TUESDAY, **NOV 18, 2008**

Mobile applications development: Observations

Technology and democracy: both great in theory, a little harder in practice. One of the key challenges is that one successful model doesn't - by default - work somewhere else. For mobile techies, if we can't easily 'transplant' a solution from one place to another, where does our "figure out what works" mantra leave us? What relevance does it have? Have we ever managed to "figure out ***what*** works" and make ***it*** work somewhere else, geographically? Is it even possible, or is it just a good industry sound bite?

Progress in the social mobile field will come only when we think more about best practices in the *thinking* and *design* of mobile projects and applications, rather than obsessing over the end products themselves. By then most of the damage has usually already been done. In my experience, many social mobile projects fail in the early stages. Lack of basic reality-checking and a tendency to make major assumptions are lead culprits, yet they are relatively easy to avoid. I would argue that successful mobile projects - those aimed at developing countries in particular - have a better chance of succeeding if some or all of the following are considered from the outset.

Understanding your users and their context is key if your application is to succeed (Photo: Ken Banks)

- Firstly, think carefully if you're about to build a solution to a problem you don't fully understand

- Check to see if any similar tools to the one you want to build already exist and, if they do, consider partnering up. Despite the rhetoric, all too often people end up reinventing the wheel

- Be flexible enough in your approach to allow for changing circumstances, ideas and feedback. Don't set out with too many fixed parameters if you can help it

- From the outset, try to build something that's easy enough to use without the need for user training or a complex manual, and something which new users can easily and effortlessly replicate once news of your application begins to spread

- Think about rapid prototyping. Don't spend too much time waiting to build the perfect solution, but instead get something out there quickly and let reality shape it. This is crucial if the application is to be *relevant*

- Never let a lack of money stop you. If considerable amounts of funding are required to even get a prototype together, then that's telling you something - your solution is probably overly complex

- Learn to do what you can't afford to pay other people to do. The more design, coding, building, testing and outreach you can do yourself, the better. Stay lean. These tasks can be outsourced later if your solution gains traction and attracts funding. The more you achieve with few resources the more commitment and initiative is shown, increasing the chances a donor will be attracted to what you're doing

- Don't be too controlling over the solution. Build an application which is flexible enough to allow users, whoever and wherever they may be, to plant their own personalities on it. No two rural hospitals work the same way, so don't build an application as if they did

- Think about building platforms and tools which *contribute* to the solution for the users, rather than one which seeks to solve and fix everything for them. Let them be part of it. Think about how your imported solution looks to a local user. Are they a passive recipient of it, or can they take it and turn it into *their* solution? A sense of local ownership is crucial for success and sustainability

- Ensure that the application can work on the most readily and widely available hardware and network infrastructure. Text messaging solutions aren't big in the social mobile space for nothing. And, for the time being, try to avoid building applications which require any kind of internet access, unless you want to restrict your target audience from the outset

- Every third party the user needs to speak to in order to implement your solution increases the chances of failure by a considerable margin, particularly if one of those parties is a local mobile operator

- Be realistic about what your application can achieve, and wherever possible look for low-hanging fruit. Remember - big is not better, small is beautiful, and focus is king. A solid application which solves one element of a wider problem well is better than an average application which tries to solve everything

- Bear in mind that social mobile solutions need to be affordable, ideally free. Business models, if any, should be built around the *use* of the application, not the application itself. Easier said than done, so try to engage business studies graduates at universities, many of whom are always looking for cool social-change projects to work on

- Leverage what local NGOs (or users) are best at, and what they already have - local knowledge, local context, local language and local trust among local communities. Remember that it's unlikely you will *ever* understand the problem as much as they do, and that it's always going to be easier to equip them with tools to do the job than it will ever be for you to learn everything they know

- Don't waste time or energy thinking too much about the open sourcing process (if you decide to go that route) until you know you have something worth open sourcing. (And, by the way, the users will be the ones to let you know that)

- Don't build an application for an environment where it may politically (or otherwise) never be adopted. For example, a nationwide mobile election monitoring system would need government buy-in to be implemented. Governments which commit election fraud to stay in power are unlikely to adopt a technology which gives the game away

- Consider controlling distribution and use of your application at the beginning. Not only is it a good idea to be able to contact users for feedback, donors will almost always want to know where it is being used, by who and for what. Neglect to collect this data at your peril

- Promote your solution like crazy. Reach out to people working in the same technology circles as you, post messages on relevant blogs, blog about it yourself, build a project website, try and brand your solution, and make use of social networking tools such as Twitter and Facebook. Although your target users may not be present, many are likely to be fairly resourceful, and the more people talking about your solution the more likely news is to filter down to them

- Finally, build a community around the application, encourage users to join and share experiences, and to help each other. Don't be afraid to reach out for additional information, and work hard to keep it active, engaging and growing. Communities are notoriously hard to build, but when they work they're worth it