Aquila Heywood

Agile – changing legacy perspectives

Whitepaper

Adopting **Agile for successful business** change in Life, Pensions & Investments

Contents

A challenging industry

The Life, Pensions and Investment industry deals with long-term policies and customer relationships that extend for decades.

It is no surprise that the enterprise architecture and technology landscape in many Life, Pensions and Investment providers (LP&I providers) is highly complex, with a mix of older legacy applications, new platforms and enterprise utility solutions. Frequent mergers and acquisitions, the drive to digital delivery, cost pressures and mandated market changes all contribute to the challenge of keeping pace with the needs of the business and its customers.

Product providers must find effective methods to adapt to frequently changing legislation and regulations, while reacting positively to new market opportunities and protecting their position from external competition.

The IT functions in LP&I providers also face many technical challenges. Many organisations, Aquila Heywood included, are already committed to implementing Agile within their software development approach. When implemented well, Agile has much to offer to help LP&I Providers meet their IT and business challenges.

Not only is Agile a better 'fit' to the highly flexible, rapidly-changing IT demands in financial services organisations, it also directly addresses the key negatives of traditional waterfall development. No longer is there an extended time gap between requirements and delivery:

- Software is tested as it is built, not months later.
- Project plans are fluid with a focus on short-term delivery of business value.
- Staff are more engaged in the whole development process, feeling more motivated.

Although Agile provides many benefits to LP&I providers, it is **how** Agile is embedded into an organisation that is important in ensuring these benefits are achieved. There are many challenges involved in introducing a new way of working into an organisation, especially when that company also has strategic systems implemented with COTS[1] software partners. These challenges need to be solved before a provider can fully recognise the benefits of using an Agile method.

[1] https://searchdatacenter.techtarget.com/definition/COTS-MOTS-GOTS-and-NOTS

This whitepaper considers some of the key benefits of Agile and explores the challenges of implementing an Agile method into an organisation and how to overcome them successfully.

For the Life, Pensions and Investments market, becoming Agile is crucial to future success!

The Agile solution

The Agile Manifesto and Principles[2] were written and agreed by a number of pre-eminent methodologists in 2001.

There are 12 principles supporting the manifesto that further clarify the Agile focus on customer value, delivering early, seeking feedback and creating autonomous, cross-functional technically disciplined teams.

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	**02** Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	**03** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
04 Business people and developers must work together daily throughout the project.	**05** Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	**06** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
07 Working software is the primary measure of progress.	**08** Agile processes promote sustainable development. Sponsors, developers and users should be able to maintain a constant pace indefinitely.	**09** Continuous attention to technical excellence and good design enhances agility.
10 Simplicity - the art of maximizing the amount of work not done is essential.	**11** The best architectures, requirements and designs emerge from self-organizing teams.	**12** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Figure 1. The 12 Agile principles

[2] http://agilemanifesto.org/principles.html

From the manifesto and principles, a whole industry has emerged of different methods, tools, processes, skills and change management to support companies in their Agile journey.

The Agile mindset focuses on the creation of a Minimum Viable Product (MVP) or minimum marketable feature (MMF), that is, the minimum required to establish a working service or feature and gain feedback from customers and the market. This is then improved and developed further through iterations. The ability to be able to complete this cycle effectively and efficiently requires significant change in the way most companies invest, define, develop and deliver their software to market.

Those who can change effectively will reap substantial benefits in the following areas:

1. Delivering continuous business value from change investment

Product requirements are defined in discrete pieces which can be prioritised based on business value, built incrementally and further improved on a regular basis. This allows LP&I providers to adapt and respond to change quickly, ensuring that value is provided to internal stakeholders and customers in every system update or product release.

2. Solving the brain drain

In many organisations, there is a heavy reliance on Subject Matter Experts (SMEs) who are able to solve specific problems as they occur. This reliance can cause issues if the SME leaves the organisation. Often there is a lengthy time for their replacement to attain the same level of knowledge.

In Agile, teams are formed of individuals from different departments (for example, technology, business analysis or product proposition), who share knowledge and work collaboratively to deliver business functionality. Teams must produce the required output within a short sprint (for example, a two-week period), so whatever skills are required defines the shape of the team. This sharing of knowledge ensures that, when a problem is encountered, more than one person will have the ability to solve it, reducing the reliance on any one member of the team and mitigating knowledge loss during staff turnover.

3. Improving product quality

A key principle of the Agile method is that 'business people and developers must work together daily throughout the project'. Business representatives form part of the Scrum teams, ensuring there is a constant focus on the end-user demand. This thereby makes deliveries relevant and of value. In addition, having testers within each team ensures higher delivery quality, with each product functionality feedback loop taking place in a shorter space of time.

Implementing Agile – ensuring a successful transition

There are many aspects to making a successful transition to Agile. This section of the whitepaper focuses on the following:

- Using Agile to address legacy enterprise landscape issues
- Leadership in an Agile environment
- Security and risk
- Scaling Agile
- Implementing Agile with offshore teams

Using Agile to address legacy landscape issues

In the LP&I market, most large enterprises, whether they evolve from organic or acquisition-driven growth, have a complex, aged, multi-owned, multi-language software landscape.

Strategically, where a system or platform is critical to the commercial success of the business and / or provides the unique selling proposition of the business, the LP&I organisation should have as much control as possible around ongoing investment in that solution.

Key facets in exerting control over the design, delivery and support of critical solutions are transparency, feedback loops and collaboration:

- Transparency of progress and delivery enables stakeholders to understand the situation, through the demonstration of working software. (An Agile principle is 'the primary measure of progress is working software'.)
- Feedback loops provide a mechanism to understand how the solution is received and adopted by target users. To do this, the cycle time to market needs to be short in order to evolve the product to users' needs.
- Collaboration across Product, Technology, Operations, Infrastructure and Customers (internal and external) provides a rounded solution, more effectively meeting the needs of all stakeholders.

"...it is equally imperative to have a close working partnership with the COTS vendor..."

With many organisations utilising COTS packaged solutions as critical elements of their enterprise estates, it is equally imperative to have a close working partnership with the COTS vendor, that is itself Agile. This is to ensure the key tenets of transparency, feedback and collaboration can be adopted between the partner organisations.

When deciding how to approach the challenge of Agile adoption, the focus should always be on creating value by assessing the factors of control, cost and 'Go to Market', that is, the best business value will be obtained by prioritising Agile adoption on core systems / solutions that the organisation needs to control and that are critical to support the organisation's main product lines and go-to-market propositions.

By understanding these factors, a company can start to understand its software landscape and make appropriate decisions about what it needs to control and what can be commoditised.

Once the enterprise legacy landscape has been understood, there are three steps to consider in developing a programme to move to Agile.

1. Monetise the landscape.
2. Turn off the tap.
3. Create better code.

1. Monetise the Landscape

By assessing the value created by the software in the enterprise landscape, (both the current and future value), informed decisions can be made about where the highest areas of risk are and which parts of the software landscape should be top contenders for investment.

For all systems where control is required, the way the software is created and evolves is a strategic, mission-critical discussion. Often non-technical executives will challenge the need to invest in software production processes and tools. It can therefore mean that a lot of time is spent educating and communicating with peers in the wider business. This communication is best supported by 'monetising the landscape'; This involves having an Architect provide a view of the application landscape and assigning revenues, costs, and operational risk numbers to that picture. For example, an

e-commerce solution provides revenues, a call centre solution provides cost reduction and efficiencies, and a trading platform that supports assets under management represents significant operational and financial risk. By monetising the landscape, all senior stakeholders can have a joined-up view of each system's value to the business. This aids prioritisation and the case for investment into core systems.

2. Turn the Tap Off

Generally, when a company has a large complex software landscape, people are aware of the legacy problems and are taking remedial steps, that are often palliative in nature. As a metaphor, the problem could be described as finding your basement completely flooded, but with a line of willing volunteers in a row with buckets, emptying out the basement to get rid of the 'technical debt'[3].

Everyone is pleased that something is being done about the problem. Then someone points out that, while it's great that the water is being emptied, no one has turned the tap off!

In terms of a software landscape, this means that software is still being created in a non-disciplined way and so the underlying issue is not being solved. Without changing the skills, processes, tools and environment within which software is created, the organisation will end up with the same result. Hence, once a key application is identified, the key tactic for dealing with the technical debt is to 'turn the tap off' and stop code being produced using older, failing methods.

3. Create Better Code

So how is the tap turned off? By introducing disciplined programming techniques such as pair programming, test-driven development (writing unit tests first), continuous integration, clear coding standards with static analysis, gated builds and build monitors for visibility. In addition to this, a strong, scalable infrastructure for the development and test environments is needed. All of this requires investment in the automatic provisioning, configuration and deployment of environments, as well as automated tests at each step in the process to ensure the system / solution has not regressed.

Finally, in order to turn the tap off, the changes in programming, infrastructure and testing techniques need to be embedded into a culture of continuous improvement and quality first.

Turning the tap off at scale is not a small activity, but completing this task is a start toward a less complex, more visible and less risky environment, that can better meet business needs.

[3] https://www.agilealliance.org/introduction-to-the-technical-debt-concept/

Leadership in an Agile Environment

Leading in an Agile context requires a considerable mindset shift for executives, moving away from a focus on cost and time and instead toward a focus on value-creation and empirical feedback from customers.

Through empowering teams, responsibility and decision-making are naturally shared, which can be uncomfortable; the cultural shift towards a more meritocratic and flat-structure can also be disconcerting. However, the benefits of establishing team-based delivery and shortening cycle times to market are huge for executives.

It enables a more flexible approach to investment decisions and greater visibility and transparency of progress and delivery.

There are significant cultural and structural changes required to truly maximise the value of Agile methods. These changes are explored to a greater extent in our blog series: 'Leading in a changing context in the 21st century'.

Traditionally, in many companies, there are multiple layers of management, which can lead to an employee mindset that climbing the ladder is the most important focus, rather than excellence in their role. When advancement and reward is based on the number of employees that work for you, the natural behaviour is to seek more employees to manage – to empire build, and protect that area or silo.

By moving to the Agile framework, the shape of this process is restructured and a change in behaviour occurs:

1. Work Distribution

As development teams commit to smaller chunks of deliverable business value called User Stories, this empowers the team to take ownership of the work. The team determines the criteria under which a shippable product is 'done', and the team is collectively responsible for delivery.

In order to understand how long a piece of work might take, teams assign 'points' to each piece of work, allowing the team to understand its 'velocity' and provide forecasting information based on what they can deliver in a fixed time period. Once each story is completed, the points for the stories are added together to provide an indicative pace for the team's performance.

In this way, self-organising teams are established that take collective ownership for delivery.

2. Management of work

Using the 'Scrum' method, work is not monitored in the same way that a project manager would traditionally do. Instead, it becomes a collaborative item. All work in the development pipeline is prioritised by a Product Owner[4], whose role is to seek and prioritise value and represent the customer within the team at all times. The role of a Scrum Master[5] is to help the team be successful by whatever means necessary; that person teaches and encourages the use of Scrum, coaches the Product Owner and removes impediments from the team's progress; in effect, to become 'Servant Leaders'.[6]

Unlike Project Managers, Scrum Masters are not responsible for scope (this is with the Product Owner) nor are they responsible for the delivery (this is with the team). They are responsible for the overall improvement of the 'system'.

Although Scrum Masters still provide a leadership role in guiding the team through Agile frameworks and principles, there is a switch in management style from 'telling' to facilitating, guiding, coaching and encouraging.

3. Self-Organisation and Improvement

To ensure every team member gains knowledge and experience in different areas, improves their skills and the quality of the product, Agile techniques such as 'Pairing'[7] are encouraged. This involves using group work to solve challenges and develop software functionality. At the end of each two-week development period (known as a sprint[8]), a retrospective takes place. This meeting, facilitated by a Scrum Master, allows the team to determine actions that will lead to improved efficiency.

Another method to encourage team improvement is by increasing recognition and learning opportunities. This can happen during retrospectives, where the team declares which members have been most appreciated for their efforts or can be at a wider scale through systemic organisational team-based recognition and reward systems.

[4] https://www.scrumguides.org/
[5] https://www.scrumguides.org/
[6] https://www.greenleaf.org/what-is-servant-leadership/
[7] http://www.extremeprogramming.org/
[8] https://www.scrumguides.org/

Security and Risk

Conceptually, in shifting from a traditional software development lifecycle (SDLC) to an Agile approach, the IT Director is devolving the responsibility for delivery to their teams. In doing this, rather than attempting to control the discipline of the activity through project management, the discipline is shifting to the team.

Agile methods such as Scrum and Extreme programming are about creating value in rapid feedback loops using extremely disciplined development and testing methods.

This fundamentally reduces risk, and has evolved over the years to respond more proactively to security and protecting systems, products and services.

Security is increased and risks reduced in an Agile environment due to the adoption of test-driven development, continuous integration and static code analysis. Tools that analyse and report back on the state of the code base in real time or at build times (which are continuous) are now mature and include security-based rules covering areas such as OWASP[9]. There are also automated tools dedicated to security analysis that can be integrated into the build pipeline. These tools provide immediate feedback to developers on security risks, allowing the proactive removal of any potential security risks.

Agile pipelines include 'infrastructure as code' that is, the automatic provisioning, configuration and deployment of virtual environments, associated pre-requisites (such as OS & Database) and the software application itself through the use of scripts. Adopting this sort of automated pipeline, along with a Security Incident & Event Management (SIEM) tool, also significantly reduces risks in the environment.

[9] https://www.owasp.org/index.php/Main_Page

Further Protection

Penetration testing by an independent, external body remains an essential part of the software delivery process but, rather than treat this as an 'outside' issue to overcome, a more integrated approach should be adopted. When establishing penetration testing policy and moving to rapid deployment cycles, all the security tests within the lifecycle should be considered, to provide a holistic approach to security that is understood by all participants:

- Product
- Development
- Testing
- DevOps

By integrating security into the culture and processes of development, using static analysis, automated tests, DevOps processes and automation, internal threat analysis and scanning and automated penetration testing, companies can establish a defendable security position.

This will significantly mitigate the risk of small releases. This solution, coupled with a regular annual or quarterly external penetration test, enables rapid deployment cycles to take place within a secure, defined and auditable policy. In addition, implementing encryption at rest, encryption of application traffic, DDoS mitigation filters, security tooling and increasing the depth and breadth of monitoring can further de-risk the environment. These changes can be made at 'relatively' little cost and without impacting the development process.

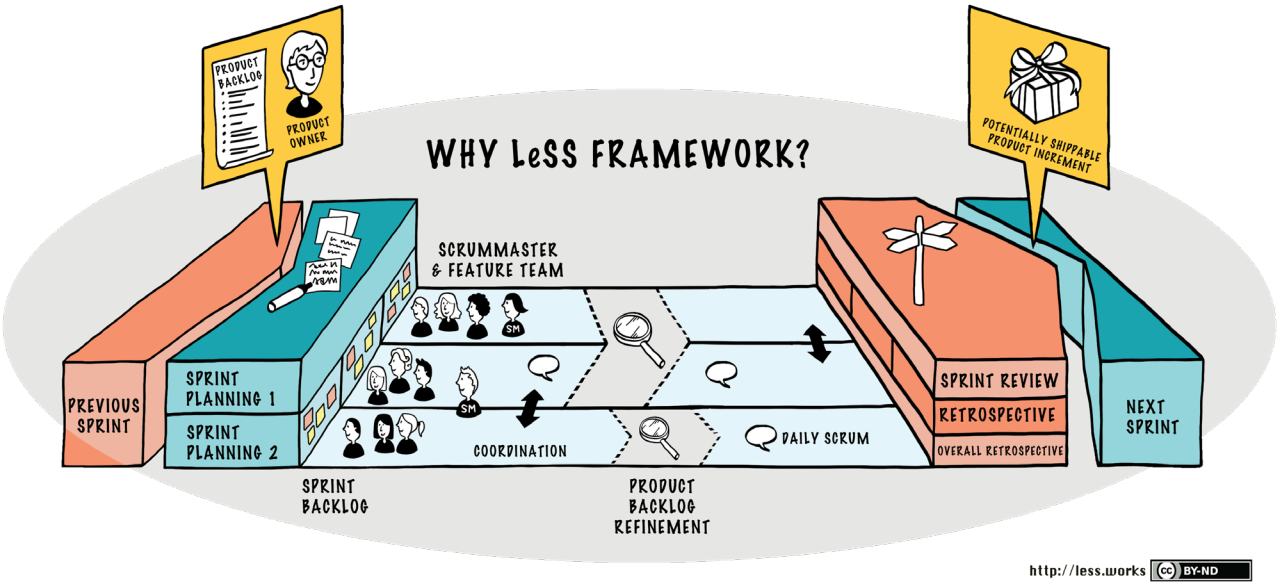
Scaling Agile

One of the biggest challenges for any organisation when implementing Agile is scaling up the Agile process, particularly when applying the Agile method to 50, 100 or 300 developers.

Scaling Agile has a number of challenges, some in direct conflict with the Agile principles. However, several companies have shown success, or at least a demonstrable improvement, after scaling Agile and a number of lessons have emerged on how to approach the problem.

These lessons have evolved into Scaled Frameworks such as Large Scale Scrum (LeSS)[10], Nexus[11], Scaled Agile Framework (SAFe)[12] and Scrum at Scale[13]. Each of the frameworks has evolved from slightly different perspectives and should therefore be seen as a toolkit, where certain parts are used that fit the problems, structures, governance, customers and market, that is, the organisation's context.

At Aquila Heywood, the Large Scale Scrum (LeSS) framework has been adopted.



[10] https://less.works/

[11] https://www.scrum.org/resources/nexus-guidereference

[12] https://www.scaledagileframework.com/

[13] https://www.scrumatscale.com/

Regardless, of whether a specific framework has been adopted, there remain some common problems to overcome to scale Agile capabilities across the organisation:

1. Complexity

There are many facets to creating working software and ensuring it is delivered at the right time, to the right people, to meet the right need. Both product (functional complexity, technical complexity, domain complexity and usage dynamics) and delivery complexities (budgetary constraints, resource constraints, resource capabilities and delivering in a timely manner) need to be considered. Scaling Agile to a wider range of people adds another level of complexity to this list.

Recruitment, induction, domain understanding and integration into the programme all mean that a decision to scale an activity is a long-term decision, a strategic one that recognises that, to have scale, there will be a percentage increase in administration. This is a necessity and has to be planned for. It is imperative to understand that, when scaling Agile, Increasing the number of people working on the programme or activity will not increase the output over the short term.

2. Product Ownership

Scrum[14] is the most predominant and well known of the Agile methods, and for a single team states that there should be one named Product Owner. The role of the Product Owner is to steer the Scrum team toward the highest-value, lowest-effort outcomes to satisfy the customer.[15]

Once it is decided to use Scrum at Scale, then the issue of scaling the Product Owner capability arises.

Each of the various scaled models has a different approach to solving this issue: the LeSS model suggests that the role of Product Owner is one of coordination, communication and decision-making, whereas the other models all use a hierarchical method of some kind. The core issue is that the Product Owner's role within Agile is essentially directing the investment in the product or service. However, for most large organisations, the responsibility for investing significant levels of resources in an activity is not held within a single role. It is, therefore, a significant challenge to establish a single Product Owner role. A solution to this challenge is to create a Product Owner team.

[14] https://www.scrumguides.org/

[15] https://www.youtube.com/watch?v=502ILHjX9EE

Product Owners are crucial in maintaining a prioritised backlog for software. Internal and external customers have competing demands. For most large organisations, it is critical that there be a business-shared view on priorities. This ensures that just one or two Product Owners can coherently prioritise a single product backlog, focusing on prioritising the valuable activities and communicating with all stakeholders, and leave the solution detail of the stories with the Scrum teams.

However, while it is important that all stakeholders meet regularly to discuss the backlog, only a few people should assume the Product Owner role of managing it. Handovers between stakeholders, Scrum teams and customers then become clear and consistent. Product Owners collaborate with teams, regularly connecting them with the end-user perspective. In the LeSS framework for scaled Agile, less is certainly more when it comes to product ownership. A balance should therefore be found between having one backlog to avoid a chaotic pipeline and just enough Product Owners to solve the workload problem of building a complex product across many teams.

Some important points to observe when scaling the Product Ownership model are:

- Investigating the investment model for the portfolio (how are money and resources made available) and whether the process can be shortened or improved
- Ensuring the Product Owner structure understands what value means to the business and that the structure is able to articulate it clearly
- Having a suitable tool to help maintain a single view of the truth that is visible to all stakeholders

The Product Owner should also have:

- The ability to communicate with all stakeholders effectively, relevantly and in a timely manner
- Passion for the product or service and the vision and energy required to execute it
- A close customer relationship and a close partnership with the teams delivering the outcome

3. Internal Governance process

One of the most difficult and contentious items to change when scaling Agile is the investment governance process within the company, that is, how and when does a company decide on its budgetary spend and how does this work in an Agile environment. Often companies plan their investments as part of an annual cycle and require up-front design and estimation activities to support a business case.

This invariably leads to a long lead time for ideas to come to fruition as products or services. This is because they have to be funded through an annual cycle. Each of the scaled models mentioned earlier has attempted to address this issue, but there is no set blueprint. Fundamentally, the company will need to find a way to release investments to fund projects with a higher level of ambiguity, at a smaller and more frequent level. This usually leads to a keen focus on Product Ownership and a significant empowerment within that role to ensure product decisions are made that inherently carry an investment cost.

4. Cross-functional, multi-skilled teams

A Scrum team is self-organising and a cross-functional structure. In the LeSS framework, cross-functional is defined as:

> "Group of people with a clear purpose representing a variety of functions or disciplines in the organisation whose combined efforts are necessary for achieving the team's purpose" [16]

This means that Business SMEs, Developers and Testers must be on the same team to achieve a Scrum team's goals; additionally, within the LP&I world, teams must also be multi-skilled in terms of product knowledge. When developing complex LP&I solutions for the enterprise, the competing demands of stakeholders from different departments can test cross-functionality and multi-skilled Scrum teams.

The challenge is that stakeholders may often steer their specific resources to their own needs rather than towards the objectives of the cross-functional team, sacrificing this element for short-term gain elsewhere.

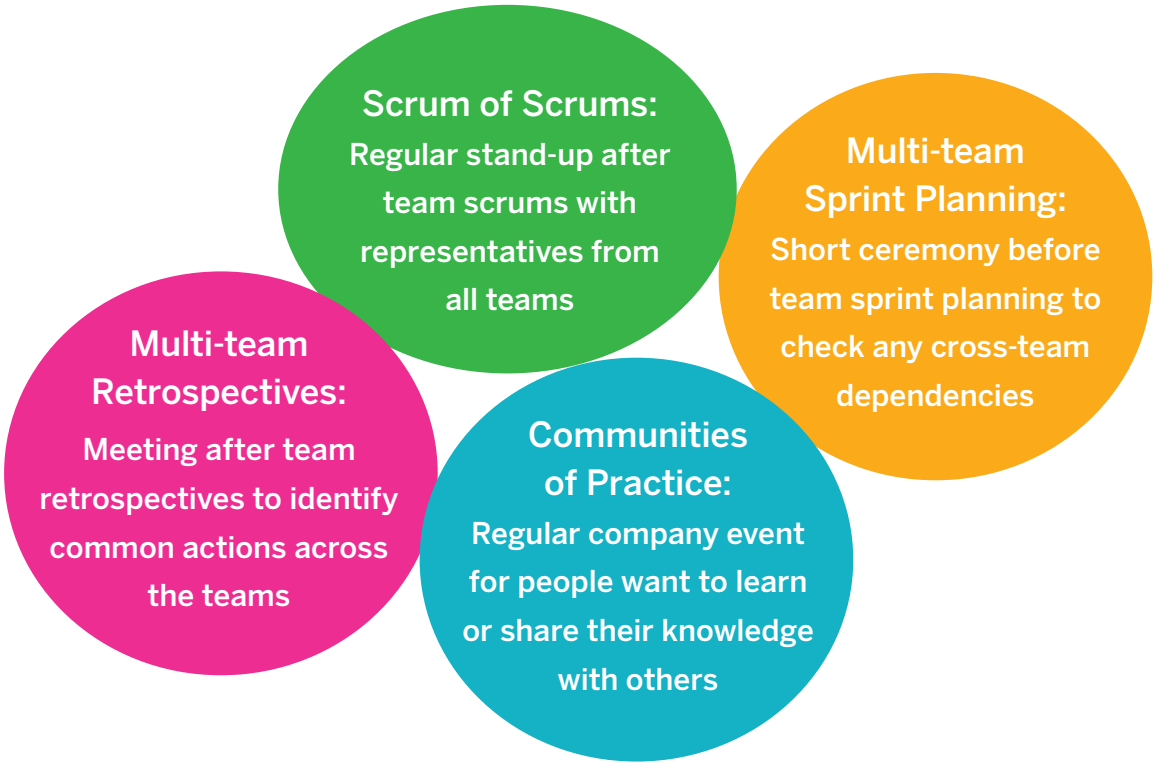
[16] https://less.works/

In terms of allocating work to teams, the natural reaction when assigning top-priority items is to assign each item to the team that has the most knowledge around the subject. In the early days of scaling up an Agile method, this might seem a sensible starting point, but it will add risk of creating silos of skills even more so when it is not just the same team doing certain pieces of work, but actually the same individuals on a team doing the work.

Over time, the strategy should be for each team to develop all the functional skills needed to produce a shippable product; it also needs a wider product appreciation, so that a team may pull any item from the top of the Product Backlog, regardless of the domain area. Cross-functional teams and multi-skilled teams compliment a single product backlog, but it must be accepted that it will take time to build up the same level of knowledge across all teams. It is a balance between developing new skills across teams and making the necessary practical decisions to meet time and commercial obligations.

To help achieve the right balance, the LeSS framework encourages cross-team co-ordination in an informal way decided by the teams, which can be translated into a number of cross-team events. This helps spread knowledge and awareness of issues that affect all teams, encouraging the creation of multi-functional teams.

Some examples include:



Offshore & Agile Adoption

Scaling Agile and encouraging Agile adoption becomes even more complicated when considering offshore teams or offices. The nature of software development is to digitise the services, interactions, processes and data of the real world to achieve a goal or purpose.

In order to do this, development teams need to establish a strong understanding of both the business and technical domains. Without strong technical domain knowledge the solution is more likely to lack quality, be poorly designed and have a higher total cost of ownership. Without business domain knowledge, the solution is less likely to meet the needs of the business goal or problem the team is trying to achieve or solve. A key challenge remains within Agile adoption of how to ensure offshore teams are strong in both the business and technical domains, particularly with higher than average attrition rates, for example, in India - 14%.[17]

Adopting Agile approaches with offshore teams is a complex undertaking and a significant part of that is identifying how to enable some of the 12 core principles of Agile[18] for teams that are often split across thousands of miles and challenging time zones.

Luckily for anyone considering implementing the Agile method into offshore teams or offices, many of the 'lesser' challenges of offshore Agile development have been solved. Compared to 15 years ago, the networks, environments and tools available to offshore teams have improved dramatically, and culturally many offshore teams have learnt to adapt to the time-zone differences, with compromises across all sites to ensure project teams have 'core-hours' for collaboration.

However, implementing all the 12 Agile principles requires effort and investment for both onshore and offshore teams. When specifically considering offshore teams, alongside the challenges listed in the scaling Agile section, decision makers should consider how to address the following principles:

- Business people and developers must work together daily throughout the duration of the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Simplicity - the art of maximizing the amount of work not done is essential.

[17] https://www.firstpost.com/business/attrition-in-india-higher-than-global-average-whats-the-problem-839443.html

[18] http://agilemanifesto.org/principles.html

Table 1. Illustrative tactics that can be employed to help offshore teams live the 'Agile principles'

Invest in the working environment

Agile offshore teams need:

- Strong, stable, secure WAN for availability of environments
- Agile delivery tools: Atlassian; Rally & Version One
- Collaboration tools: Instant Messaging & Video Conferencing
- Engineering coaching & tooling
- Test coaching & tooling

Flexibility in working hours

A little earlier in the UK, a little later in Asia and investment in travel to ensure teams can fit a face to an email address and can have a personal relationship that enables the professional relationship.

Build knowledge sharing into the programme cadence

Examples include:

- 3 sprints of delivery, 1 week to retrospect and refine the backlog (co-located at offshore base)
- Lunch & Learns & Communities of Practice
- Structuring Product Ownership to focus on establishing business domain knowledge

Establish structured career development capability for off-shore teams:

- Technical coaching
- Pair programming (incl. remote pairing)
- Competency frameworks
- Personal Development plans
- Training investment

Ultimately, successful offshore Agile approaches requires investment - significant investment. A company's ability to adopt Agile approaches with offshore teams successfully often comes down to the pressures, culture and context of the company, and its ability to embrace and effect change.

Conclusion

Our own experience has shown that Agile adoption is a challenge, but one that can be successfully achieved with clear benefits being obtained. Even with proper guidance and support, it is not easy. Overcoming staff resistance to change, obtaining management support and buy-in to overhauling the organisation's processes and dealing with existing legacy systems are all challenges to overcome when implementing Agile. This is even trickier for LP&I product providers, who are large, international, complex corporations where customers are serviced across multiple products and multiple channels, and whose teams stretch across countries. However, as we have set out in this whitepaper, there are solutions to the challenges of becoming Agile and multiple benefits once Agile is implemented.

Adopting Agile allows the Change and IT functions of a LP&I product provider to be more responsive and flexible to their business's changing needs and expectations, and to adapt more readily to a market that is in constant competitive flux. Ultimately, that enables the business to meet the needs of its customers better and thereby to thrive.

Agile is the preferred approach whatever the technology and change challenge, be that modernising legacy platforms, meeting the latest regulatory requirements, addressing competitive threats or adapting to emerging customer needs. The quicker feedback loops and creation of a minimum viable product allows product providers to ensure they are truly meeting the needs of their customers and maintaining high-quality services.

Aquila Heywood is well versed in Agile methods, both for our own product developments and also when working as a partner with product providers implementing our Life, Pension and Investment COTS platform solutions.

We are happy to share our experiences. Take advantage of an 'ACT engagement', which is our defined process through which we discuss and debate best practice with our prospects and customers. We would be delighted to walk through the practical steps an organisation can take to move to Agile development and delivery, particularly when also using COTS packages as key elements of its strategic landscape.

For more information, contact us at enquiries@aquilaheywood.co.uk