January/February 2012 $9.95   www.TechWell.com

BETTER™
SOFTWARE
A TECHWELL PUBLICATION

GO! GO! GO!
Managing mobile development

LOW TECH, HIGH IMPACT
The humble sticky note

Creative Agility

$ Ways to Save

Take Advantage of Our Best Value—
VIP CONFERENCE PACKAGE

Includes two Tutorial days (2 Half- or 1 Full-Day Tutorial on Monday AND Tuesday); all Keynotes, conference sessions, and the EXPO on Wednesday and Thursday; all continental breakfasts, lunches, breaks, and receptions; also includes The Testing & Quality Leadership Summit on Friday. Combine with Early Bird and group discount for even more value!

Super Early Bird Offer
Receive up to $400 off the regular conference registration fee if payment is received on or before February 17, 2012.

Groups of 3 or More Save 20%
Register a group of three or more at the same time and save 20% off each registration. To take advantage of this offer, please call the Client Support Group at 888.268.8770 or 904.278.0524 or email them at sqeinfo@sqe.com and reference promo code **GRP3**.

PowerPass Discount
PowerPass holders receive an additional $100 off their registration fee.

Alumni Discount
STAREAST alumni receive up to an additional $200 discount off their registration fee.

Multi-day Training Class + Conference
Attend any of the multi-day training classes and the conference and save an additional $300.

Please Note—We will always provide the highest possible discount and allow you to use the two largest discounts that apply to your registration.

THE EXPO *April 18–19*

Visit Top Industry Providers Offering the Latest in Testing Solutions

Looking for answers? Take time to explore this one-of-a-kind EXPO, designed to bring you the latest solutions in testing technologies, software, and tools. To support your software testing efforts, participate in technical presentations and demonstrations conducted throughout the EXPO. Meet one-on-one with representatives from some of today's most progressive and innovative organizations.

For Sponsor/Exhibitor news and updates, visit **www.sqe.com/stareast/sponsors**

To become a sponsor/exhibitor, please contact **sales@sqe.com**

MOBILE CHALLENGES FOR PROJECT MANAGERS

PART 1: PROJECT FACTORS

BY JONATHAN KOHL

ISTOCKPHOTO

IF YOU ARE NEW TO DEVELOPING mobile apps and you are in a project management or leadership role, you might be in for some surprises. If you are used to developing software for computers, you probably are tempted to take the same approach that you have with other projects. Some concepts transfer directly, but mobile projects have some unique challenges of which you need to be aware.

Device Support

While device support may not be the decision of a project manager, it affects you, and you'll want to be aware that there is a combinatorial explosion of mobile devices out there. It is vital to have a strategy to manage this, so you don't end up in a situation where it will be impossible to meet your public claims.

Take popular smartphone application platforms, for example: Apple iOS, Android, Blackberry, Windows Phone 7, Symbian, Meego, and Bada, just to name a few. Within each of those platforms, there are several models of hardware. For each hardware offering, there are multiple firmware versions. Also, since a smartphone is a communication device, there are contracts with different carriers and data plans with different options. If yours is a client-server application—i.e., it needs to communicate to a remote server to operate—then you also need to consider if you need to use secure or insecure communication protocols. Now, imagine a blanket product strategy like we use for computers: "We'll support iOS version 3 and up" or "We'll support all Windows Phone 7 devices." This may sound reasonable, but if you map out all the versions of everything listed above—platform, hardware, firmware, carrier, network, and security—you are looking at dozens or hundreds of combinations with which to test, rather than the handful you consider with a non-mobile platform.

With computer software, there are a few operating systems and OS upgrades to consider, and we don't really pay attention to hardware that much. But in mobile, the devices are smaller and less powerful than computers, which can have an impact on how your software works. A less-powerful device with less memory than another can have a big impact on how your app operates, particularly if you aren't paying attention to memory and other resource management.

In some cases, different hardware versions are different products that provide similar functionality, as is the case with Apple iOS products. There are different hardware versions for iPod Touch, iPhone, and iPad. All of these are incredibly popular, and if you support one, it makes a lot of sense to support the others. In other cases, such as the Android and Windows Phone 7 worlds, different manufacturers supply the hardware that supports the OS. When Windows Phone 7 was released on November 8, 2010, there were around ten hardware devices that supported it. Sometimes hardware is rushed to market, and some versions are more reliable than others.

Firmware and carrier settings often get pushed down to devices, so backward compatibility is difficult to maintain without a way to store and update multiple versions in the test lab. There are a lot of updates to keep track of if you want to use a blanket statement on device support. It's a good idea to mitigate this by being cautious about what you say you will support. For example, if you already have a web application, you may choose to target web browser support for most popular smartphone platforms and limit application development to one or two smartphone platforms.

Device Procurement and Storage

We take a lot for granted these days when creating computer or web-based software. We all have computers to use, and network infrastructures are available, stable, and perform quite well. When we are planning a project, we sometimes need to buy new servers to use for development and testing, but everyone on the team has access to a computer. In the mobile space, we don't all have mobile devices to use. Furthermore, we don't have a network infrastructure to just plug a mobile device into. We need to buy a contract with a telecommunications company, often with separate data plans, just to have a device that will work the way it will be used. Based on what your company has decided your target platforms for development will be, you also will need to find out what telephone companies your customers use and what technology will be used for data transfer, and you'll need to purchase communication and data plans along with the actual devices.

While we have licensing for hardware and software to deal with on software projects, it can be a bit different with mobile devices. Different kinds of licenses are required for development, and, with some platforms, these can be locked to a particular device and computer. Some platforms are quite restrictive about developing and installing applications on them. If you rush in and start developing on a mobile device, it is easy to end up with that device locked to a certain computer—often the first developer who tried it out.

It is also a good idea to come up with a practice for installing firmware and carrier updates. Unless it's part of your strategy, you don't necessarily want all devices upgraded at once or to have different settings on each device. Without a strategy and a practice, you will end up with a combination of firmware, carrier, and OS versions. It can be difficult to upgrade these devices quickly, since they can require special software that is offered by the platform provider. You'll need to research what configuration tools you have available and outline the steps required for installing development and testing licenses, updating firmware, adding in test SSL certificates if required, etc. Without a strategy for updating development and other licenses and figuring out how many you will need for your project (or possibly for the year), rushing ahead in the short term can have consequences for you weeks or months down the road. There is nothing worse than trying to use devices for testing or demos and finding you can't install software onto them because you've used up all your

development licenses.

Build updates can be challenging due to special tool requirements, development licenses, licensing files for individual devices, and the fact that updates need to be pushed out from a computer to the device. Often, installing updates or new test versions of your software during development requires technical knowledge and specialized tools. It's a good idea to train and to plan and schedule down time when updates are needed. If you have a lot of devices, it can take hours for one person to update all of them.

Having enough devices charged and available for your test team can be difficult. It can be incredibly frustrating to try to work with the devices in the morning, only to discover that they are all out of battery power. You need practices for collecting, charging, and storing devices each night. These devices are small, so they can disappear easily. It's wise to set up a practice for accounting and storage of the devices and cables used to sync with computers and charge them. Labeling devices and cables once they are purchased helps, as does locking away the devices so they don't get stolen or borrowed by other groups. Don't forget to lock away the telephone company contract information for each device as well—for that fateful day that a needed communication and data plan expires and needs to be upgraded.

These devices can also get quite banged up if they are shared, so be careful how you store them each night. If they are all rattling around loose in the bottom of a desk drawer, they will wear out prematurely. Store them in their original boxes if you can, and be sure they are clean before putting them away. On one team I was on, a team member liked eating salty snacks while working, which left salt crystals that actually ate away at the screens of the devices.

It can be tempting to always want to buy the latest and greatest hardware, but that can quickly eat up budgets. Try to encourage a purchasing practice, and be strategic about what you buy.

Store Submission

The Internet has made application distribution much easier. We don't have to worry nearly as much about getting shrink-wrapped software on disks onto store shelves. In fact, most teams don't think about this much at all. In the mobile space, though, your application is delivered to consumers through a third-party online store. These are popping up all over, with the platform creators developing and hosting the most popular ones. Apple has the App Store (synced with its popular iTunes platform), Android has the Android Market, and Blackberry has App World, just to name three. Often, these application stores also require specialized software on your computer to transfer the application from the store to your device. Telecommunication companies are developing their own application stores, as well.

Application stores have requirements, and submitting your application to be hosted and available to your target market but having it rejected can have a big impact on your schedule. It's important to do the following at the beginning of your project:

- Determine what stores you will submit to.
- Learn their submission requirements.
- Ensure your application type will be acceptable.
- Factor in at least one rejection in your schedule, particularly if you haven't submitted before.

Store submission rejections are often a surprise to development teams. The schedule is set out, a target delivery date is agreed on, the team delivers, and then a rejection occurs. This can throw a team into a tailspin. You have to change your application to resubmit, which requires more redesign, development, and testing. In some cases, the application store may not accept the kind of application that you are developing at all. It's important to find this information as early as possible so you can plan for it and develop accordingly.

In part two of this series, I look at some of the human factors that may impact your mobile project and how to plan for them. **{end}**

jonathan@kohl.ca

This article originally appeared on TechWell.com.
Visit http://well.tc/ProjectFactors to post comments and questions for the author and http://well.tc/HumanFactors to read part two in the series.