



Hella and agile innovation

JOHN BESSANT
Managing Innovation



Chapter 12: Agile innovation

There's a paradox about innovation – it's all about change and yet the way we deliver that change needs to be part of a routine set of activities. Anyone might get lucky once but delivering a steady stream of innovations is going to need some form of structure, some repeatable process, some organization. As the noted management writer Peter Drucker pointed out, there is an underlying discipline to the act of innovation, particularly in the context of established organizations¹. Start-ups might be able to get away with improvising and winging it but to survive and grow a mainstream business on innovation requires a lot more structure to the way we handle innovation.

And that's where the problems often start; how to balance the discipline and control over scarce resources with the creativity and inspiration which finding a novel solution requires. It's a challenge which becomes even more difficult when the context changes – with increasing rates of technological change, market demands and competitive pressures for faster development cycles and customers looking for more and more customisation to their particular needs.

The role model for this is often the fast entrepreneurial start-up or small firm, quickly and nimbly dancing round the feet of lumbering giants, who are too old and set in their innovation ways to learn to move to a different tune. Those giants try in their way to recapture the entrepreneurial spark (le) of their youth - but doing so is not easy.

The place where we might expect to find some degree of entrepreneurial speed and flair is, of course, in the formal R&D/design and development function. This is supposed to play the role of being the front-end engine of innovation for the company. Invest in this as the vehicle to drive creativity and entrepreneurial flair, make this the mode of working anyway? Except that it's often not the case. Development projects are just that – *projects* which carry high uncertainty and which we often try to control by planning and prediction. We've got some sharp tools for resource allocation – think SAP, Prince 2, etc. – but the challenge is often that we get enmeshed in the systematic way we move forward. We get there in the end but slowly and safely – and sometimes too late or with a less than dramatic solution which doesn't fully delight the customer or meet their needs.

Project management is all about anticipating and planning for risks and uncertainties and, not surprisingly, its implementation often has a built-in tendency to play safe, to review slowly, to change direction reluctantly. And the larger the scale the more controls we bring to bear, slowing and constraining innovation in the pursuit of safety and quality.

This approach is reinforced by an underlying culture in many large organizations which can remember problems when corners were cut or projects went wrong in the market place. Damage to reputation and direct costs associated with putting the problem right build up a strong resistance to taking risks and further press for control and detailed planning.

At the same time there is a sense that the organization needs to develop greater flexibility, speed, agility in handling innovation projects in a context where market needs, competitor behaviour and core technologies are changing faster than ever. There is a real push towards trying to bring a new style of working to the organization, relearn the tools and techniques of the start-up. The vision here is pretty exciting – with limited resources build a high performing team to deliver an uncertain project within a short timescale, one which keeps investors on board and gathers momentum from initial idea to full implementation. Even a little of this magic dust might be useful – but how to bring this into the mainstream ways of working with quality standards and careful controls? And how to do this in such a way that you can repeat the trick, make a process out of the experience of a single agile project? How to distil the methods and make them a methodology which can be repeatedly applied, trained and shared across the organization?

That's the story of what has come to be called 'agile innovation' – a back to basics rethink of how to get a more entrepreneurial but still controlled approach into innovation projectsⁱⁱ. It's got a fairly long history, dating back at least thirty years as a systematic and documented approach. And the accumulating evidence suggests that it works – when used appropriately (we'll return to that shortly) agile innovation contributes to:

- Higher productivity and lower cost
- Improved employee engagement and job satisfaction
- Faster time to market
- Higher quality
- Improved stakeholder satisfaction

A brief history of agile

Agile thinking began in the world of software development and the big challenge of failure in large-scale projects. Traditionally the idea of developing software followed a 'waterfall' model in which projects ran in sequential fashion through various different functions in the organization. Sales and marketing might identify a customer need which would then be passed to design and then to development, testing, and deployment. The whole process took a long time and there was a high risk that what emerged at the end was less than successful, either at a technical level or in terms of meeting the customer specification. On the way through this journey the project would rely on extensive controls, detailed documentation and project management according to a 'master plan' developed and worked out at the outset of the project.

Some developers began to search for alternatives and in particular Jeff Sutherland and Ken Schwaber began experimenting in the early 1990s with an approach they adapted from work on physical product development in Japanⁱⁱⁱ. Several studies of the processes in operation there - notably by Nonaka and Takeuchi, reported in Harvard Business Review in 1986 – suggested there was a new approach based less on the idea of a sequential relay race' than a 'rugby team' metaphor, in which everyone was moving forward but in parallel and passing the ball along as they did so. This 'new innovation game' had made significant inroads to cutting development time and improving quality in projects^{iv}.

Sutherland and Schwaber took some of these ideas, drawing on the rugby metaphor, to create their ‘scrum’ approach; others worked in different ways on the same lines, with names like ‘extreme programming’. In 2001 at a meeting of like-minded developers they coined the term ‘agile’ to describe the umbrella under which these new approaches might sit. Their work had significant impact on software projects – for example a recent survey by the management consultancy Bain and Co, reviewing ‘tens of thousands of software development projects’ suggested that *‘agile methods boosted average success rates to 39% from 11%, a more than threefold improvement. In large, complex projects agile’s success rate jumps to six times that of conventional methods’*.

Whilst software development was the crucible in which the idea of agile was fused together it drew on many earlier contributing streams. For example, a key element used ideas adapted in a variety of ways around concepts of early involvement, and simultaneous or concurrent engineering. The emphasis was in trying to work in parallel, sharing relevant knowledge early and quickly and making sure all the functions involved could explore the emerging solution. Tools to help included various DfX approaches (design for manufacture, design for assembly, etc.) which tried to bring some of the downstream knowledge to bear at an early stage.

Another key idea was the concept of a dedicated team with the responsibility and authority to explore and experiment – something which had emerged in several studies during the 1990s including Wheelwright and Clark’s work where they identified the ‘heavyweight project manager/team’ model as relevant for major innovation projects^v. In many ways this model dates right back to the early days of the ‘skunk works’, an approach pioneered at Lockheed during the 1940s and based on setting up a small, dedicated team with a strong leader and the autonomy to explore their own (sometimes unconventional) route towards completing a major project within a tight schedule and with limited resources^{vi}.

Another important contribution came from the world of ‘lean’ thinking. This radically different approach to manufacturing evolved out of post-War Japan where a resource-constrained economy forced a new way of working emphasising waste reduction. Through experiments in companies like Toyota a powerful set of principles and tools emerged which provided a disciplined methodology for working in lean fashion. They soon found application outside of manufacturing and one direction in which they were applied was in thinking about the challenge of new product development. Versions of ‘lean NPD’ began to emerge in many companies, and it was also very successfully transferred to the world of the start-up^{vii}.

Lean start-up

By their nature start-ups are resource-constrained operations, often trying to achieve something very different and with high potential impact. Start-ups need to experiment and explore – but if they do so carelessly they will soon run out of support and, more importantly, resources. So there is a strong underlying pressure to learn how to innovate, to move quickly but at the same time to manage and control resources carefully. An extreme version of our paradox from earlier.

Lean start-up (LSU) is an approach very similar to agile which was developed by Eric Ries and popularised by him and Steve Blank in various books and articles^{viii}. It draws on his own experience as

an entrepreneur and his reflections on what went wrong with the process. At its with agile innovation, at heart is the view that starting a new venture is about a series of short fast experiments rather than a carefully planned and executed big project. Each cycle is carefully designed to generate information and test ideas out on the market – and after each prototype the venture idea is adjusted. Key principles are the ‘minimum viable product’ (MVP) which is a simple basic version of the overall product idea which can be tested on users to gain feedback, and the ‘pivot’, which is changes in direction as a result of that feedback.

The origin of the ‘lean’ idea comes from the low waste approach pioneered in manufacturing and widely used across all sectors. It has been applied to product development to reduce time and resources spent and in software in particular has been allied to a second principle, of ‘agile’ development. Here the main project s broken down into a series of fast short cycles of prototypes and learning, with the development team effort concentrated in fast bursts of intense activity – the ‘scrum’.

LSU developed in the field of software and web applications but the underlying philosophy can be applied in any project.

Fifty shades of agile...

Whilst there is growing evidence for the success and value of agile approaches it is very much not a case of ‘one size fits all’. There’s a real risk of organizations jumping into this new approach because it seems fashionable; the evidence is that it will only help when it is adapted and configured in the right fashion.

Agile isn’t a single technique but a system, a way of thinking about projects, an enabling methodology and a toolkit from which organizations can configure their own approach. It addresses many common problems in new product development so a good place to start is by understanding where current systems have weaknesses and adopting /experimenting with agile approaches to try and improve them.

For example major projects need to bring in many different perspectives, involve different functions all of whom contribute their expertise. The result is often a large and unwieldy team, and one in which the risks of things being missed because of the complexity of communication means that quality needs to be checked and controlled on a regular basis. Project planning is heavy at the front end, trying to anticipate all the many possible problems which might emerge and considerable efforts are spent on working with customers to ensure the specification of their needs is tightly nailed down. All the key features are designed in at the outset and then documented – again adding to the time in delivering the project because these need to be captured and detailed. And only when the entire project, with all its features is ready and checked and documented is it tested – and inevitably bugs emerge which need to be ironed out, adding further to the time and budget.

This ‘heavy’ front end serial model involves several stages:

- Idea →

- Features identified and discussed ->
- Design->
- Develop->
- Integrate->
- Test->
- Deliver and deploy

By contrast an agile approach makes use of a small core team (which can still draw on a wider network of expertise but on a 'just-in-time' basis. They are in constant contact with the customer, testing and checking out the development as making sure that they are only building in what is of value to the customer. Design is based on an outline and a module approach which prioritises key design features and can adapt and add as the project develops. Development takes the form of a series of learning loops/cycles testing and fixing as they go and in particular testing out customer reaction and evaluation. Modules are added, features built in a process of rapid short cycle improvements.

It's not a case of replacing the old project management discipline with a small team 'trust us' mentality – agile can deliver the relevant controls to fit, it relies on the same core disciplines of collaboration, testing, customer involvement, etc. But it does so in a more flexible configuration which is particularly suited to certain environments and project types.

Table 12.1 shows where agile might offer benefits:

(based on Bain^{ix})

Table 12.1: Agile and traditional approaches to innovation

<i>Where agile might help</i>	<i>Where traditional models are appropriate</i>
Fast changing environments and customer choice and changes in their needs and wants	Stable markets, technologies, predictable customer requirements
Risk is high, many dimensions not fully elaborated	Risk is low, project is well-understood
Close collaboration where there is learning on both sides, customers discover what they value and what features really matter, co-creation	Customer specification is clear and unlikely to change. Customers may not want or may not be able and available for close collaboration
Problems are complex and solutions unknown – high uncertainty where	Problems are well understood, perhaps an update along a familiar trajectory.

experimentation and learning is the pattern. Relevant knowledge sets not fully defined so need cross functional collaboration/involvement	Technologies are stable and understood, proven solutions are available, relevant knowledge sets are known and projects can run by serial/sequential involvement of different knowledge silos. Project outcomes can be clearly forecast and relevant tests and measures for quality control are available
Project lends itself to breaking down into modules. Customers can have their value in modular chunks, can change and add to their specification as they learn what would be valuable modules to add. Late changes are acceptable to basic core platform functionality. MVP and pivoting are possible and desirable	Project is indivisible into modules so can't be tested and checked until fully built. Late changes are expensive or impossible. MVP is the final product, no pivoting possible
Mistakes can be found and rectified in the next development cycle – fast failure mode and extensive in process learning	Interim mistakes have big negative impact so detailed testing and checking at every stage is required.
Suited to team-based collaborative culture with extensive horizontal /process flow of knowledge	Suited to top down control environments and specialist functional knowledge silos

It follows from this that aligning the wider organizational culture is also a key theme – it's a way of working which can easily suit start-ups but which might conflict with long-established traditions in older bigger businesses. This doesn't mean that there is no scope for agile but that some form of piloting and exploration would be needed to help learn, configure and adapt methods. For many organizations the idea of a 'spectrum' of choice of project methods is useful, running from traditional approaches at one end, through adoption of some agile practices, through to agile execution around key tasks and project phases to full agile. And in most cases there is scope for running 'hybrid' models combining agile strengths with traditional approaches.

What's in the agile tool box?

It's worth lifting the lid on some of the more common agile tools to get a sense of how they might contribute to faster and more flexible product development. Agile innovation has evolved into a suite of methods which have been increasingly applied outside of software development to other new products, services and even process re-engineering. At its heart is an approach which emphasises focused high intensity team work (often called a 'scrum'), stretching

goals and rapid cycles of prototyping, testing and learning. Where conventional project management techniques set a goal and then break down the various tasks needed to complete it into key activities and allocate resources to them agile methods are more open-ended, allowing considerable creativity and flexibility in the execution of activities which will move nearer to the stretch target.

The basic framework in an agile approach involves setting up a core self-managed team, drawing on different functions and with a clear and stretching target. The team use various creativity tools (such as brainstorming and design thinking) to generate a list of key features which they think will be of value to the end user. Two key roles operate – a team leader who represents the end user’s point of view and ranks these features from that perspective, and a process facilitator whose role is to help manage the support and psychological safety aspects of the team.

Once the stretch goal (vision) is broken down into a ranked list of contributing projects the team work on short problem solving cycles (‘sprints’) around these issues. Typically there is a short review meeting at the start of each day to explore progress, challenge and strengthen ideas and develop experiments which they then test out during the day. The results of those experiments provide feedback and data to fuel the next day’s review meeting and drive the sprint forward. Experiments may be of a technical nature – for example writing code or developing a working prototype – or they may be market tests, trying out the ideas with potential end users. In both cases the idea is to move through a fast cycle of experiment and learn, with the prospect of failure seen simply as a learning opportunity rather than a block to further progress.

To enable such an approach there needs to be a core team and they often have a physical space set aside – a ‘war room’ is an analogy often used here – in which progress can be tracked, scrum meetings held and each day’s activities planned and reviewed.

Some useful additions to the agile repertoire come from the world of lean start-up, including the following:

(a) Build-measure-learn

The principle here is to design a hypothesis to test an idea and then adjust the project on the basis of that feedback. So, for example, it can be used to test a particular feature where the hypothesis is that people will like and value it; if they do then retain the feature, if they don’t, drop it.

(b) Minimum viable product (MVP)

This is the minimum configuration of the new venture idea which can be used to run a build/measure/learn cycle – a simple prototype whose purpose is to generate data which helps adjust the core idea for the venture.

(c) Validated learning

An important element of LSU is to work with data which provides useful information and helps learning about the venture. Ries talks about the problem of ‘vanity metrics’ which might appear to be measures

of success but don't actually reveal anything useful. The number of people visiting a web-page for example is not helpful in itself but the amount of time they spend or the features they click on may be because it gives information about the underlying things that people are valuing – at least enough to send some time on. Equally the number of return visitors is a useful metric.

(d) Innovation accounting

Linked to validated learning is the idea of using data to ensure resources are being well spent. To do this requires establishing a baseline and then improving on the performance linked to that by varying elements in the MVP – a process Ries calls ‘tuning the engine’. For example a simple baseline could be set by a market survey which asks people if they would buy a product or service. Then launching an MVP cycle would generate data which suggested that more (or less) of them would be interested – and the core concept could be pivoted before a re-test cycle. In this way the scarce resources associated with innovation can be carefully tracked.

(e) Pivoting

The core assumption in LSU is that the only way to get closer to what customers actually need is to test your idea out on them and adapt it according to feedback from several learning cycles. So there is a need to use data from experiments to adjust the offer – the idea of a pivot is not that you change the idea completely but pivot it around the core so that it more exactly meets market needs. YouTube was originally a dating site on which one of the many features offered was the ability to share short video clips. During MVP tests it became clear that this feature was particularly valued so the original idea was adapted to put this more up front; further tests showed it was sufficiently valued to make it the core feature of the new business venture.

The essence of pivoting and MVP could be summed up as ‘launch and see what happens’ – inevitably something will and if the experimental launch is well designed it will help sharpen and refine the final offering without too much resource waste. Even if the MVP is a ‘failure’ there is valuable learning about new directions in which to pivot.

Ries talks about several versions of the pivot:

- *Zoom-in pivot*, where a single feature in the product now becomes the entire product (as in the YouTube case).
- *Zoom-out pivot*, where the whole product becomes a single feature in something much larger.
- *Customer segment pivot*, where the product was right, but the original customer segment wasn't. By rethinking the customer target segment the product can be better positioned
- *Customer need pivot*, where validated learning highlights a more important customer need or problem
- *Platform pivot*, where single separate applications converge to become a platform
- *Business architecture pivot*, essentially changing the underlying business model –for example from high margin, low volume, to low margin, high volume.
- *Value capture pivot*, where changes involve rethinking marketing strategy, cost structure, product, etc.

-
- *Engine of growth pivot*, where the start-up model is rethought. Ries suggests 3 core models for this - viral, sticky, or paid growth – and there is scope to change between them
 - *Channel pivot*, where different routes to reach the market are explored
 - *Technology pivot*, where alternative new technologies are used but the rest of the business model – market, cost structure, etc. – remains the same

(f) Single unit flow

An idea which originated in the Toyota Production System and is one of the cornerstones of 'lean' thinking. In essence it is about working in small batches and completing the tasks on those rather than working in high volume. Think about doing a mailshot which would involve stuffing envelopes, addressing them, stamping them, posting them, etc. Doing this in high volume one task at a time runs the risk of being slow and also of errors being made and not detected – for example, spelling someone's name wrong. Working one unit at a time would be faster and more accurate.

Applied to LSU the idea is to work at small scale to develop the system and identify errors and problems quickly; the whole system can then be redesigned to take out these problems.

(g) Line stop/ Andon cord

Another idea drawn from Toyota is the ability to stop production when an error occurs – in the giant car factories this is done by means of a cord and a light which flashes above the place where the employee has found a problem. In LSU it is the principle of making sure there are error checks and that the process is stopped until these are fixed.

(h) Continuous improvement

Another Toyota-based principle which is to keep reviewing and improving the core product and the process delivering it. By working in small batches (see (f) above) it is possible to experiment and optimise around the core idea.

(i) Kanban

Yet another 'lean' feature this refers to the system of stock management associated with just-in-time production. Applied to LSU it puts improvement projects around the core product/venture idea into 'buckets' which are processed and progressed in systematic fashion. It is a powerful aid to managing capacity since new projects cannot be started until there is room for them in the system.

(j) Five whys

A powerful diagnostic tool this helps find root causes of problems and directs action towards solving those problems rather than treating symptoms.

Making agile innovation work

Like any good idea the test comes in the implementation; much depends on configuring an appropriate model for a particular organization rather than assuming here is a single 'plug and play' version of agile. Research suggest several areas where agile projects often run aground:

- (a) – lack of understanding. Agile is still evolving and the risk is that it is adopted as a fashion accessory without a deeper understanding of where and why as well as how.
- (b) Lack of skills/ experience – in a survey by Bain and Co 44% of respondents blame failure on lack of familiarity with agile methods with another 35% saying that there are not enough personnel with the necessary experience.
- (c) Lack of management support, often stemming from concerns about losing control of projects
- (d) Agile principles at odds with the company's operating model. As we have seen agile approaches challenge much of the plan and control culture and moving to models like scrum may not fit with the dominant project culture for innovation
- (e) Trying to fit agile elements into non-agile framework

These are not surprising and they suggest an approach which recognises the need to develop an understanding and the relevant skills and to pilot before attempting to spread across the whole organization.

The good news is that there is an emerging pathway to successful implementation with a number of guidelines on the roadmap. These include:

- Build on the principles and adapt/evolve the practices. Just like lean there is a small set of well-proven ideas at the heart and then a variety of enabling tools and techniques, not all of which will work in different situations. It's also about learning to use those tools, building capability around a new way of working with innovation
- Use a pilot and learn approach rather than a 'big bang'. This aligns with the above point about acquiring the skills and capability, plus it allows for the organization to have a 'mixed economy' – not every project fits the agile template. The value of starting small is that a team can be built who can become evangelists and experts to help spread the principles; whilst external support might be needed early on to teach the principles and tools this can quickly be internalised. In addition pilots provide an opportunity to collect data which can help convince sceptics that the approach can work in our particular company context.
- Work with matrix/temporary models of structure rather than change the whole organization. Agile isn't for all projects and there may well be a strong argument for maintaining knowledge concentrated in functions. But within agile teams allow for a different mode of working which emphasises fast cross functional sharing, collaboration, shared experimentation, etc.

-
- Create psychological safety – agile teams work differently and have much in common with ‘skunk works’ – the sense of being able to experiment and fail fast. This requires team members to trust each other but also an external sense that they are ‘licensed’ to experiment and play
 - Build the team – invest in teambuilding to enable high performance creativity. Key in agile is facilitation and support – central role of scrum master. Reward and recognition become more intrinsic
 - Measurement frameworks – may need to adapt to value different KPIs but eventually also need to align with bigger picture
 - Give autonomy and heavyweight project management authority. Key part of the model is the war room and fast focused decision-making – can’t be constantly referring outside and upwards. Clear lines around decision authority and remit and autonomy within them

Why agile matters for Hella

For a company like Hella innovation is essential – it’s one of the first words in the Annual Report, it’s part of the company DNA. But with it comes a need to balance – innovation is about taking risks and exploring but it also about delivering. Much of Hella’s impressive spend on innovation – close to 10% of turnover on R&D and around a fifth of their employees involved in it – is linked to major projects with clients. Big partners in the car industry, big projects in which there is a lot at stake.

For this reason the systems for managing design and development have become increasingly important as ways of controlling and managing risks, ensuring delivery and success. But with this has come an increasingly systematic mind-set, a sense of doing things inside a slow but steady machine.

Outside the world is changing. Technologies have much to offer but rates of change are increasing. Customers can define expectations but then find themselves wanting more, engineering changes, newer features, customisation. The pressure for a more agile approach is certainly there – and it needs to be linked to a context in which Hella also has to find ways of projecting an image of being innovative, not just in its product technologies but in the ways it creates and delivers those offerings.

There are also important external reference points; agile is a ‘hot topic’ in industry conferences and major competitors are making a feature of it in their sales presentations and technology road-shows.

What Hella is doing

Experimenting with the concept began around 2013, not least with the arrival of Michael Jaeger to head up the Electronics Division’s innovation activities. With a background at Bosch he had first hand experience of some of these approaches and quickly found some kindred spirits across the organization. An i-Circle discussion on the theme gave senior managers an insight into the principles of agile – but also rang some alarm bells about the conflict between this mode of working and the current dominant company model. Whatever else, agile was a challenge to a culture where ‘the way

we do things round here' didn't normally extend to small teams working autonomously and emphasising learning fast – even if it meant failing fast! Control and planning was what had built the company's innovation reputation there was too much at stake to jeopardise it with a new fashion.

An internal team of champions began working in 2016 with external consultants to review and help build Hella's capability in agile methods. After three months of working with agile in several project areas in the USA, Germany and Romania an internal survey suggested that it offered significant improvements in:

- Job satisfaction
- Team spirit
- Project visibility
- Management of project risks
- Co-ordination with other teams and roles
- Co-ordination within team
- Work progress transparency
- Productivity
- Development processes applicability
- Quality of work results
- Time to deliver
- Personal skills
- Change and priorities management

So far 14 projects have been completed and there are a further 6 running pilots; the aim is now to capture initial learning and codify Hella's version of an agile approach. Importantly this would provide a roadmap but also flexibility for configuring different versions of agile for different situations – building the basis of a Hella capability in this important area.

Notes

ⁱ Drucker, P., *Innovation and entrepreneurship*. 1985, New York: Harper and Row.

ⁱⁱ Morris, L., M. Ma, and P. Wu, *Agile Innovation: The Revolutionary Approach to Accelerate Success, Inspire Engagement, and Ignite Creativity*. 2014, New York: Wiley.

ⁱⁱⁱ Rigby, D., J. Sutherland, and H. Takeuchi, The secret history of agile innovation. *Harvard Business Review*, 2016(April).

^{iv} Takeuchi, H. and I. Nonaka, The new new product development game. *Harvard Business Review*, 1986(January-February): p. 137-146.

^v Wheelwright, S. and K. Clark, *Revolutionising product development*. 1992, New York: Free Press.

^{vi} Rich, B. and L. Janos, *Skunk works*. 1994, London: Warner Books.

^{vii} Womack, J. and D. Jones, *Lean thinking*. 1996, New York: Simon and Schuster.

^{viii} Ries, E., *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. 2011, New York: Crown.

and Blank, S., *Why the Lean Start-Up Changes Everything*. *Harvard Business Review*, 2013. 91(5): p. 63-72.

^{ix} Rigby, D., et al., *An executive's guide to agile innovation*. 2016, Bain_Management_Consultants: <http://www.bain.com/publications/articles/agile-innovation.aspx>.

Further resources

You can find a number of useful resources – case studies, video and audio and tools to explore some of the themes discussed in this chapter at www.innovation-portal.info

In particular:

- Case studies of organizations taking an agile approach to innovation – for example in the turbulent conditions around humanitarian innovation or in the operations of ‘skunk works’ type groups
- Tools and frameworks (like lean start-up) to help explore themes raised in the chapter

Reflection questions

1. At first sight lean/ agile approaches seem to be essential to enable innovation in organizations. But there are limits to their applicability and occasions when a more structured approach will be of value. Using examples try to identify the conditions under which lean/agile would be appropriate and explain why.
2. Choose an organization with which you are familiar. Where do they need more agility in their innovation processes? Where do standards help and where do they get in the way?
3. You’ve been asked as consultants to advise a company on how it might develop a culture of ‘intelligent failure’ – that is, able to learn fast from experiments. What advice would you give in terms of structures, skills and tools which they might implement?