing it and designing it promises to be a rich and profitable exercise for you and your potential users.

So go for it!

We encourage you to share your experiences with us at [mobiThinking](#), so that the rest of the mobile web can design a better tomorrow. Check in to the site and leave us your commentary.