

Is Agile specific to Software Development?

By Allan Kelly, allan@allankelly.net, <http://www.allankelly.net>

The \$64,000 questions, or given the consulting revenues at stake: the \$64 million question.

The short answer is Yes.

Given that Agile is built on Lean and Lean which has been shown to work in multiple domains there is every reason to expect it does. Add to that the fact that Agile and Lean are both built on the ideas of Organisational Learning and, as far as I know, nobody has ever question whether that is domain specific the answer should be Yes.

But, while many of the ideas behind Agile - short iterations, high quality, etc. - are easily transferable many of the tools do not. Try doing test-first development in marketing, JUnit will not be a lot of help. If you take the toolkit perspective of Agile then quite possibly you need a new toolkit - which may, or may not, be called Agile.

In Changing Software Development (Kelly 2008) I suggested that software developers are the prototype of future knowledge workers. This suggestion was based on the observation that much modern knowledge work depended on software tools, e.g. e-mail, wiki's, and now social media like Twitter and Facebook. Software developers, because of their work have, early access to these tool and are early adopters.

For software developers Agile is the process change enabled by these tools and in response to them. As other domains embrace these tools many of the lessons learned in Agile Software Development will be applicable, but not all.

I have heard of Agile and Agile-like processes being used by legal, marketing and publishing groups, and I am sure there are many more examples. I believe that in time Agile will permeate many more fields. The more another field looks like software development the easier it will be for Agile to jump across. Take a look at the principles and ideas described in the early chapters of this book, the more those principles hold in another field the easier it will be to adopt Agile working.

Right now most of the knowledge, experience and case studies comes from the software world. Outside of software development there is work to be done by early adopters. With that work there comes risk and the promise of competitive advantage. Never forget: profit is the return for risk.

While I truly believe much of what I am writing here is broadly applicable outside of software development I prefer keep the ideas grounded in what I know and have experienced. I hope others, with or without my involvement, can show these ideas working elsewhere.

References

Kelly, A. 2008. *Changing Software Development: Learning to Become Agile*. John Wiley & Sons.

(c) Allan Kelly 2013 allan@allankelly.net

This essay is a work in progress. The author welcomes comments and feedback at the address above.

About the author

Allan Kelly has held just about every job in the software world, from system admin to development manager. Today he works as consultant, trainer and writer helping teams adopt and deepen Agile practices, and helping companies benefit from developing software. He specialises in working with software product companies and aligning products and processes with company strategy.

He is the author of two books “Business Patterns for Software Developers” and “Changing Software Development: Learning to become Agile”, the originator of Retrospective Dialogue Sheets (<http://www.dialoguesheets.com>), a regular conference speakers and frequent contributor to journals.

Allan lives in London and holds BSc and MBA degrees. More about Allan at <http://www.allankelly.net> and on Twitter as @allankellynet (<http://twitter.com/allankellynet>).