What is a good mobile application anyway?

Jaap van Ekris

There are many mobile applications out there, but some are better than others. Good mobile applications are designed for people who are on the move. But when is a mobile application a good application for daily use? Are there any common pitfalls that developers and users step into? If you have to buy a mobile application, what are the important features and criteria to look for in such an application? In this article we show some of the common misconceptions developers have, the problems they cause for end-users and what to look for if you are buying an application.

Not every mobile application is suitable for use in the real world

Users want to access their data wherever they are. Applications on mobile devices allow you to access your data whenever you want. With the introduction of .Net it has become a lot easier to develop mobile applications. But have they become better applications because of this? Porting an application from the desktop to a mobile device is becoming easier every day, but simply porting the application is not providing the best application to the user. It seems that besides the normal expectations of desktop applications (response speeds, user friendliness, etc.) there are new demands. In my personal opinion, good mobile applications are more than simply providing the same functionality and quality that the desktop provides: it is about balancing the device limitation, user environment and the needs of the user into a very delicate mix that will become a well designed application which performs well when you use it in the real world.

A good mobile application saves you time

First of all, mobile applications should be designed to make your life easier. The application should save you time, effort or stress. This means that there is few room for duplication of data or operations on the device. Especially when the Windows Mobile platform provides a lot of services for you that are standard for every device. When the application does not integrate completely with these services, you end up duplicating information by hand. Most notorious are the contacts: some mail applications and navigation applications, both thriving on addresses of people you already know lack the integration with your addressbook, forcing you to enter these items by hand. This results in a lot of wasted energy and time. Another example can be found in your calendar: some applications deal with a certain type of time management, but ignore the build-in

calendar. There are numerous applications that allow you to track time or usage of time (including travel planners, work-out management applications and Time registration applications that ignore to integrate into your calendar by reading or creating calendar appointments. In the end, this forces you to duplicate your efforts: you have to copy data from one application to the other on the same device. This results in time being wasted and mistakes being made while the user is really trying to do something else.

Limitations of mobile devices put limitations on mobile applications

A mobile device does have its limitations, which proves to have some implications for mobile applications. These limitations are very dominant in the design and use of these applications. The small form factor of mobile devices come with a price: the screen size is more limited, input devices are different, device memory is very limited, CPU-power is limited and batteries have a limited capacity.

Although many current devices have a more than 64Mb of program memory available, an application using more than 4 Mb of that memory still causes problems for the speed of the device. Applications with that kind of footprint in memory tend do load slow and because of their volume slow down the entire device. This is not the only side-effect. Applications with a bigger footprint or requiring extra memory tend to consume more energy. This is because the processor starts consuming more energy if it has more processing to do. So heavy applications are not only slowing down your device, they are limiting the practical use of your batteries as well. The battery-life is extremely important. It can make the difference between a device that is usable and a device that is not usable for half the day. So good applications should have a small memory footprint.

Good mobile applications promote the feeling of freedom

One of the most appealing aspects of mobile devices is the idea of freedom: do what you want to do, when you want to do it, whenever you want to do it. This is partially because of commercials showing people being happy with internet and e-mail in the most exotic places. In the end we expect to be independent of fixed infrastructure. Every extra dependency which is introduced by an application can hamper the feeling of freedom. If an application becomes dependent of an internet connection you simply get the feeling that you have traded your dependency on a desktop into the dependency of specific (wireless) infrastructure. Nothing can be more annoying to a user than being in need of certain information and having to conclude that the dependency on internet or desktop blocks obtaining that information.

Some ideas of mobile freedom, like constant access to mobile internet, do not mean that you have to be connected to the internet all the time. Being connected all the time costs energy and money. Besides that, access to mobile internet can not be maintained under all circumstances. The environment of mobile device can differ greatly, having a lot of impact on the type and quality of network connections. Heat reflecting glass can block GSM/GPRS/UMTS networks, making these networks unusable. GSM/UMTS and WiFi also never have 100% coverage within areas. This results in unstable connections at

best, a lot of annoyance for end-users and the risk or errors in the data. Therefore, unless a perfectly operating connection to the internet can be guaranteed, the dependency on the internet is not a good thing for an application.

For example, *a purely webbased logistical planning or registration system for mobile employees becomes completely useless when the internet connection fails. As we mentioned earlier, Wireless LAN and GSM coverage is not guaranteed, basically making the application useless because it could fail to deliver critical data at moments that people want to act upon them.*

Another option is to allow the application to connect to the internet, but also allow the application to work with locally stored data. When an application stores the most important data on the mobile device, you can continue to work even when your internet connection is unreliable or even non-existent. Most of the time only recent data is interesting anyway, so the data is pretty small most of the times. By synchronizing data, when it is possible, you can update the data when technology allows it and it is cost-effective to do so. Making appointments in busy parts of your schedule might trigger you to refresh your calendar before you make the final appointment. Having the choice between 100% accurate data and less cost (both in money, time and device-energy) is a very important one.

Mobile applications should fit the user more closely than a desktop application

One of the biggest challenges for mobile applications is coping with the context and the user. Shrinking the desktop application to a smaller screen is not what constitutes a good mobile device. In sharp contrast to what software developers seem to think, mobile devices are not only used in office environments. In office environments, the user has all the time and precision to deal with the quirks of an application. Research conducted in Sweden in 2004 by Siemens, indicates that mobile devices end up in much stranger situations like being used outside in the rain or in a gym. So, as a mobile user you generally do not have time, concentration or precision to operate an application. Typically you are doing something else and your PocketPC is supposed to support you: you are running to catch a train, you are in a meeting trying to focus on what other people say, you might be standing outside in the cold trying to find some information, you might be driving a car or might even be standing inside a dark bar with a lot of shady characters around you. This has some implications for the mobile application: it should anticipate on a user that is not patiently operating the application.

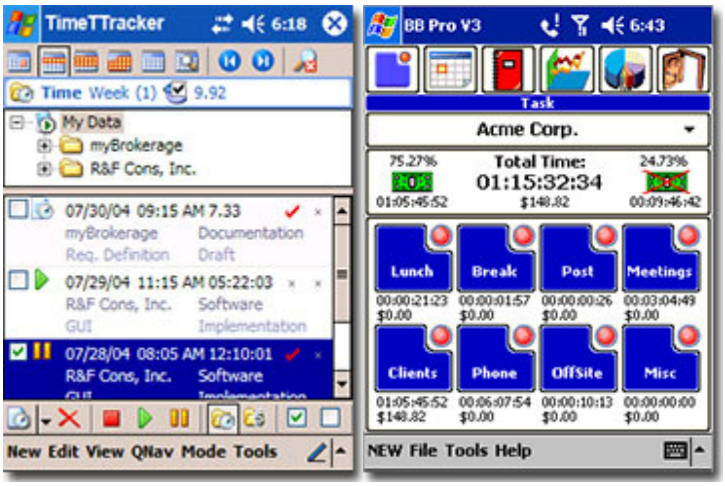
Some applications proof to be much better at coping with your environment than others. As I see it, there are many factors that are important for use of applications in the real world:

- Are you using the application in a comfortable office environment or are you using the application outside, in the rain and cold?

- Are you able to focus on using the application, or do you have to use the application while focusing on more important things, like the road while you are driving, other people during meetings, your environment while walking?
- Are you able to operate the device with a very precise stylus or keyboard, or will you be using fingers or handwriting recognition with a bad handwriting (because of the cold or you using gloves) even on a very small screen?
- Do you need the application in a dangerous situation right on the spot, can lives be in danger when you don't have the information at hand? (e.g. medical information)
- Do you have enough time to study every detail the application presents (for example small roads on a map) or do you need clear, simple and unambiguous directions (like arrows telling you where to go)?
- What level of detail do you need at a given moment anyway?
- Would you detect audio and video signals like alarms at a given moment, or is your device placed in an invisible location?

This implies that when you choose an application, you might want to consider investigating the application with these situations in mind. It could be that you do not have such a stressful life, but it could also mean that you, your device and thus the application could face a lot of challenges posed by your environment.

A great example of specialization can be found in the Time registration applications, where a lot of applications have been developed, some focussing on specific environments:



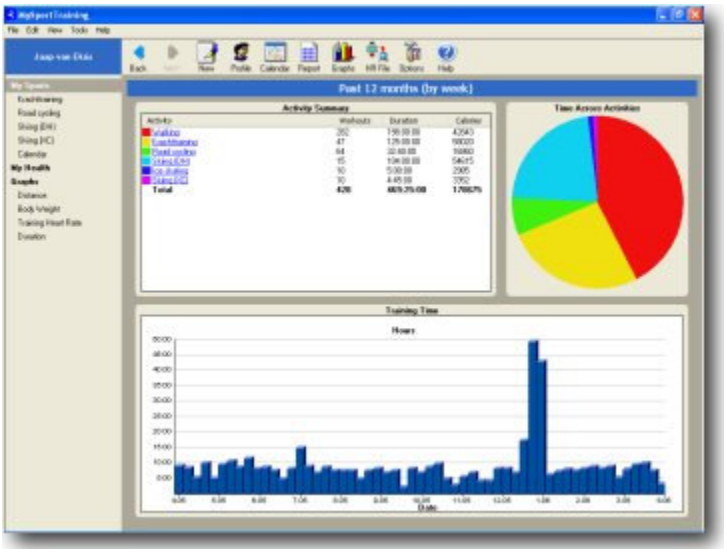
Both the user interfaces of TimeTTracker (left) and Billable Buddy (right) have the possibility to enter records for billing. TimeTTracker's records contain much more information, but starting and stopping the registration for a specific project is more complex for TimeTTracker. Billable Buddy allows you to start a task with the push of a thumb on a 2.8' screen of the smaller Windows Mobile Phone Edition screens, allowing you to start the timer when you are "on the run", but providing less detailed information.

Some applications are smarter: they try to match badly entered input on the expected input. The best example is *OV Reisplanner*, an application that has very few entry fields, and that matches all its input on the expected input, correcting it for errors. Even when you have heavily corrupted input (say 60% typos) the application still manages to find the right railway station for you.

Good mobile applications seemlessly integrate with the desktop

People do not only use Windows Mobile devices. Mobile devices typically are part of a larger chain of systems. The last thing you want as a user is that your data is stuck on the mobile device, increasing the possibility of data loss. The mobile device complements the more fixed systems of laptops, desktops and servers. In this chain, the mobile device is a great tool for entering and retrieving data *in the field*, but you do not want your data to be imprisoned on your device. You want to share it with your desktop, to be able to backup the data, archive them (reducing the strain on the memory device) or even let you post-process and analyze the data. The big disadvantages of mobile devices to perform major tasks for the end-user are the lack of CPU-power, memory and screensize. For analysis of large chucks of data (gathered with a mobile device) a desktop and a large screen could prove to be invaluable. The bigger screen generally provides a better overview over lager sets of data. For example, trend analysis of your data could be a lot easier to grasp if it is presented on a large screen than on the small screen of a mobile device.

For example, MySportTraining is a workout application that allows you to synchronize all your workouts. On the desktop you not only have more data, but you also have a better overview.



By synchronizing the data with your desktop, the data on the mobile device is not only secured on the desktop, but also you can analyze the trends in the workouts. By purging the data older than (in my case) 3 months on the device, the footprint of the application stays small while I still can make decisions during a workout based on historic data. On the desktop, I can look back about 3 years, allowing me to analyze trends and to plan workouts for longer periods of time.

It is a challenge to find an application that not only suits functional needs, but also performs these functions in a way that is can be perform well in the daily environment it is in. As said, the following aspects certainly are worth considering when you buy an application:

- Does the application integrate well into all other applications, or is there a risk to duplicate information or operations?
- Is the application (unnecessary) dependent on external infrastructure, like the internet?
- Is the user-interface usable under your normal usage circumstances, for example usage during driving, meetings, workouts or other environments?
- Does the application keep your data imprisoned on your device, or does it allow you to backup, analyze and post-process your data on your desktop or central server?