CAPITA

Using Agile techniques in a distributed environment

Adapting to change; delivering successful outcomes for the business

Organisations delivering large scale IT programmes have come up against the same problems time and time again. They struggle to adapt to change, deliver late and over budget while the end solution is all too often rejected by the end user because the business landscape has changed significantly.

One of the key issues facing them is that most are still using methodologies similar to the traditional waterfall delivery which is better suited to projects where the requirements are known and stable up front and change is infrequent through the lifecycle.

However this is rare in today's changing world. In order to be competitive and manage complex business challenges, most programmes need to work to increasingly aggressive delivery timelines and promise an uncompromising feature-set evolution while still coming under constant cost pressure. This has led many to explore new ways of working; using methodologies such as Agile that are aimed at making IT delivery more responsive and adaptable to change while delivering more frequent releases and early benefits.

The choice of using Agile methods will impact on the way that quality assurance and testing is undertaken. It will impact on structure of the team; the setting of specific quality objectives and adoption of risk-based approaches. Key deliverables need to be clearly defined and progress reported using dashboards and meetings to empower teams to make decisions with regular user feedback.

Often Agile programme streams are integrated with Waterfall / V model programme streams which means designing and implementing delivery frameworks and governance processes to ensure they both synchronise, integrate and deliver across dependencies harmoniously. While the benefits of Agile can be brought to bear on traditional models there is a need for increased communication and collaboration between developers, testers and business users as well as new governance and assurance issues to consider. This needs further consideration when programmes are delivered in a distributed environment.

Constraints of working in distributed environments

Due to the size and nature of today's modern IT projects, many run across distributed environments, whether it is in multiple UK sites, across continents or a mixture of them both. This can present a number of opportunities. Organisations can attract the best talent irrespective of location; costs can be kept down by locating offshore and there are opportunities to increase the length of the working day.

There are also disadvantages. Communication can be more difficult because of differing time zones and dissimilar team cultures. If core business, development and test teams are distributed over multiple locations, knowledge sharing and open discussion between the teams can be hard. These difficulties will be compounded if tools and processes also vary between the sites.

Typically work is passed from discipline to discipline, so for instance an analyst passes their analysis outputs to another analyst in the next centre, they do further work on it, and then pass it on. However, this tends to shatter one of the key underlying principles of all Agile related methods - that of the close coupling of analyst, developer and tester! This model also generates enormous overheads in communication and wastes effort in overlapping tasks. However, all is not lost as there are ways of adapting Agile techniques to work across different sites and timescales.

CAPITA

Adapting Agile to a distributed environment

Being able to identify how some of the more adaptable principles of Agile can be used outside of a pure Agile environment is a key success factor.

For instance, where work items are assigned to a single team, these can be broken down into smaller modules and allocated to tightly coupled Agile teams reporting into a single management layer. Each team, which is usually located in a separate geographic location or centre, focuses on building a module using Agile principles.

In this way each team is contributing to the product backlog. Each module then needs to be integrated which is performed by a central integration centre that performs a 'rolling' integration. See Figure 1.

To support this way of working, a guiding principle is that there should be a single instance of the delivery tool sets, version management and release management tools. Centralised code repositories, databases and a defect repository should be available for all teams to access wherever they are located globally. This ensures that there is a common, single instance of software, tools and code that all teams are utilising and working with. This reduces integration issues across the projects and enables all teams to work more effectively.

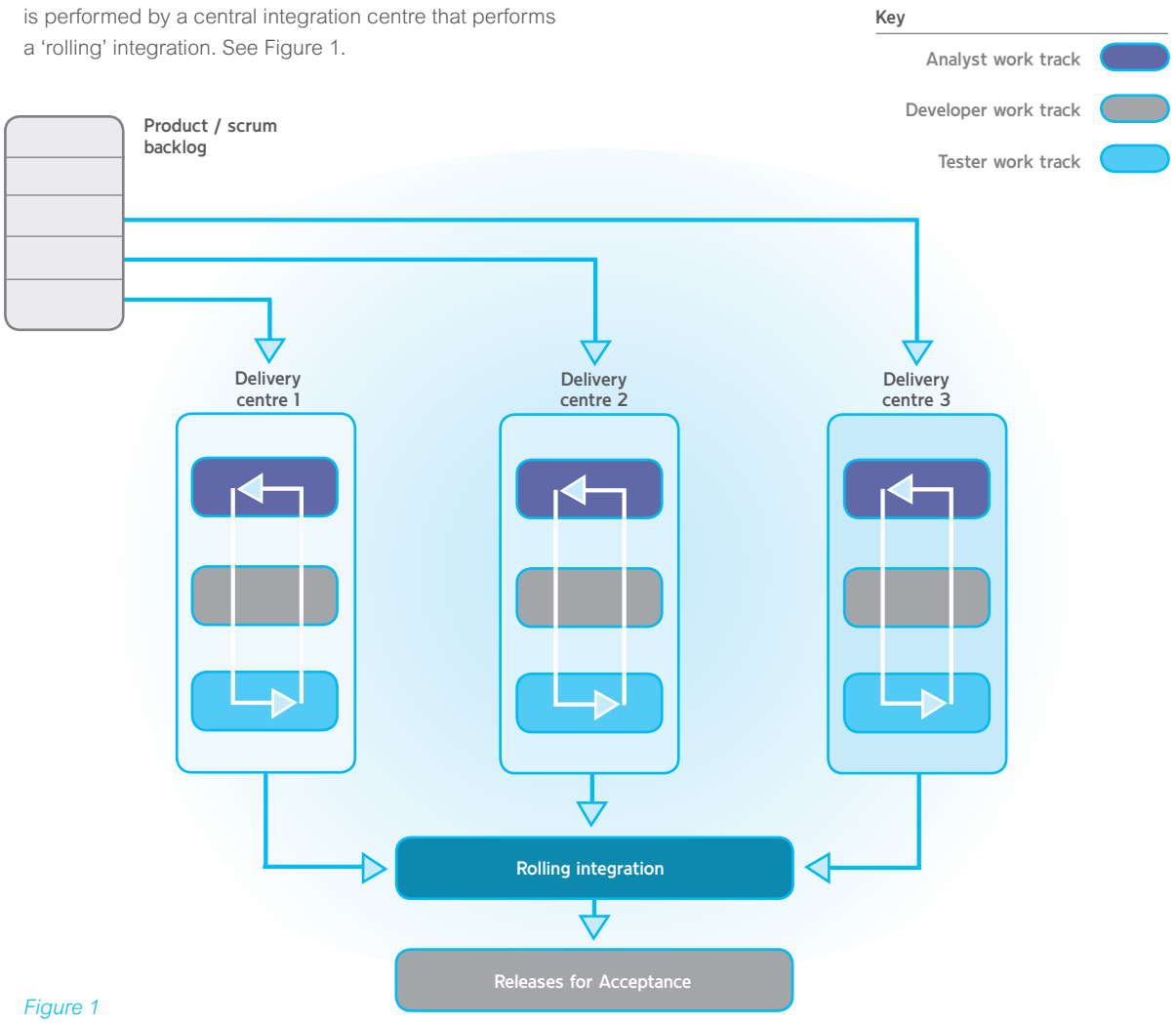

Figure 1

CAPITA

Delivery of a large multi-product migration

The project included the transformation of 20 separate insurance products and migration of over one million policies using teams that were located across multiple sites within the UK and India.

Small Agile teams were created across these sites, consisting of Developers, Business Analysts, Testers and Users. The delivery was broken down at transaction level across all products and the work was divided amongst these teams. Each team iteratively designed, developed, built and tested in line with an overall delivery plan.

Telepresence sessions were held regularly in order to review status and discuss any issues. These virtual 'face-to-face' conversations helped communication and removed some of the barriers usually experienced within such a project.

The programme management team worked across the programme, ensuring all deliveries were aligned and key governance principles were upheld. A central integration centre was created within a single site, whose responsibility it was to continually integrate and test the modules.

At key points the integrated packages were reviewed by end users to ensure the product still aligned to the initial require-ments module. Where the product deviated from the requirements, or where new requirements or changes were identified, these were then easily fed back into the development stream and quickly delivered within the next iteration.

Once the product had been built, full systems integration could be performed. This was undertaken using distributed test teams, adopting the 'follow the sun principal'. This allowed for a faster, continual integration of the completed product.

How to adapt and improve processes to achieve the results you are looking for and reduce the risk of failure

There is a shift in the way senior executives are adopting Agile. It has moved away from 'delivery speed and associated cost savings' to it being able to better support 'change' and therefore to be more responsive and successful in delivering the required business outcomes.

Putting the time and effort into getting requirements right at the start will pay back dividends in the long run. This is one of the key points at which it is useful to bring some remote team members to the main location to enable face to face meetings to happen and knowledge transfer to take place.

Including domain experts within each team and ensuring constant interaction between the teams and with users enables the implementation to be flexible and fit the nature of the organisation. It also ensures that early feedback is provided to enable mid-course corrections to deliver better business outcomes.

Good communication is pivotal to achieving the advantages that Agile techniques can bring to a distributed programme. It is possible to replace face to face communications with remote technical options such as video conferencing, instant messaging, web cams and applications sharing, summary notes such as using a Wiki or SharePoint. Daily reporting is needed to keep the team up to date with status, however these sessions need to be short yet informative. This includes regular touch points with users to ensure that delivery is meeting requirements and if not empower the users to make decisions about the design to allow for change.

A useful team building technique to use in a distributed environment is to adjust working hours to enable some common working hours for the distributed team so that video conferencing can take place at these times. It is particularly important to ensure that a remote location is not left out of any communication loop but this needs to be accomplished in a natural, informal way. Pairing of colleagues or 'buddying' across locations will also drive the need to work together and Wikis enable everyone to contribute and share their experiences.

As time zones shift it is essential to ensure that everyone has enough work to occupy their working day while being conscious of the implications of scheduling tasks across all of the teams.

CAPITA

The role of testing

An Agile approach enables testers to adapt and respond to change. As we have seen, the role of testing in Agile development is less about being quality police and more about working as part of the team to deliver what the business wants and needs.

Testing can be used as a means of engaging with the business right from the start to really get requirements right and collaborate closely with users to define the criteria for acceptance. The requirements are worked into functional tests while keeping users engaged throughout the end-to-end process. It is important to liaise with development and the business to ensure that the systems design is understood and which elements of the build can be prioritised. Tests can be automated where possible to enable a faster turn around and consistent levels of quality.

The responsibility of testing in this environment is crucial to the success of the programme and delivering the required business outcomes. Testers not only need good all round communication skills, they need a good understanding of the underlying applications; to be able to participate in release or iteration planning; to provide statistics about the level and coverage of testing. They also need to provide input and rapid feedback to developers on implemented changes while making sure any defects can be identified and quickly resolved.

A distributed environment makes collaboration and communication challenging but, provided Agile techniques are adopted in a structured way, the benefits will generally outweigh the difficulties as the need to be adaptable within complex distributed change programmes continues apace.