One of the common misconceptions that gets raised when discussing an agile transition is “agile doesn’t scale”. Needless to say, at Software Education we firmly disagree and we have experience helping large organisations implement “scaled agile” using a number of approaches. Our courses discuss the “scaling” challenge and we work with change leaders to help them identify the appropriate approach for their organisation.

The first question we ask is “what does the organisation mean by agile at scale?” The approach we recommend will depend a lot on the answer to this question, and will be unique to your context.

There are many possible approaches to tackling the “scaling” question, and as always we advocate taking a carefully considered context driven approach to selecting the way you choose to adapt your organisation. What’s right for your team, your organisation, your environment? There truly is no “one right way” – agile adoption is about cultural change and shifting people’s behaviour to more collaborative, cooperative and adaptive ways of working, not about implementing a set of processes or practices.

Consider aspects such as your level of agile fluency (http://martinfowler.com/articles/agileFluency.html) and the elements of context identified by Philippe Kruchten in his “Octopus Model” (http://philippe.kruchten.com/2011/02/10/the-frog-and-the-octopus-go-to-snowbird/) and work with your teams to identify what is needed to best achieve your organisational goals.

There are a number of well publicised approaches to large scale implementation of agile practices; Software Education’s agile team have experience with and knowledge of a number of them, and are able to help our customers identify the different elements which may be useful depending on the organisation’s context.

Here are a few of the possible approaches to the “scaling”, but please recognise that there truly is no single “right” way. Rather look to help your teams BE agile (http://blog.softed.com/2013/08/27/reflections-on-attending-the-agile-mindset-masterclass/) and then find the processes and practices which will help best achieve the organisation’s goals.

- SAFe (Scaled Agile Framework http://scaledagileframework.com/) is a branded approach which has gained a lot of publicity lately. We regularly get asked about SAFe in the classroom and in client conversations.

  We see SAFe as being possibly relevant in very large organisations that are building software-intensive products to sell to defined markets.

  SAFe is for big, tightly interdependent and software intensive products. It can be used if there is a need to build a single very large product which has multiple parallel streams of work which need to be coordinated, with a single reference architecture and tight dependencies between the work items and a need to build underlying technical architecture as part of the project. It may also be useful in internal IT development with very large teams where the products being built have tight coupling and common architecture.

  SAFe also requires that all teams across the organisation are working at the same level of agile maturity, which may not always be the case.
There is quite a lot of controversy surrounding SAFe, as can be seen in this InfoQ article: http://www.infoq.com/news/2013/08/safe.

One of the risks of an approach such as SAFe is that it can seem like a panacea – very prescriptive and not context-aware. Beware the snake-oil salesman who promises you that implementing a defined set of practices will make your organisation work better. If SAFe is implemented in the right context with a clear understanding of the cultural aspects in an organisation, with mature agile adoption it could be very effective. If it is adopted as a way to “buy agile” then we see it failing like any other heavy process-driven approach.

Software Education courses reference this approach and we discuss in the classes the potential uses and pitfalls of this way of working.

- If the need is to have a number of teams evolve an architecture incrementally (such as implementing a Services Oriented Architecture) while rapidly delivering products to market then there are other factors which apply. For this environment one of the end-to-end models such as DAD (http://disciplinedagiledelivery.wordpress.com/introduction-to-dad/), DSDM (http://www.agilexp.com/presentations/DSDMexplained.pdf) or (more likely) an adaption of them for your organisation may be more appropriate. These methods have an identified set of initiation activities which are used to ensure projects fit within a portfolio; they ensure that project teams are enterprise aware and follow enterprise-level standards. This approach allows for growth in teams through geographic distribution, supports partnering and outsourcing, and is highly adaptive in its implementation. This is the model we find works best for many of our customers and is imbedded in the Software Education Agile process, again with the strong caveat that part of the implementation needs to be tailoring the approach to your specific needs.

- Where teams are largely independent, products have very loose coupling and the goal is to speed time to market by having more work done in a given time period, then a straightforward Scrum-of-Scrums approach may be appropriate (http://www.scrumalliance.org/community/articles/2013/june/scrum-of-scrums-running-agile-on-large-projects).

- If the product teams are working on independent activities and there is little or no need for coordination across them, scaling becomes merely an employment/engagement issue with a need to ensure that team members’ skills are constantly evolving to meet the current organisational needs – here we recommend the need for good communities of practice, cross functional teams and strong facilitation skills in the teams. There is nothing particularly special about scaling in this environment, it just requires a lot of trust and maturity in the organisation. Arlo Belshee talks about this approach in his blog: http://arlobelshee.com/post/scaling-agile-the-easy-way.

At Software Education we truly believe the goal should never be to “go agile” or to adopt any particular framework or approach. The goal is to help make your organisation more effective, deliver value to your customers sooner, engage your teams more deeply and help make the world of software development more “humane, productive and sustainable” (the mission of the Agile Alliance www.agilealliance.org). This is what we aim for when we help our customers BE agile, rather than just doing agile, or trying to buy agile.