

**MobiCASE 2010**  
**Workshop on Mobile Software Engineering**  
**Position Paper**

Paper title: *Key Challenges and Emerging Trends in Mobile Software Engineering*

Name: *Sarah Allen*

Role/Title: *President*

Organization: *Blazing Cloud*

Preferred Contact: *sarah@blazingcloud.net*

---

1. **Your experience as a software engineer?** I've developed desktop, web and mobile applications professionally since 1990. In recent years, my focus has been web development in Ruby on Rails, and mobile development. I have been involved in open source since 2004, initially through my work on OpenLaszlo. The most notable commercial software that I've developed includes AfterEffects, Shockwave, Flash Media Server and Flash video.
2. **Your experience with mobile software development?** I've developed mobile applications for iPhone, Android, BlackBerry and Windows Mobile using cross-platform frameworks and developed natively for iOS and Android. I recently authored a book on mobile development, [Pro Smartphone Cross-Platform Development](#), published by Apress.
3. **How does traditional software engineering relate to the engineering of mobile applications and systems?** I'm not convinced that traditional software engineering practices are effective for web and mobile development. I'm seeing a lot of innovation in the industry and in open source with agile processes that are much more effective for the fast-paced evolution of mobile devices and consumer expectations for mobile applications.
4. **What are the distinguishing features of mobile software specification, architecture, development and testing that need special attention, skills, or innovation?** The key differences and biggest challenges are in human interface design and creating the development tools and methodologies that can speed time to market and support the quickly changing landscape of mobile applications.
  - a. Human Interface Design
    - i. **Small form-factor** leads to screen-based approach (more similar to 90s web apps or desktop wizards than the modern desktop or web GUI)
    - ii. **Portability** leads to task-based approach and integrated geolocation, contacts and network connectivity. People expect interactions to be quick and context-aware.
    - iii. **Branded experience** is common in mobile applications, where applications have more in common with their web or off-line visual design than with the native OS, following the trend in web applications, rather than desktop apps.
    - iv. **Privacy** needs to be better addressed
  - b. Architecture / Development techniques
    - i. **Cross platform frameworks** are inevitable -- not lowest common denominator, instead reuse of code for common patterns, surfacing unique strengths of native platform. The diversity of programming languages required to develop natively for the range of devices popular on the market today is not sustainable.
    - ii. **Web Services** for connected applications. Most mobile applications are just a part of a larger service. In thinking about mobile apps, we need to think about an architecture that supports synchronized data, asynchronous mostly-on

communication and applications that span web, desktop and multiple mobile devices.

- iii. **Web UI techniques** are commonly borrowed for native mobile development
  - 1. Layout with mark-up (HTML/CSS)
  - 2. Using URLs for identifying screen layouts and visual state
  - 3. incorporating dynamic languages, such as Javascript and Ruby
- iv. **Mobile web and native capabilities merging** and will continue to become more similar until there is no practical difference. This continues a trend we are seeing on the desktop where most people do not know the difference between a web or desktop application.
- v. **Testing** is an area where there needs to be more attention. There has been some good work in this area: Jasmine, originally developed for WebOS has become popular for web application testing, Rhomobile uses mspec, and Apple's unit testing framework continues to evolve slowly for the iPhone.

**What is the suggested focus and agenda for mobile software engineering research and education?** Every student should be required to build a system that includes a mobile device. Agile methodologies should be taught and practiced starting at the undergraduate level. The privacy implications of context-aware applications needs research.