

Software Engineering is a key enabler for Business Innovation (Part 2)

6 January 2015 | **Business Innovation, Software Engineering** | [Wolfgang Emmerich](#)

Reading time: 4 minutes

In my [last blog post](#), I reviewed the many dimensions in which businesses can innovate. The most obvious and visible innovation that a firm can make is when it innovates its offering. This requires the firm to develop and launch new products and services in their markets. Many service innovations are digital or have substantial digital components. These need to be implemented in software systems. Likewise, the innovation in many modern products, such as consumer electronics, cars or medical devices is increasingly achieved using (embedded) software. Such software is rarely available as a shrink-wrapped package and if it were then the innovation would not be very well protected as competitors could achieve the same innovation by licensing the same package. Thus my first insight of this post is that innovation in both products and services critically relies on the ability to design, build, deploy, operate and improve the bespoke software that realizes the innovation.

The speed with which product and service innovations occur has increased significantly over the last three decades and I have no reason to believe that it will level out or slow down again. Those who cannot keep up the pace will perish. A prominent example is Nokia, who had more than 40% of the mobile phone market in 2008 but did not manage to keep up with the speed of innovation achieved by Samsung, Apple and others. By 2012 Nokia saw its market share halved and today Nokia is irrelevant as a mobile phone maker.

Because innovation in software intensive products and services is dependent on software development, the speed with which software projects can be brought to a successful conclusion constrain the speed of innovation. The required speed is hard to achieve in settings where software is produced across different continents and time zones by large teams of engineers with mediocre talent. Instead, successful innovators use small teams of highly skilled software engineers that are co-located with product managers and product marketing leads. To be fast, they use development processes that minimise waste, employ a high degree of automation, detect defects as soon as possible and employ very short feedback cycles.

These feedback cycles increasingly extend to the end consumer. To get innovations in front of product and marketing managers at the end of a fortnightly or monthly iteration is only sufficient if folks really know what innovations the market wants. Others need to use ideation techniques to generate a larger number of innovation ideas and subsequently test these

ideas with customers. The [Lean Startup](#) movement has developed techniques for this. They apply a cycle of idea generation, coding these ideas in software and collecting data about consumer behaviour when using the realization of the idea such that the data can then be used to draw conclusions and derive new ideas. Those who apply this successfully go around the idea-code-data cycle in minimal time, sometimes completing the cycle several times a day during the early stages of the development.

This challenges several long-standing practices, in particular around processes required to acceptance test software and to deploy them into production. It requires a [Continuous Delivery](#) approach that ensures that any build can be put into production instantaneously. To have the degree of confidence required to do this, skilled software development teams automate test, build and deployment process and break down organizational barriers between those involved in software development and operations.

In summary, to prevail in the face of increasingly stiff global competition firms need to innovate their product and service offerings. As the Nokia example shows, being a good innovator alone is not sufficient if the competition is faster in bringing innovations to market. The software development required for the innovation in products and services becomes a critical success factor. Thus, innovation in software development itself, such as Lean Startup techniques or Continuous Delivery tools is a key enabler of competitiveness at the level of individual firms within their network of suppliers and in entire economies.

I'm interested to hear your views. What other challenges might arise in the development of a new product or service?

My next blog entry wants to look at platform innovation. Platform innovation enables firms to create new products and services more easily through derivation from the platform. I want to focus on emerging software engineering techniques, such as software product lines / families and domain specific languages, which support the systematic creation of a product or service platform.