How Fast?

I spent a day with our research division recently, and got chatting to one of the chaps presenting a project. Now he'd got it going from idea to mock-up and working-if-you're-careful system in 3 months, and that 3 months struck me instinctively as a very good timeline for developing an innovation prototype system. In some industries (and I'm looking at you, City Boy) 3 days is a long time to develop some mock-up software, but for a reasonably complex idea developed within a larger company, 3 months seems about right. That's pure engineering judgement from a lot of years of experience, but it's interesting to try reasoning that out. So...

There are insightful geniuses out there who see things coming a long time before anyone else, but a) unless they're working for themselves it can be difficult to get the boss to go out on that far a limb, and b) most ideas, even the far-out ideas, are usually of their time and context. So if you've reached the point where you've thought of a new idea, then chances are that your competitors are likely to think of it sooner or later too. Which in prototype terms probably gives you a year to demonstrate to market, at best. Before you persuade the customer you usually have to persuade people inside the company, so add in 3 months getting the people you need and bringing-the-board-round lag, and you've got 6 months tops to go from thought to prototype. And then the money kicks in. If you're running a set of risky innovations developments (and innovations technologies by their very nature are untried and hence risky) then you want to spread that risk over several projects that might not come off rather than a much smaller number of projects that might not come off. The resources that you have available are more than just money: they're people's time, their mental energy, their enthusiasm and the goodwill of their colleagues and sponsors. All of these things wear down if they're maxed out over time... most of them would leave people in small heaps after 5-6 months of sustained effort, energy, goodwill etc, but can usually be focussed and channelled over a smaller period of time. 1-2 months is not enough to really understand all the underlying issues and technologies for a project, or to really get any hardware tested and together (unless you already know said hardware very very well). Which leaves 3-4 months as a sensible timescale. Of course, once a prototype is seen and begins to be accepted, there are usually months or even years of more work to come, but an initial short burst of effort seems to be right.

Well, I tried to explain that. I'm not sure I did very well at it, but I think it's an important concept to chew over - maybe at more length and with less haste next time.