Risk	L'hood	Impact	Mitigation
Underestimate of kiosk complexity[1]	L	M	All components have been used elsewhere previously. Obtain as much example code as possible. "Proof of concept" kiosk early in development.
Difficult recovery from failure	M	L	Need a failure mode analysis and written, tested procedures + user training
Difficulty in managing airport variant software[2]	M	H	Define clear API to EPPS functions. Off-site facility for stress and regression testing new apps.
Lack of end user (passenger) acceptance, leading to rework	L	H	Early "proof of concept", non-live passenger "trials" in airport etc.
EPPS Authority change scope (e.g. wanting phase II apps in phase I)	L	M	Generally increases likelyhood and impact of all risks, particularly kiosk complexity. Needs customer management + tight contract terms.
Lack of IT expertise / support[3] in airport	M	M	Need to specify IT skills needed early and provide appropriate training or contracted, on-site support.
More integration with GoC immigration system required (e.g. provide suspect information to immigration officials)	M	L	Need to design applications with this in mind, and make sure it is clearly an extra cost to customer! I.e. agree tight scope with EPPS authority.

[1] This is the largest part of the bespoke development and involves integrating a range of peripherals with the Suspect Index Search Engine, probably multi-threaded in an NT environment.
[2] The specification allows individual airports to define (and procure?) their own local applications, but these must run against the core EPPS application.
[3] E.g. the kiosks will need cleaning, checking and restocking everynight.