



Uniface rides on a Rocket

A report from some Rocket Uniface workshops

David Norfolk @Bloor October 2022

I have always taken an interest in Uniface; I first met it back in the last century when the IT group in the bank I was working for was blocking a business-led purchase of Uniface, because it was so much more productive than the IT's chosen C++ based development strategy. Dysfunctional or not?

Anyway, Uniface now belongs to Rocket Software and I've been attending a couple of Uniface workshops, just to see how things are going. Pretty well, I think – morale seems to be good. Uniface is still being actively developed and aspects of the Uniface Agile culture and its eLearning capabilities are being migrated into the rest of Rocket Software. Uniface is a tier 1 product for Rocket, the presenters told me.

Properly designed eLearning is, or should be, a major facilitator for stakeholder engagement in the Mutable Business. Taking a course in something achieves buy-in, reading documentation, less so. And eLearning can be available to any stakeholder, especially if you encourage self-education on the firm's time (with some sort of portable accreditation, perhaps, and the involvement of project management). Learn more about Uniface eLearning [here](#).

The first workshop was about the use of containers and Docker with the current version of Uniface. Later versions of Uniface have always been able to work with containers and Docker, in fact – just as long as you were pretty expert in both Docker and Uniface. Now that containers are becoming ubiquitous, Docker containers are being built into Uniface itself – which makes things a lot easier for Uniface developers and is the sort of thing one expects for an actively-maintained low-code, high productivity, development platform. I am no Uniface coder, but the workshop (with practical coding examples) made it all look pretty straightforward to me.

The second workshop was on Uniface exception handling facilities. As well as its classic error handling (which is prone to human error and can impact performance), Uniface provides a structured approach to exception handling. This is now built into its templates and example code snippets (the technical details are documented [here](#)). I think this Uniface approach to exceptions is good, and well-

structured, although developing exception-enabled Uniface applications is a little different to developing in other languages, and Uniface documents best practice for exceptions [here](#). For instance, you should only process an exception for specific error codes – “catching” every possible error and returning the dreaded “*something unexpected happened, press OK to continue*” is not good practice! Exception-handling has to be designed and tested just as any other part of the application is. For example, in some parts of an application “*customer does not exist*” may be a serious error and should stop processing; but while acquiring a new customer, it may be the expected situation. Deciding whether to stop execution is important – if a purchased item hasn't been entered into the system properly, you don't want to set the price to £0.00 and continue processing the order. And, always remember that exceptions and error situations often go along with worried and stressed customers, especially if large sums of money are involved, so that how you handle exceptions (the user experience) can gain you a customer for life – or lose them forever.

Runtime exception handling is generally a vital part of business automation and is often neglected. Exception handling is partly an aspect of Application Assurance and is an important part of automation that must be built with the same care, expertise and good practice as the rest of an automated system. Often it isn't, perhaps because of unrealistic time pressures, or because developers take pride in the working process and hate rooting around in the mess when things are going wrong. Which is why Application Security attacks often focus on the processing of bad data and error conditions (areas that may not even get much testing in practice, whatever the good intentions).

I think that the takeaways from these workshops are that Uniface is an important part of Rocket Software and is being actively developed. It is still an effective, modern, high-productivity platform for developing business automation in mutable organisations. And that a properly designed eLearning facility is a great idea.

Article Reprint



Bloor Research International Ltd
+44 (0)1494 291 992
info@Bloorresearch.com
www.Bloorresearch.com