

The 6 Most Common Problems in Software Development

(Original creator: aschetse)

Today I would like to take a look at problem solving. Now what is a problem? According to the dictionary, a problem is "a matter or situation regarded as unwelcome or harmful and needing to be dealt with and overcome". Fast. What I will do below is discuss the 6 most common problems in software development - segregated by different stages of a Software Development Life Cycle – and obviously will tell you how Uniface helps in a very simple way.

Problem #1: Requirements Gathering

Garbage in, garbage out... Any design is as good as the completeness or correctness of the requirements. If the requirements are not good, the project will fail. Because people will hate the result. Uniface helps as it is a very agile ; it help developers and end-user to quickly develop a prototype that further clarifies the correctness of the design. And then revise the, quickly built the next prototype. And so on. Prototyping with Uniface is straightforward and simple.

Problem #2: Planning & Estimation

Very often estimations of cost and duration of projects are too optimistic resulting in overspending and a much reduced time to market. And frustration from management, something you don't want. Uniface helps as it simplifies app development by separating the design from the actual technical implication(s). Building an app becomes simple, hence everything else about it becomes simpler as well. Simple.

Problem #3: Development

Moving targets, feature creep: it does happen. Requirements do change. I am not one of those religious nut-bags that will tell you "thou shall not change the requirements, ever". Change happens. Change is upon us. Full Stop. Uniface helps as many objects are defined in one unique place. Which makes Uniface very versatile and a Uniface app very easy to change. Uniface is made for change. Because it's so simple...

Problem #4: Testing

Bug-free software doesn't exist. Learn to live with it...YOLO Uniface helps as it needs 7 times less code than Java. Less code, less bugs. Simple.

Problem #5: Collaboration

Project Management and multi-user development are key processes that need a lot of attention. Uniface helps, as there are no multi-user and collaboration issues. Uniface is multi-user. By design. Simple.

Problem #6: Deployment

One more thing, Uniface runs on many platforms too – from mobile to mainframe and you can actually change from one of those platform to another just like that. Without any redevelopment. It's really that simple. There's just one simple decision you now need to make for yourself: build your next app with Uniface.