



Quality in agile teams - testing in the driver's seat

An insight into quality in agile teams

October 2017

[KPMG.com/in](https://www.kpmg.com/in)

01

Agile adoption – Introduction



- 1.1 Drivers for agile adoption
- 1.2 Agile methods and practices
- 1.3 Barriers to agile adoption
- 1.4 Fully agile or just fast waterfall
- 1.5 Why agile projects fail

page 01-08

02

Agile – an antipattern to testing?

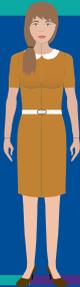
- 2.1 Agile an antipattern to testing
- 2.2 Deconstructing testing in agile
- 2.3 Testing maturity in agile teams

page 09-11



03

Quality in agile teams



- 3.1 Improvising quality in agile teams
- 3.2 Measuring quality success in agile
- 3.3 Automation tooling
- 3.4 Management tools

page 13-16

04

Key takeaways

page 17

05

Summary

page 18



Executive summary



As digital transformation continues to play a critical role in the IT landscape, the software development lifecycle too is transforming

While the agile adoption rates have been steadily climbing, there has been a paradigm shift in the focus of agile teams. With digital initiatives driving technologies such

as cloud, analytics, the Internet of Things (IoT) and mobility, testing has now evolved into business assurance rather than an end-of-cycle activity. Agile adoption has

doubled over the past five years while an improvement in quality is a major driver for agile adoption.

Automation leads the way for agility

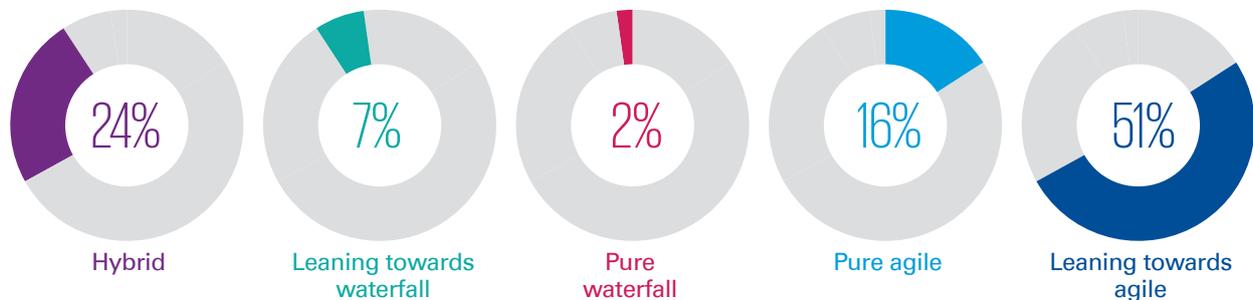


Testing is now at the forefront of the development cycle adding the most business value to organisations. More agile teams are adopting test-driven development and acceptance test-driven approaches. As organisations are embracing agile, there has been an 85 per cent⁰¹ increase in test automation over a two-year period across all industry domains.

01. KPMG in India's research and analysis 2017 based on KPMG testing survey



1.1 Agile adoption - Introduction



Source: Tech Beacon survey - Is agile the new norm?
Accessed on the 14 September 2017

Agile adoption percentages

Software development landscape

Today's software development lifecycle is more outcome oriented when compared with traditional development lifecycles such as the waterfall development methodology. The pace of change today is so unrelenting that agility has become completely critical in order to respond to the customer's demands and rapidly changing market landscape.

The idea of an approach where the designs are made upfront, even before any code is written is seen as risky in today's times as the feedback is slow and does not allow for changes quickly. Lately, adoption of agile development methodologies is seen as a remedy for most problems that besets the development groups in an organisation.

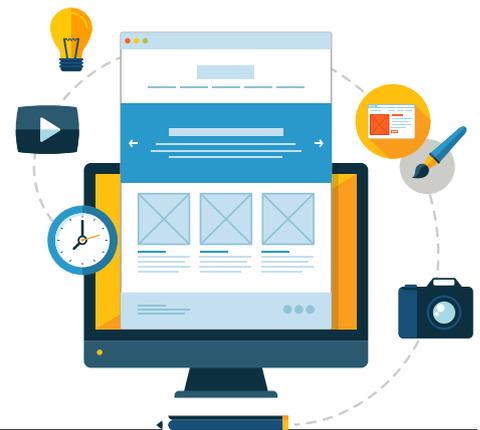
Agile software development originated 15 years ago and has been seen a worthy opponent to the ailing traditional software development cycles. The agile methodology was recognised in 2001 at a summit of practitioners who found consensus around core values captured in 12 principles called the '**Agile Manifesto**'.

Adopters of agile strongly believe that this new development methodology is centred on the customer and helps teams collaborate better to deliver higher business value. Though the adoption of agile has been fairly slow, it has gained momentum in the past five years.

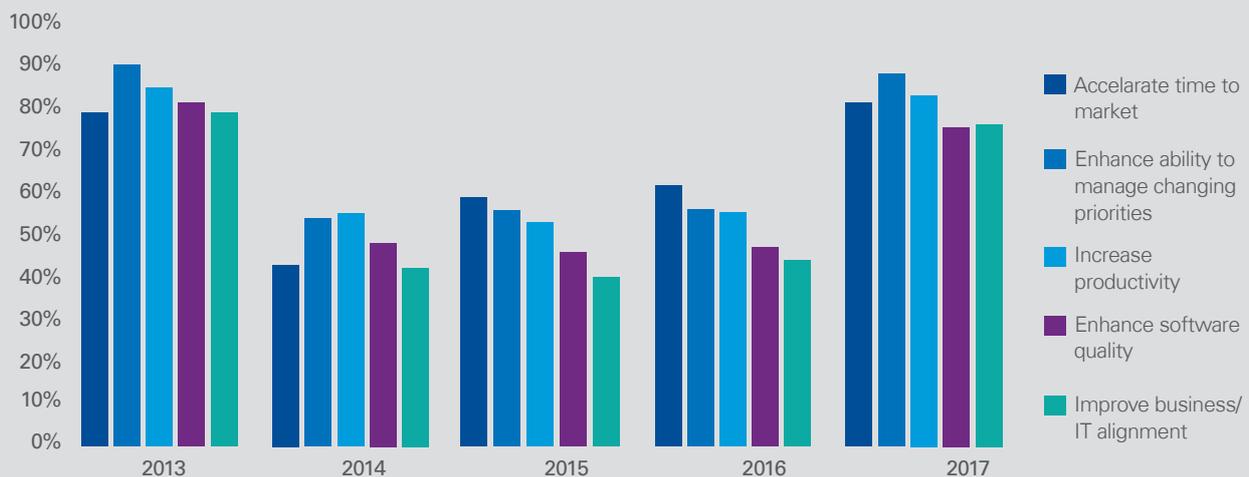
With the new values, practices and benefits - agile is seen as the unconventional alternative to the traditional command-and-control-style management. Agile methodologies are spreading across a broad range of domains and functions and even into the C-suite from broadcasting to leading agriculture machines developer to producing fighter jets to logistics providers to wineries for their warehousing to senior management.

Adopting agile methodologies is not only seen as a means to accelerate revenue growth but also help create a new breed of competent professionals by taking people out of their functional silos and putting them in self-organised and customer-centric multidisciplinary teams.

Lately, adoption of agile development methodologies is seen as a remedy for most problems that plagues the development groups in an organisation, but where does testing and quality fit into the agile conundrum? Will the puzzle make a complete picture with the current focus on development and engineering? There has been a paradigm shift in the focus of accepting and adopting agile across the organisation - that shift from engineering and development to testing and quality outcome. With approaches such as test driven development and Acceptance Test Driven Development, organisations are now accepting that testing should be in the driver's seat rather than just the backseat.



1.2 Drivers for agile adoption



Source: State of agile survey, 2013,2014,2015,2016, 2017, Version One accessed on 14 September 2017

Drivers for agile adoption

Why have so many organisations moved to agile? What were the factors that pushed organisations to adopt agile methodologies?

The top five reasons for organisations to adopt agile five years in a row has been to accelerate time to market, ability to manage shifting priorities, increasing efficiency and to improve software quality. Most responses were centred on better customer focus, enhanced quality and increasing predictability across the delivery lifecycle. The other major benefit with adopting agile is widely seen as the ability to deliver higher value work by helping create

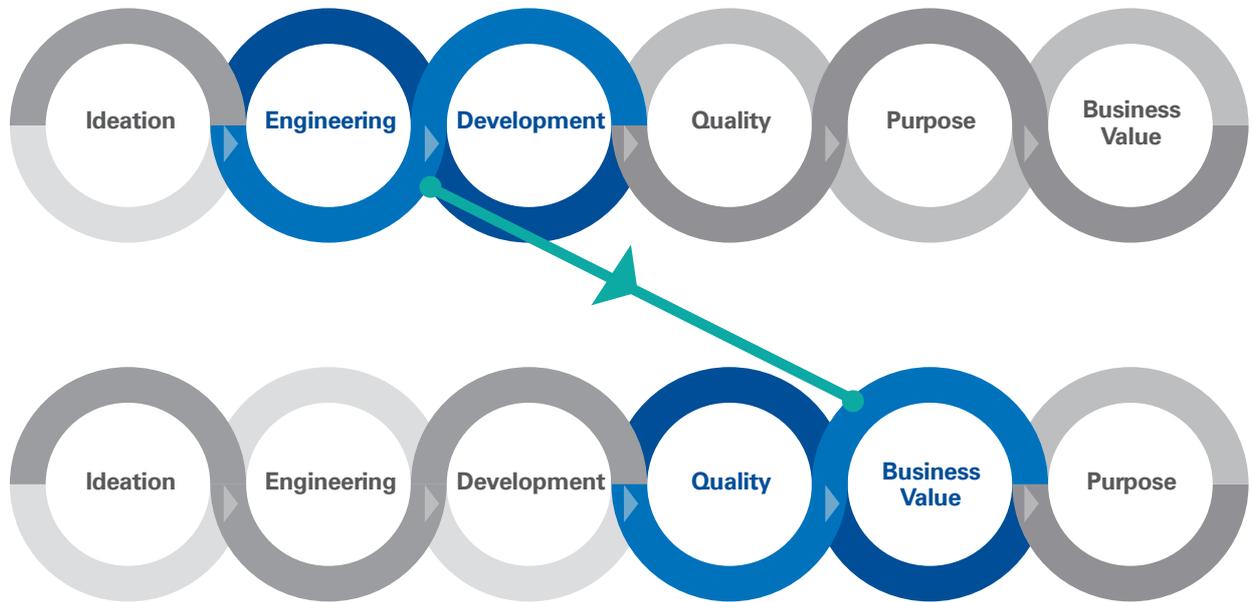
corporate vision; prioritise strategic initiatives; simplify and focus work; assigning the right people to tasks; all while enhancing team collaboration and helping teams raise the bar on quality consistently.

Even as organisations are looking to embrace agile for various reasons, most still feel the learning curve is steep and consider that their organisation was at or below a 'still maturing' level. As organisations ramp up their game, responding to customer expectations and responding to increasing market demand are the major drivers for change.

With the implementation of agile methodologies, organisations feel that there is more freedom to adapt to the changing requirements as compared to the waterfall development practices where the user requirements were frozen early on and the customer may not have access to quick feedback loops. With the adoption of agile, teams within the organisation also feel empowered to deliver business value in relatively short amounts of time.



1.3 Agile 2.0



With the wide acceptance and adoption of agile methodologies across the broad spectrum of industries, the focus of agile has also slowly evolved over the past five years. The reasons for adopting agile initially were faster time-to-market and to increase productivity amongst teams. The reasons for adopting agile has now pushed the focus back on quality and customer satisfaction. The primary motivators for agile adoption continue to be -faster time-to-market, increasing software quality and customer satisfaction. The centre of the viable, profitable cosmos is no longer the organisation: it is the customer. Agile, therefore, has a massive effect on how an organisation is run and how the organisation is

able to respond to change.

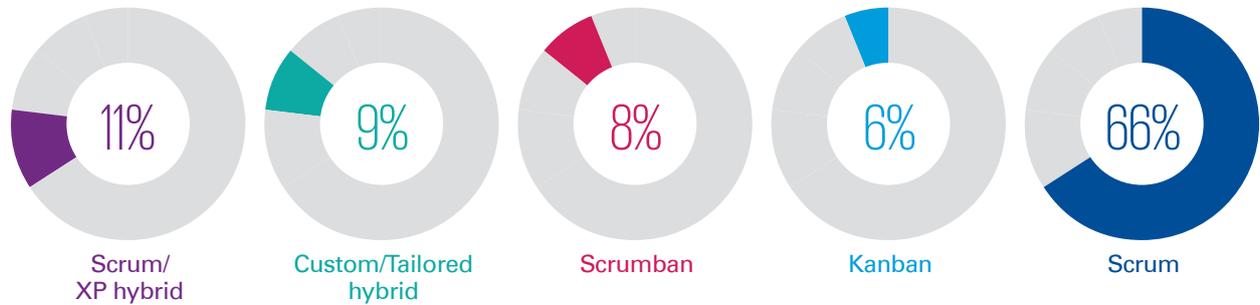
Factors such as faster time-to-market, better software quality and customer satisfaction and not productivity gains in terms of improved engineering discipline, were seen as the leading benefits associated with the methodology. Over the past five years the focus has changed from engineering practices to testing and quality.

Initially, agile was focussed on unit testing and acceptance testing, while the functional testing did not appear to play an important role and was assumed that it could be completed by the developer itself. There was to be a cultural shift for the traditional system tester to be part of the agile team.

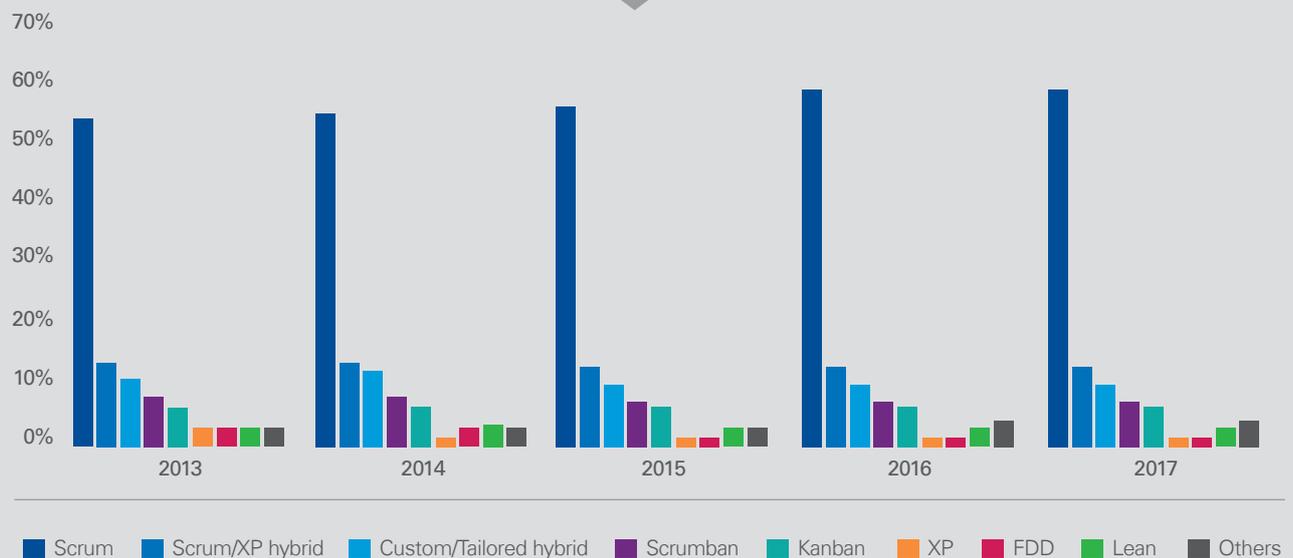
As agile matures, the focus on quality has been reinforcing the importance of the role of the tester and its beneficial impact on the agile team.



1.4 Agile methods and practices



Key agile methods and practices



Source: State of agile survey, 2013,2014,2015,2016, 2017, Version One accessed on 14 September 2017

Scrum and Scrum/XP Hybrid (68 per cent) continue to be the most common agile methodologies used by respondents' organisations. Adopting scrum in combination with the desirable features of other agile processes is seen as more beneficial among agile practitioners.

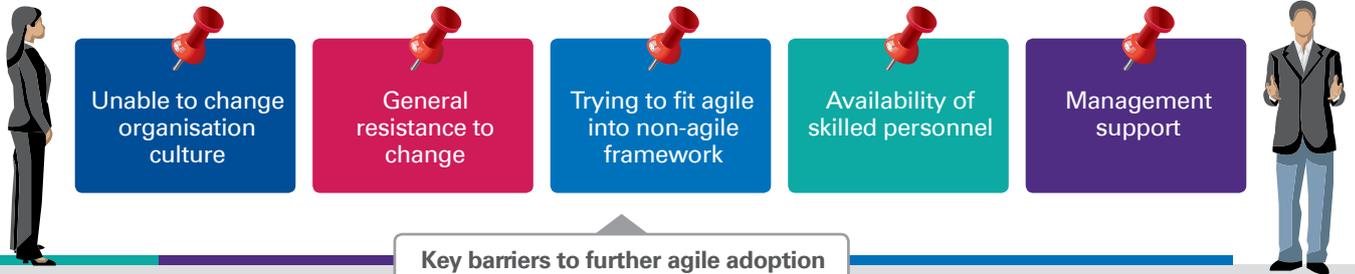
While the usage of XP as an independent methodology continues to decrease (<1 per cent), the practices associated with XP are still prevalent.

Though Scrum, Kanban and XP popular agile methodologies, all three methodologies follow the principles of the Agile Manifesto that aims to provide maximum business value to customers in shorter iterations. The differences between them are due to the methodologies trying to uphold the agile principles in fundamentally dissimilar viewpoints.

Scrum is seen to be apt for teams who can dedicate their shared time to a project or product. Scrum allows teams to decide on how

they want to engineer solutions, helps them engage productively with the customer through frequent communications. Kanban could be appropriate for teams dealing with continuously fluctuating backlog of items. Kanban helps with limiting the intake of backlog items while ensuring efficiency. XP adds another level of intricacy, bringing a strong focus on quality by maintaining a set of core engineering practices which aims to keep the code clean and software stable.

1.4 Barriers to further agile adoption



Source: State of agile survey, 2013,2014,2015,2016, 2017, Version One scssessed on 14 September 2017

Barriers for agile adoption

One of the biggest factors that is inhibiting organisations to adopt agile fully is the inability of the organisation to accept and adopt change - the organisation's DNA does not allow change and there is a general resistance to change.

Key barriers to adopting agile methods and practices are:

1. Ability to change the organisation's culture
2. General resistance to change
3. Trying to fit agile elements into non-agile framework.
4. Availability of skilled talent
5. Management support.

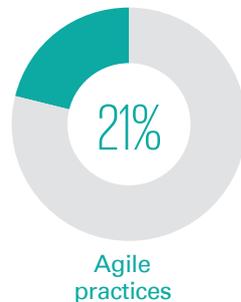
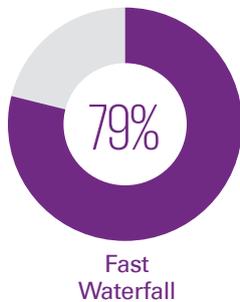
Though the main priority for business leaders continues to be agility, they are unclear on their organisation's goals and purpose. Business leaders need to prioritise the most important business goals rather than focus on multiple initiatives, as executives may launch multiple initiatives with urgent deadlines where nothing actually gets done. Albeit with their best of intentions, they erode benefits that agile innovation can deliver, all while falling back into the vortex of old practices.

The unavailability of skilled personnel to empower self-

organised agile teams is also seen as a factor inhibiting agile adoption. Finally, the lack of management support to usher in new changes – in terms of structural changes, cultural changes, etc. leads to the failure of agile adoption across the organisation.



1.5 Fully agile or just fast waterfall



Source: State of testing survey 2017, accessed on the 15 September 2017

Fully agile or Fast waterfall

Fully agile or fast waterfall?

The broad adoption of agile has not solved the pressures for constantly faster releases with better quality. Organisations still want to deploy even faster with better quality. The timeline for deployments and releases has only gotten smaller from months to weeks to daily to hours.

Though 79 per cent⁰² of the respondents are said to have adopted agile in some form but only 21 per cent consider themselves fully agile. The factors that encourage teams to consider themselves fully agile are:

1. Writing tests whenever code is checked in
2. Frequency of releases
3. Speed of defect fixing
4. Automation percentage.

The most important best practice for testing in agile is that code is tested as soon as it is written. The goal for a fully agile team is to achieve small iterations, where features are broken down into small testable requirements to speed the identification and fixing of bugs while the code is still fresh in the developer's mind. Unfortunately, only a very small number (24 per

cent) of organisations that use the agile software have achieved this micro-iterative approach. But for most agile teams (59 per cent) it is still the standard for individual features to be completed before being passed on to the Quality Assurance (QA) team, which essentially leads them back to the traditional waterfall way of testing. When the tests are not written whenever code is checked in, rather when it is written after completion and passed on to the QA team it is just fast waterfall and not truly agile.

02. State of testing survey 2017, accessed on the 15 September 2017



1.6 Agile projects failure

Census fail

- The 10 million dollar⁰³ contract was awarded by the Government to conduct the online census, but the combination of inadequate non-functional testing (performance and security) took the systems offline for two days
- The heavy traffic load crashed the router, and there was a backup – but in configuring the backup, vendor hadn't conducted a full power-down-reload test.
- The vendor and its network providers had to block traffic that didn't originate in the country - but that did not happen
- An audit across its 400-strong technology staff found it needed to train more scrum masters, product owners and team members - the failure was largely due to a lack of skilled personnel and the lack of focus on non-functional testing⁰⁴.

Healthcare release

- Healthcare website is an example of software development gone wrong. The original budget was USD93.7 million, which grew to USD292 million prior to the launch of the website. The final cost of building the website was close to USD2 billion⁰⁵.
- While the front end employed Scrum, it missed the second principle of the Agile Manifesto: 'Working software'
- Regulations pertaining to large government contracts, are not a good match for agile software development
- The project involved 60 different organisations, managed by government employees instead of a lead contractor
- Lack of a clear test strategy and a lack of a state-wise release plan.

Siren

- Another more recent example of a U.K. government agile project failure for a crime, intelligence and custody suite. The official report blamed the inexperience of the in-house team in the ways of agile development along the governmental project management and internal controls
- This was seen as a failure of agile adoption at every level along with the the gap in communication between external and internal teams and a lack of transparency when considering the timelines of the project (eight years).⁰⁶

Welfare reform project

- High profile failures such as the U.K. government's welfare reform project – were examples of high profile and costly (estimated final cost GBP2.4 billion⁰⁷) project failures
- In the case of the welfare reform project, the National Audit review of the project clearly indicated that the failure of the project was an agile adoption failure with work culture failures mentioned
- Secondly, a failure of procurement and contracting, reinforcing the hearsay and rumours. One of the major problems noticed was a lack of transparency and a failure of performance reporting.⁰⁸

03. Australian Bureau of Statistics, accessed on 15 September 2017

04. ABC Australia, accessed on 15 September 2017

05. Cio.com accessed on 19 September 2017

06. BBC accessed on 15 September 2017

07. IT governance UK accessed on 15 September 2017

08. computerweekly accessed on 19 September 2017

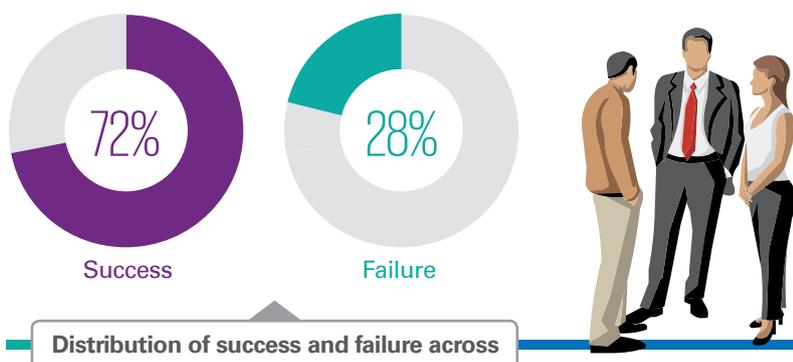
As we look at agile adoption, its approaches and the barriers to adopting agile; the recurring factor that was driving organisations towards being agile was responding to customer expectations to deliver business value, faster release cycles and quality.

Earlier, the focus was hugely on development, engineering and delivering quickly and testing took

a backseat. With coding and short release cycles, quality suffered and retaining the customer base was the elephant in the room. Agile teams now had to grapple with developer testing and their take on quality rather than an independent voice on quality. Projects would fail regularly due to lack of a long term strategy and plan, lack of requirements management across iterations, technically unaware

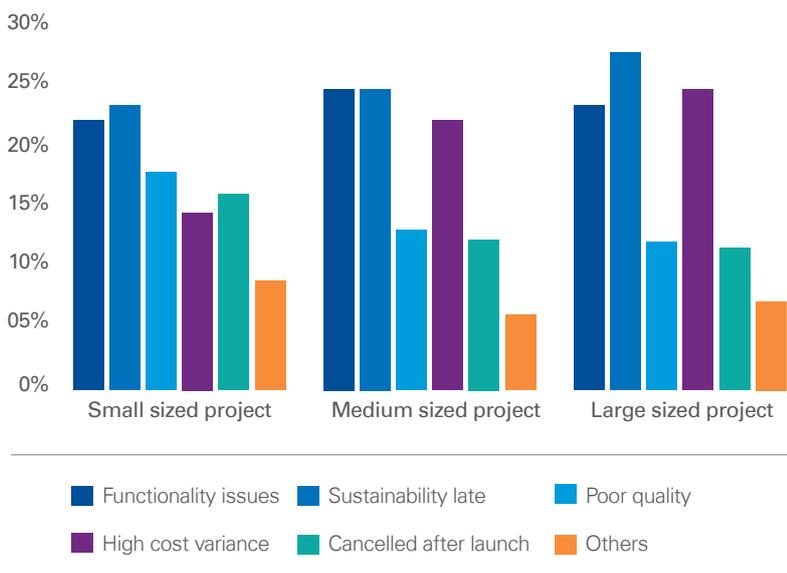
end users, lack of programme management for large scale projects and lack of testing as demonstrated in the above instances. While trying to solve one of the problems of unyielding requirements, these projects created several other problems leading to complete project failure.

1.7 Why agile projects fail



Source: Gartner-Why projects fail accessed on 15 September 2017

Why agile projects fail?



Source: Gartner-Why projects fail accessed on 15 September 2017

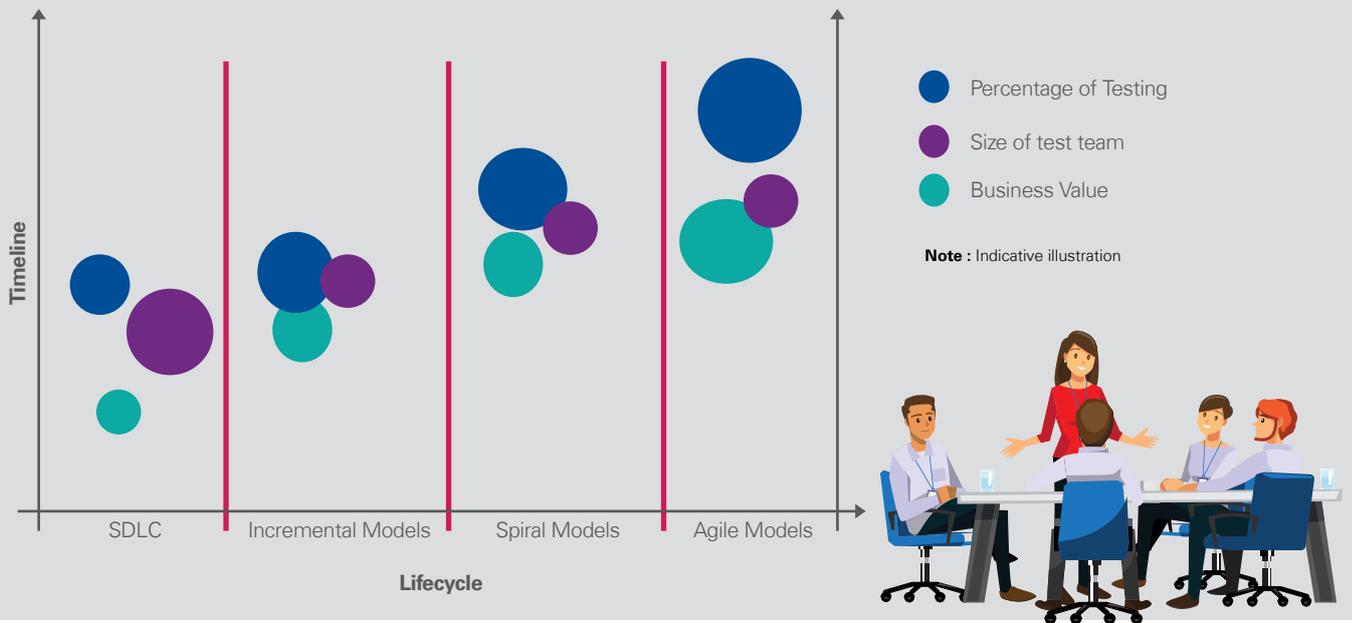
Gartner’s research explores why agile projects, causes of project failure across three project sizes. The major takeaway for test professionals is for them to analyse the balances required when delivering projects in shorter iterations with greater quality and all the agreed functionality.

The leading cause for failure of agile projects across all project sizes is the issue around functionality – especially when the teams are unable to deliver the business value promised, leading to either expensive rework or cancellation of the work orders.

Lack of quality comes a close second for the failure of agile projects across the different project sizes. As seen in the above instances of project failures, a lack of proper testing can lead to epic failures. Despite the teams trying to deliver in shorter iterations, the teams did not spend enough time planning for the testing types and phases. Due to a lack of proper testing, the quality of the product application was affected and the damage caused was irreparable.

It is not entirely unexpected to see that the challenges delivering projects on time, within budget and with the agreed functionality with crunched testing efforts are mentioned by two-thirds of the respondents as the causes of project failure, in line with our previous observations.

2.1 Agile - An antipattern to testing



Relevance of testing

In agile methodologies, the lack of well represented, detailed requirements, short iterations with quick delivery bring to light significant questions on the traditional view of testing. How can end-to-end testing be carried out when the complete application or product is not yet available? What levels of documentation would suffice?

What phases and levels of testing does one eliminate and what does one retain in the agile delivery lifecycle? How does one measure the effectiveness and efficiency of automation efforts? In all of this, the non-functional tests such as performance testing, load testing, security testing, usability testing, reliability testing, scalability, etc. get pushed till the very end. With developers donning the hats of testers as well, the role of a professional tester is also a bit of a riddle. What value can the tester add to this highly unstructured and

constantly changing environment?

Until recently, testing was seen as a factor that affected the agility of teams while the focus was on the development and engineering. The role of a tester and relevance was questioned as testing was presumed to be a hindrance in the agile environment.

Is there a possibility of consistently raising the bar on quality while reaping the advantages of agile – agile methodologies stress on the need for speed, consistent customer engagement and acceptance of change.

The emphasis has now shifted from development and engineering to testing. The evolution of testing has progressed from being tester-centric (manual testing) to effective testing that adds business value (automation). With the traditional waterfall development lifecycles, though the number of testers were more, the effectiveness of testing

and the amount of testing did not increase over time or add business value. The effort expended by the test teams did not yield the business value when compared with the effort invested.

With the advent of agile, the business value being delivered has seen a more focussed approach. With the adoption of agile, development teams have shown that automation is the way, thereby, reducing the manual effort and the number of testers or QA in the team. Though the number of testers in agile teams are fewer in number, the amount of testing increased sprint on sprint. This is made possible by agile teams adopting technology such as Robotic Process Automation (RPA), Machine Learning (ML), Artificial Intelligence (AI) and cognitive analysis.

2.2 Deconstructing testing in agile teams

There are many myths that shroud the mindsets of organisations looking at adopting agile methodologies. These myths are

not only around agile practices, but also around testing and the need for testing in agile teams.

Myths	Testing in agile is ad-hoc	Testing in agile is unstructured, undocumented and quick	Agile does not have defined strategies
Reality	Though different from planning in waterfall, testing planning in agile is an inherent part of sprint planning and test cycles are in-sync with the dev cycles	Agile manifesto states "Working software over comprehensive documentation" and suggests to only document that what is needed and has clear added value to document	Testing synchronized with the dev cycles, agile aims for quick feedback loops with continuous testing and frequent testing
Myths	Only unit tests are sufficient-Test Driven Development(TDD) is enough	Agile practices can be "tailored"	Unit tests eliminate the need for manual testing
Reality	Test Driven Development (TDD) is good, but one has verify if the code written meets the specified requirements and units are insufficient to cover this	Tailoring could be done to get around an organizational dysfunction, while ensuring that the Agile Manifesto is preserved. Quasi-agile does not reap the benefits intended	While Test Driven Development may reduce the manual testing efforts, it will not completely remove the need for manual testing
Myths	Developers have adequate testing skills, agile does not need testers	Unit tests form 100% of your design specification	
Reality	The focus for the developer and tester is different- An independent testing team serves as a check point for feedback on quality, to validate the functionality	Unlike the V-model, the requirements may not be frozen in agile , but requirements are refined though collaboration for robust specification	



2.3 Testing maturity in agile teams

Self organising	Quality goals	Automation	Governance	Measure
<ul style="list-style-type: none"> • Development and test work as partners to deliver applications • Development and test teams communicate in real-time. 	<ul style="list-style-type: none"> • Quality goals embedded within sprint goals • Balance functional and non-functional tests • Bugs are fixed immediately. 	<ul style="list-style-type: none"> • Quality pyramid-based strategy • Continuous integration and continuous deployment 	<ul style="list-style-type: none"> • Actively practice acceptance test driven development • Strict definition of done (DoD) 	<ul style="list-style-type: none"> • Test efficiency • Test Coverage • Refactoring to reduce technical debt

To better understand why releases were slower than expected for organisations even though they had embraced agile, one needs to look at the testing and QA practices that the organisations followed. While there are many approaches taken to agile development, as a general rule there are certain key best practices for testing in a fully agile process. As agile adoption gains momentum, organisations could succeed if they understand the different levels of agile maturity. Product and development managers need to evaluate where their organisation stands on the maturity scale, and what would suit them best.

An agile maturity framework is undoubtedly needed to help make the case for acceptance, adoption, process improvement and benchmarking.

As part of an agile readiness assessment, IT development organisations should first assess their as-is development landscape at the technical, programme and project and organisational levels. Process and practices should be assessed for their effectiveness and ease of adoption.

The five pillars of achieving test maturity in the agile development methodology are:

Self organising teams:

Development and QA teams need to work as partners to deliver the promised business value to the customer. With self organising teams, the communication should be real time and impediments need to be addressed immediately.

Quality goals: Quality goals need to be locked down with the sprint goals. The quality goals should not decrease despite changes to the customer requirements. Agile teams need to set the test strategy and continue the test planning as part of the sprint planning activities. Quick time-to-market can only be achieved with quick release cycles – quick release can be possible only when the defects are fixed immediately and the defect ageing is controlled and monitored.

Test automation: Test automation plays a crucial part in the agility of teams and their ability to respond quickly to changes. With the automation scripts being written in tandem with the code, the effectiveness of automation tests are higher. With the quality pyramid as reference, the automated unit tests percentage should be the highest approximately 70-75 per cent, while automated API tests, integration tests and component tests should comprise of 20 per cent, the rest of the tests should

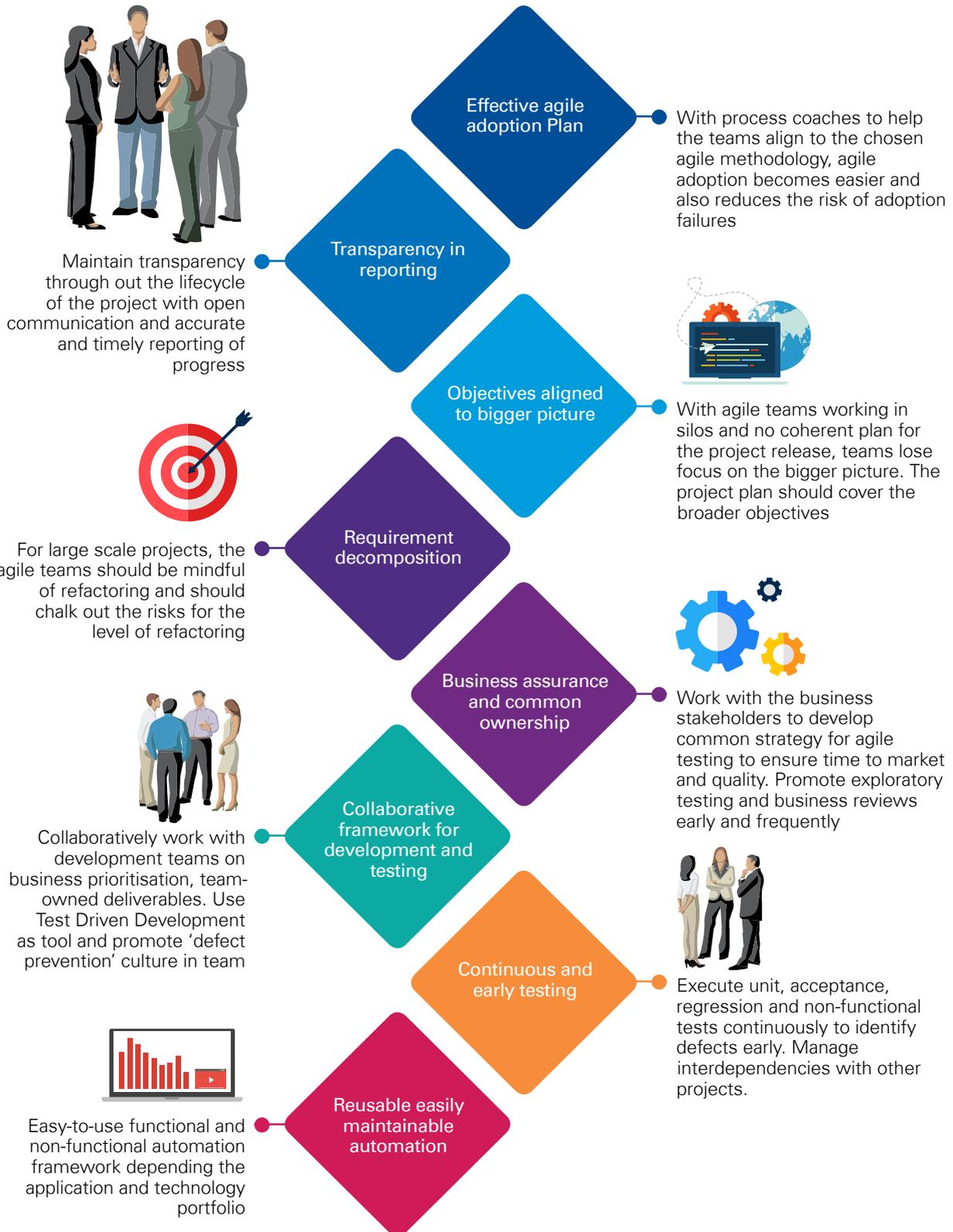
be covered by automated user interface tests. This can help teams truly achieve agility.

Governance: Self organising teams may be able to govern themselves when the team sizes are as prescribed, but as the teams scale up, governance becomes key in delivering large projects. Teams may need to strictly adhere to the definition of 'ready' and definition of 'done' as the teams scale. For large implementation, governance should be able to provide the teams with the bigger picture and the roadmap to achieve maximum business value all the while being nimble and adapting to changing customer requirements.

Measure and adapt: As the three pillars of Scrum (Inspection, Adaption and Transparency) suggest, it is important to maintain transparency, to consistently inspect and adapt to the needs of the customer. To achieve the quality levels set, agile teams have to monitor progress and quickly take corrective actions. Metrics play a crucial part in the process achieving the expected quality. Teams need to monitor the metrics such as percentage of automation achieved, automation yield, test efficiency and test coverage, which help ensure that the quality goals are not hampered.

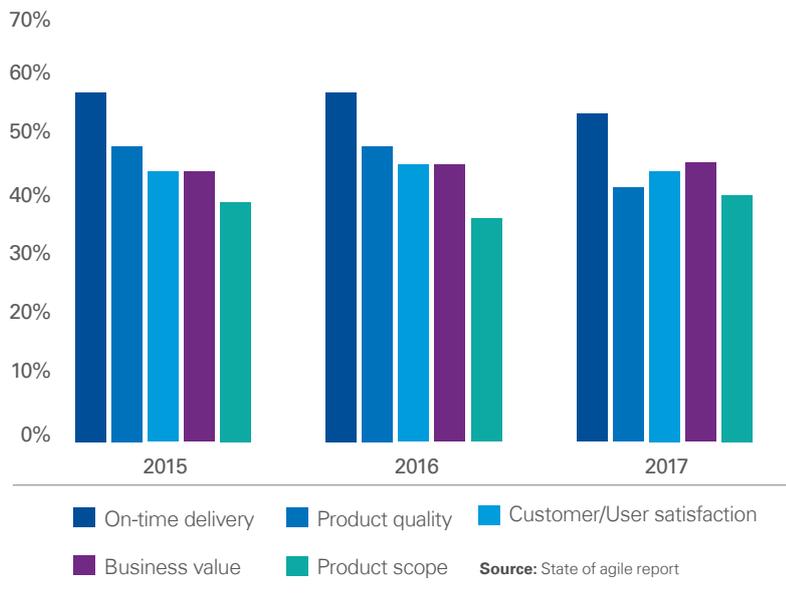


3.1 Quality in agile teams



3.2 Measuring quality success in agile

Measuring agile success



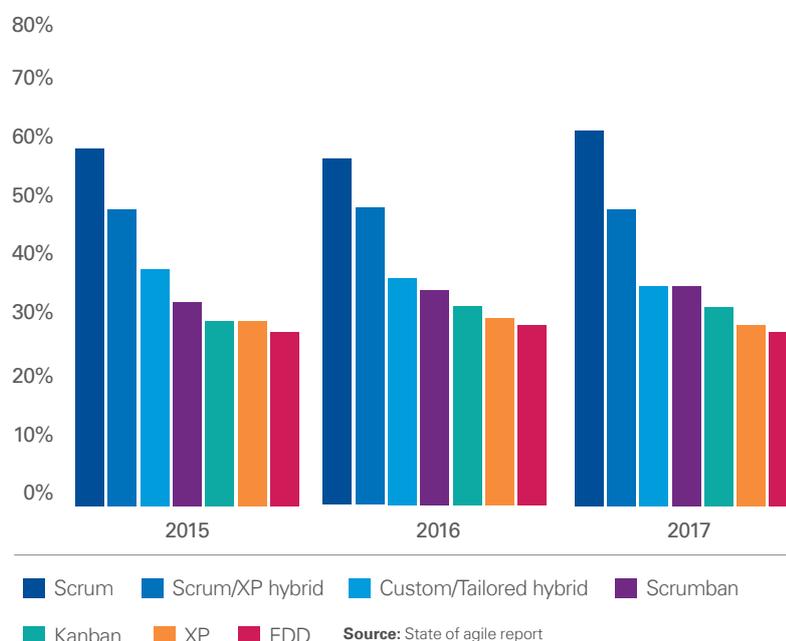
As organisations have embraced agile, it's natural for return on investment to be foremost on the agenda. But how can one measure whether the switch to agile is bringing about the desired results?

Quality metrics should be able to give agile teams a competitive edge - quicker time-to-market, enhanced customer experience, risk mitigation and more rather than just numbers in terms of cost savings.

Most organisations have a set of goals that they want to achieve through this transition – such as faster implementation, increased efficiency and mostly making customers happy by raising the quality of a project, product, or service. Business value was cited as the second most popular measure (46 per cent) of an agile initiative's success, rising from fourth in the prior year. On the contrary, it ranked as the eleventh most popular measure (23 per cent) on a day-to-day basis. This highlights an important conflict between how more strategic areas of the enterprise are measuring success and how success is being measured at the team level.

Business value as a measure of an agile initiative's success went from fourth in 2015 to second in 2016. On-time delivery of projects and customer/user satisfaction remained in the top three measures of agile initiative's success as they have in the past few years.

Agile metrics for success



09. State of agile survey, 2013,2014,2015,2016, 2017, Version One accessed on 14 September 2017

While business value was cited as the second most popular measure (46 per cent) of an agile initiative's success, it ranked as the eleventh most popular measure (23 per cent) of an agile project's success. Velocity (67 per cent) continues to be the

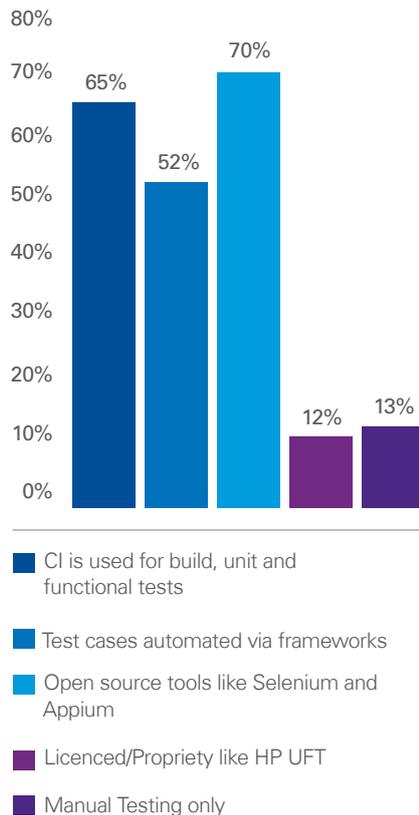
number one measure of an agile project's success.

While the success of the transition cannot be measured with a single parameter, organisations will get clarity on whether they are

delivering value and what factors need improvement. As with the pillars of Scrum, teams can get a clear indication what areas need to be further inspected, and adapt quickly by introducing, removing or amending practices.

3.3 Automation tooling

Leading automation tools in agile



Source: State of agile survey, 2013,2014,2015,2016, 2017, Version One accessed on 14 September 2017

Test automation in agile

Over time, agile test tools have become increasingly important to the performance of agile teams. The use of the right tools not only helps agile teams improve their performance but also increase their efficiency.

Automation is the need of the hour and is used to eliminate the mechanical, repetitive tasks and bring in quicker feedback loops and faster release cycles - essential in agile. With the fast pace of delivery in agile, test data management and test environment management needs to be efficient with no room for time consuming manual efforts. Organisations across all industries are expected to achieve more with less - less time, less resources and less budget. However, many organisations still continue to do manual testing.

Activities that can be automated across the development lifecycle spans code reviews, build and release management, unit testing and functional testing.

- **Build and release:** Nightly builds are the norm in agile teams and there is a need for zero manual intervention in this process. This requires good configuration management and build tools.
- **Unit testing:** These are part of the nightly build, thereby creating a quick feedback loop for the development team. The execution of these unit tests require manual interference.

- **Static analysis tools:** Static analysis tools for code review against coding standards are used to find defects and are a more feasible option than doing manual code review.
- **Functional testing:** Earlier, functional automation testing was restricted to regression testing which is aimed at reducing the cycle time for regression. With agile teams, testing starts early on in the development lifecycle with the test design being the priority while test execution should be automated.

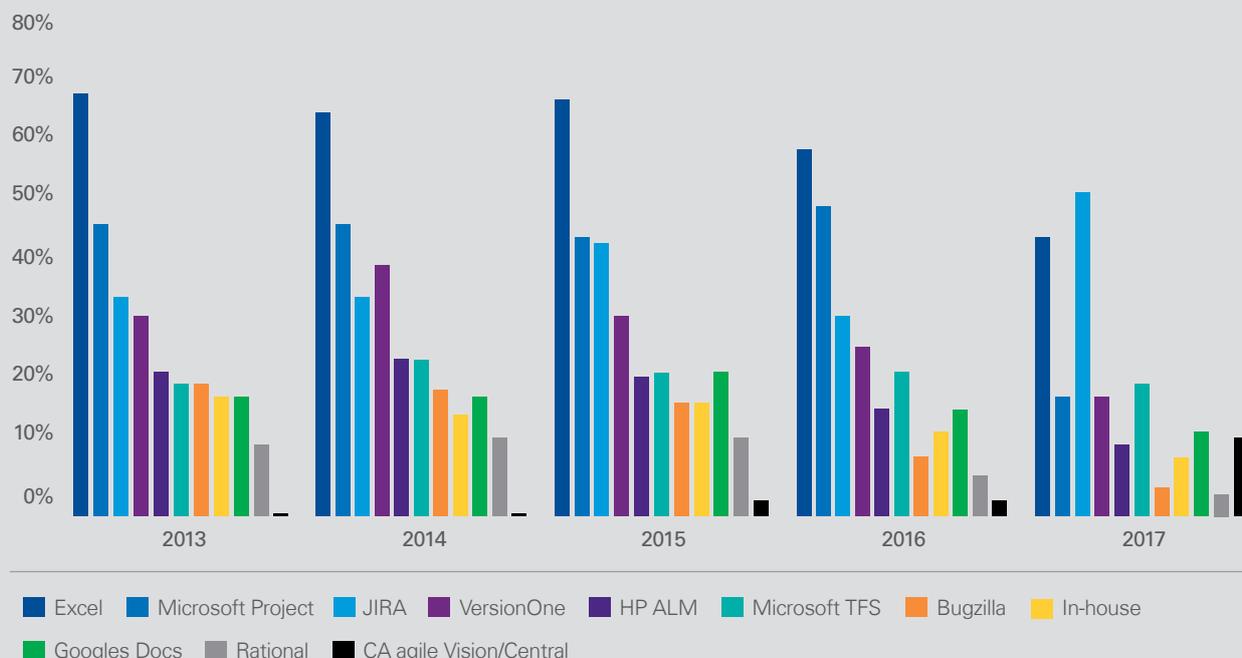
With the adoption of test automation, organisations could reap the following benefits:

- 1. Reduced time-to-market** – one of the biggest reasons for organisations to adopt agile methodologies
- 2. Improved visibility** – Test automation along with analytics allows stakeholders to fully understand the progress and status of testing
- 3. Identifying defects earlier** – to ensure that defects are identified and fixed early on, smart automation is effective in identifying patterns
- 4. More testing** – with test automation, organisations can test more in less time. More test cycles can be covered in less time.



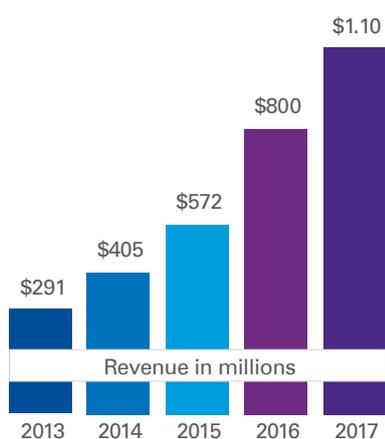
3.4 Agile lifecycle management tools

Use of agile management tools



Source: State of agile survey, 2013,2014,2015,2016, 2017, Version One accessed on 14 September 2017

Use of agile management tools



Source: IDC Worldwide agile Application Life-Cycle Management Software Market Shares, May 2014 accessed on 14 September 2017

Agile management tool landscape

With fierce global competitive pressures and ongoing economic volatility, organisations are reinvesting in IT projects. There is a steady increase in agile Application Lifecycle Management (ALM) investment. Squick analysis into the worldwide agile ALM market. The survey shows strong growth patterns for agile ALM software for the 2014–2019 time frame, with growth to around USD1.8 billion by 2019 and a high CAGR of 32 per cent (on small initial numbers)¹⁰. The complexity of deployment/DevOps environs with mobile, social, cloud, and big data and analytics is a business inevitability. Complex sourcing needs of IT projects also stress on the need for agile ALM automation for successful agile adoption. Commonly used agile

ALM tools are Microsoft Project, Atlassian JIRA and VersionOne. Agile ALM tools are now not just 'nice to have' capabilities but are core to business execution.

Agile ALM is driven by the need for consistent IT- business alignment, corporate governance, compliance and regulatory requirements. Agile vendors have been evolving their offerings across the three core functional markets that feed this competitive market — software change, configuration, and process management (SCCPM); project and portfolio management (PPM); and automated software quality (ASQ) with agile testing.

10. IDC Worldwide agile application life-cycle management software market shares, May 2014 accessed on 14 September 2017

4.1 Key takeaways

Iteration plan		Agile development Agile testing			Retrospective
Activities	Strict Definition of Done (DoD)	Test story preparation	Facilitate developer for early automation	Automate test scripts	Plan for non-functional testing
	Story task break-down	Consolidate and log feedback		Feature acceptance test cases and early automation	Monitor test progress and status
Testing within iteration					
Test Types	<ul style="list-style-type: none"> • Unit testing and Integration testing • Functional and E2E testing • Regression testing for previous iterations • Feature acceptance testing 			<ul style="list-style-type: none"> • Performance testing/Security esting • Regulatory/Compliance testing • Final UAT & Final Regression testing • Exploratory testing 	
	<p>Agile testing activities are asynchronous in flow yet harmonious with development activities and business expectations.</p>				

The key takeaway here would be:

<p>All testing activities within the iteration/sprint should be well chalked out and planned for. The iteration/sprint plans should include the testing types and test cycles as well</p> <p style="text-align: right;">01</p>	<p>Teams should have a robust definition of ready and definition of 'done' for all activities and tasks across the iteration</p> <p style="text-align: right;">02</p>	<p>All test/QA activities should be well planned and chalked out. Changing customer need should be tracked in the lifecycle management tool for transparency and efficiency</p> <p style="text-align: right;">03</p>	<p>User stories should be broken down to tasks as small as possible. All tasks that do not have enough clarity should be pushed back for more refinement until the team is clear on the expectations</p> <p style="text-align: right;">04</p>
<p>Early automation should be facilitated in terms of unit tests early on. Automation as early as possible should be the way forward. Teams can adopt a test driven or acceptance test driven approach to achieve maximum quality and deliver business value to the customer consistent</p> <p style="text-align: right;">05</p>	<p>Working with cross-functional teams helps ensure room for refinement and quick adaptation. All Scrum/iteration activities should be carried out to reap all benefits.</p> <p style="text-align: right;">06</p>	<p>Non-functional testing including performance testing, security testing and cross browser compatibility tests should not be pushed to the far end of the development lifecycle, but should be considered and planned for early on.</p> <p style="text-align: right;">07</p>	<p>Test progress and status should be tracked consistently with agile ALM tools for transparency and to enable teams to take quick corrective actions.</p> <p style="text-align: right;">08</p>

A test-driven approach has repeatedly helped achieve quality across the lifecycle. Automated unit tests help weed out defects earlier in the lifecycle, leading to quick feedback loops.

Automated unit tests precisely determine the behaviour of the system, With test-first teams, the percentage of defects found later in the lifecycle is much less . Good

test-first teams find that they get significantly less defects throughout the system life cycle and debugging time is reduced. With the Agile Manifesto emphasising the need for

documenting only what is needed, unit tests can also be used as design documentation that is in-sync with the production code.

5.1 Summary



Enable agile readiness

- Agile coach and Process coach to mentor teams
- Structured approach to enable agile readiness from people, process & tools perspective
- Standardized agile training to all team members

Shift left

- Quality Pyramid-based Strategy
- Ensure test strategy and quality goals are set in line with business goals

Test driven approach

- Adoption of techniques such as Risk Based Testing\OA based test design for enhanced test coverage and optimized test cases
- Ensure acceptance criteria and Definition of Done is aggressive followed

Enhanced test automation

- End to end lifecycle focused automation v/s focus on regression Automation
- Functional Test Automation – Build ahead automation
- Test design automation – Model Based Testing
- Test Data Management

Project management

- Agile Estimation models for predictable demand and capacity management
- Platform/Application aligned Regression teams
- Focused reporting on status and progress to maintain transparency through all phases of the project



Adopting agile has numerous benefits as compared to the traditional development methods such as waterfall or the V-model. Adopting agile brings about a visible change in the productivity, efficiency and increases team happiness. With time-boxed meetings, less documentation, agile reduces the time wasted in redundant and repetitive tasks and helps the team deliver high-value features. This helps ensure faster time-to-market, increased predictability and reduced risks. With self organising teams and transparency within teams, agile helps broaden organisational experience. With less time spent on micromanaging multiple projects, agile helps the organisation prioritise their strategic goals.

However, we do not believe that companies should adopt agile methods in all places. In many functional areas, such as plant maintenance, purchasing, sales calls, or accounting, more traditional structures and processes are likely to deliver lower cost, more repeatable outcomes and more scalable organisations.

Organisations looking at transforming their development methods first need to evaluate and assess their current landscape and the challenges they face in delivering business value. They also need to be mindful of the organisation's culture and its acceptance of change. Consistent and meaningful engagement with teams through agile workshops, trainings and coaches would help organisations build the confidence and trust in the journey of successful agile adoption. Readiness in terms of people, process/practices and tools could help achieve success.

As seen before, agile adoption is driven by the need for quality delivery. Organisations must be cognisant of the test approach and test strategy that they implement. With the adoption of models such as risk-based testing, OA for optimisation, agile teams can achieve greater test coverage. Acceptance test-driven development can be key to achieving greater business value and enhanced customer satisfaction. End-to-end automation in the delivery lifecycle could help agile team's deliver better quality in short timelines while

helping ensure higher productivity and efficiency. Though agilists may not completely agree with the need for project management in agile teams, there is a pressing need for teams to be able to see the bigger picture. While the agile development teams focus on delivering the best possible solution to the customer, project management in terms of capacity management, status and progress reporting to maintain transparency are much needed.

Today, for testers to be able to add value to the agile teams, they not only have to be technically sound but their focus needs to shift towards business value and meeting customer expectations. Only finding flaws may not suffice but the tester needs to understand if the output being delivered is complete and acceptable to the customer. For the testing to be truly agile in today's world, we need to understand the business context while helping ensure that testing is always in the driver's seat.

Acknowledgements

This publication has been prepared by a team comprising:

- Preethi Kudur
- Nisha Fernandes
- Viswanath Ramachandran
- Darshini Shah

KPMG in India contacts:

Mritunjay Kapur

Partner and National Head

Strategy and Markets

Leader - Technology, Media and Telecom

T: +91 124 307 4797

E: mritunjay@kpmg.com

Akhilesh Tuteja

Partner and Head

IT Advisory

T: +91 124 307 4800

E: atuteja@kpmg.com

Prasanth Shanthakumaran

Partner

Business Excellence Tech

T: +91 80 3980 6000

E: prasanths@kpmg.com

Rajesh Kannan

Director

Business Excellence Tech

T: +91 80 3065 4677

E: rajeshk@kpmg.com

[KPMG.com/in](https://www.kpmg.com/in)

Follow us on:

[kpmg.com/in/socialmedia](https://www.kpmg.com/in/socialmedia)



The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

© 2017 KPMG, an Indian Registered Partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The KPMG name and logo are registered trademarks or trademarks of KPMG International.

This document is meant for e-communication only.