Modern power constrained devices such as mobile phones require programmers to explicitly manage whether components (including the processor and display) are turned on or off. User defined defaults often give a period of time that a component (e.g., the processor or display) will be on, but after that time expires the device will shut down unless the program being executed actively and explicitly prevents it. Other components, such as the GPS unit, the GSM or wireless transmitters and receivers, and so forth may shut down at any time, even during a call or while establishing a network connection. The operating systems for these devices provide wake locks that allow library and application programmers to force a component to stay active until the wait lock is released. The need to manage these wait locks lead to a rich source of programming errors that result in degraded functionality or the rapid depletion of the device's battery. We will present a taxonomy of bugs that arise from this programming model, and describe initial work on a tool to detect and isolate incorrect uses of wake locks. This is work with Professor Charlie Hu, Purdue ECE.