LTIMindtree

POV

Empowering Automation Delivery Team with Hybrid Agile Principles

Author Asif Mapara

Automation has been in existence for decades and has also evolved in terms of

A. The tools/technologies/frameworks adopted to enable automation

B. From script-based automation to workflow-based to AI/ML-based

C. Life cycle of the automation development process

D. The appetite of the businesses in investing in automation

and various other ways. But despite all of this, there are challenges in implementing automation solutions successfully and sustaining the momentum of the same.

When it comes to automation, one of the critical challenges is making the teams developing automation solutions understand the need to follow the right kind of development processes, standards, and guidelines. Because, from an IT services vendor perspective or even otherwise, while the development teams take the required precautions to follow all the development and quality processes to be adhered to in developing their mainstream projects (core development/maintenance/support and client approved SOWs), the same is usually not practiced for automation solution development (value-added type of initiatives/projects by the vendor).

And while the mainstream projects follow different types of development methodologies like Waterfall, Agile, etc. and get audited by the central quality team, proactive and value-added automation projects are not generally following even the conventional waterfall method for development properly and are not subject to any quality audits to ensure the quality of delivery. The reasons for these problems are attributed to the vendor and that, at times, the clients don't provide the right kind of focus and support to the automation initiatives.

Whether it is a mainstream project or proactive /value-added type of automation project, following the proper development methodology, standards, and guidelines is paramount for successfully deploying automation and sustenance thereafter. Like, having a centralized automation CoE to enforce these development standards and processes across the vendor organization and also securing the right kind of support and focus from the client will help.

Some of the automation life cycle processes like opportunity identification and qualification, solution design, availability of the standard and custom tools/frameworks to carry out automation initiatives, and the readiness to embrace automation have matured over time , yet, when it comes to developing the automation solution, teams adopt a very ad hoc approach to development. Whether it is just developing one or two automation use cases/opportunities ad hoc or an automation team/factory that is set up to handle a huge backlog of automation use cases, it is essential to follow the Agile way of developing the automation solutions as the stakeholders can't wait to see the results after months. Since the entire industry is marching towards an Agile way of working, it is essential to adopt the Agile principles in developing automation solutions to avoid any issues with automation initiatives regarding the speed of delivery, governance, quality, scalability, and maintainability.

But how does one bring about this change? There are different approaches and Agile methods, so which ones are the best fit for automation initiative realization. Rather than following only one Agile framework, the automation team should try to adopt the best practices offered by multiple Agile frameworks and make the best use of them.

As an IT services vendor, we have adopted some of the following hybrid Agile principles in our automation delivery, which are derived from Agile frameworks like Scrum, SAFe®, Nexus, XP, etc. These hybrid principles helped us in delivering effective automation to our stakeholders.

01

Persona centric Backlog management and prioritization

Rather than defining the automation opportunities for development based on the process or technology/type of automation by which it can be carried out, it is recommended to determine the automation opportunities around the activities performed by various personas.

Any automation solution will ideally impact different functions if built around processes, technology, or automation type. Hence managing the design, development, and release of such use cases would be challenging. Rapid feedback and release will be unrealistic if automation use cases are designed in this fashion.

By defining the automation use cases around the personas, we can better empathize and understand the end user's perspective regarding the activities to be automated. Further, it also helps to restrict the impact of automation to the activities under consideration for a persona rather than the entire process.

We should prioritize the automation pipeline based on what can make things faster, automatable, and cheaper for the personas performing those activities, so that the automation we deliver can yield maximum value to the respective personas.

02

Planning council to validate and refine prioritized opportunities

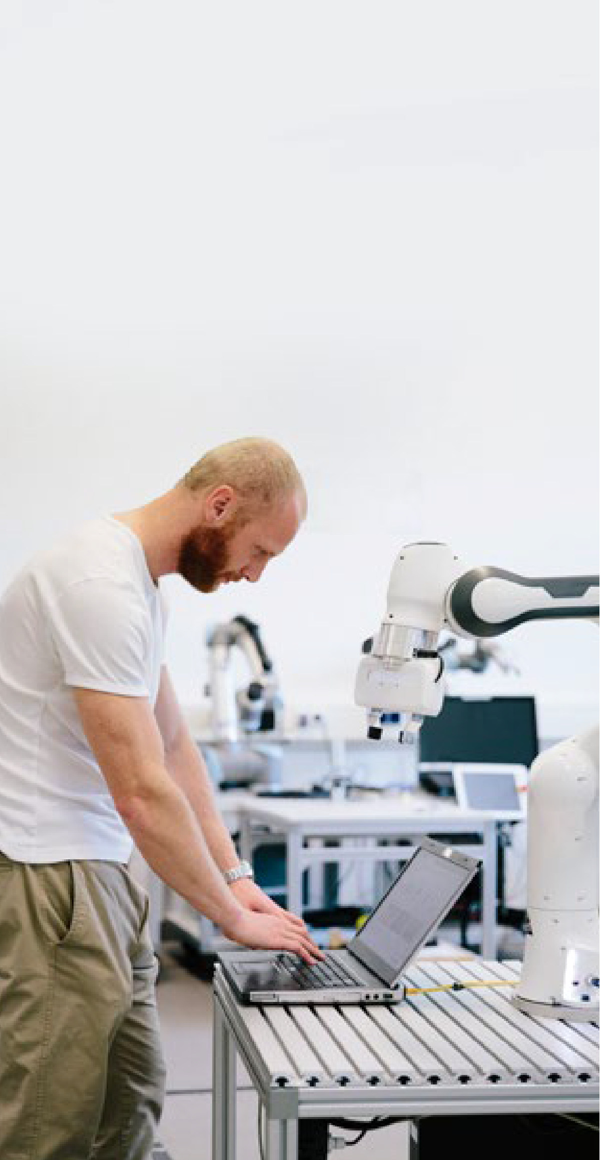
Automation assessment of a project will bring more automation opportunities to the fore. It is important to prioritize them based on many parameters that can add value to the end-user. To validate and refine the prioritized use cases, it is recommended to form a planning council (leads from the automation development team) with the following responsibilities.

- ☑ Validating whether the identified use cases are persona centric.
- ☑ Refining the requirements of the use cases with the project Subject Matter Experts (SMEs) to make it sprint ready. Also, to define the acceptance criteria with the SMEs.
- ☑ Clearly demarcate the scope of automation.
- ☑ Ensure the persona-centric use cases are split into user stories and, in turn, split into behavior-driven test cases.
- ☑ Identify dependencies and ensure they are cleared/ made available for the team before the sprint begins.
- ☑ Connect with the application/business owners to discuss the automation approach & prerequisites, get their feedback early, and set the right expectations, which will prevent the risk of any expectation mismatch at later stages.

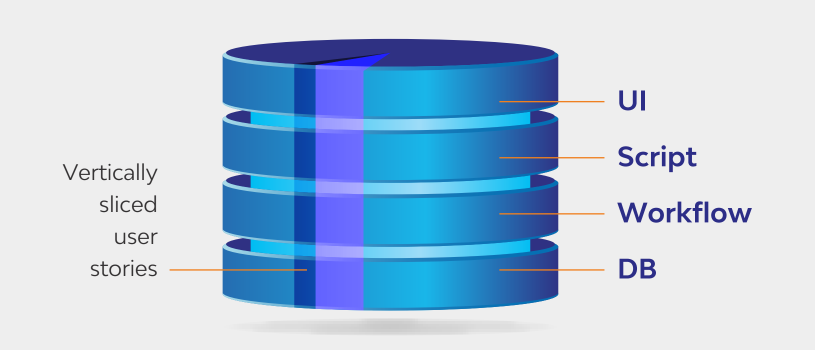
The planning council enables the automation delivery team to seamlessly run the sprint with on-time delivery and good quality.

03

Faster feedback loop through Vertical Slicing of automation use cases



Establishing a faster feedback loop is an essential component of Agile delivery. To receive faster feedback, we need to first define the activities to be automated which are currently done manually by the personas. These activities can then be mapped as user stories, and the user stories are mapped into use cases. When creating user stories, we need to ensure that they are vertically sliced rather than horizontally.



Rather than writing code for each module like UI, DB, etc., for the entire use case and then integrating them to test end to end, it is recommended to implement slices of these modules to satisfy the acceptance criteria of a user story.

By vertically slicing a use case into multiple user stories, we can achieve independent development and deployment of these user stories and receive much faster/frequent feedback. Such a feedback loop will help avoid the risks such as late detection of issues, rework, and expectation mismatch at an early stage.

The feedback can be taken at below five stages of development and release cycle to ensure built-in quality.

A. Design feedback

It is recommended that design and delivery are incremental. Too much time spent on entire design of the solution can result into waste of effort. Start delivering as soon as considerable clarity in design is achieved and not wait for full and final design signoff. It is better to seek feedback on the incremental design and proceed than to spend time discussing the entire design scope.

B. Development/Implementation feedback

It is recommended that each test case be made to pass independently. The incremental code should be reviewed by technical SMEs and tested by quality engineers regularly. This ensures that the code delivered is clean, integrated, regression tested, and the build is clean so that it will not create any impact on the automation implemented already. This approach ensures intermediate feedback during development to streamline the build process.

C. Functional feedback

As soon as each user story is implemented with all its respective test cases/scenarios passed, it is a good idea to give a demo to the SMEs to take their feedback and to ensure that the automation being done is in compliant with the activities/requirements and the standard processes being currently performed manually.

D. Acceptance

Even though we take feedback at every level and incorporate it well before the final delivery is made, it is essential to take input on the final deliverable from the business and SMEs before releasing it to production. It is recommended to validate whether the use case/user story level acceptance criteria are met before moving it to production.

E. Release feedbacks

It is essential to monitor the benefits of automation in production and seek feedback from the end-users on whether they face any issues, any changes required with the implementation, the utilization and adoption level of the automation delivered, etc.

04

Planning automation release based on Minimum Viable Product (MVP) concepts

Automation is an ongoing journey. Once we have automation opportunities in hand, the next challenge is to plan how big the releases can be and the timelines to deliver. In general, it will be beneficial to have smallest possible releases to add value to end-users without waiting for the entire automation to be completed.

Hence, it makes sense to plan the sprint/iteration in a manner that we deliver the most valuable automation on priority. This will also help get helpful and early feedback for improvements. Once the automation opportunities are identified, it is essential to identify and prioritize those opportunities that can reap maximum benefits and add more value to the personas.

05

High degree of quality through BDD and continuous integration

It is recommended that the user stories defined for an automation use case be further vertically sliced to behavior-driven test cases. The respective product/business/project owners then review and vet these test cases. While implementing a user story, rather than developing the code for the entire user story, we can develop the code at the test cases level and make the test cases pass independently. This vertical slicing of user stories can also help develop and pass the test cases parallelly and improve productivity.

We should ensure that we cover all possible scenarios (test cases) performed by a persona and implement them to make it pass. This will result in more accurate results and effective automation.

Further, as the test passes, continuous integration at user story and use case level are carried out to ensure that the use case development progresses well and meets the stakeholders' requirements without any last-minute surprises.

06

Decoupled release management

The automation development team should keep developing, testing, and deploying the automation uses cases in a staging environment and leave the decision of when to release the use cases to the production to the business because the business will take some time to decide when to release the use cases to production based on certain business conditions.

Whenever a business team decides on a production release, the automation team can come back and help with the production release. This way, the deployment to staging and production release can be decoupled effectively. The automation team can focus on developing and deploying more use cases for staging without incurring any wait time in completing the production release.

07

End of iteration/ sprint demo and review

In addition to giving frequent demos (show and tell) to the SME/PM to get their feedback and appraise them of the progress during the sprint duration, it is essential to have end of sprint demo organized by the automation development team to the project SME/PM/delivery manager/client stakeholders of the project wherein the automation is implemented, for the following reasons:

A. End of sprint demo act as a knowledge sharing session for the automation development team.

B. To get feedback from the project team, delivery manager, and client stakeholders to get final acceptance and improve sprint performance.

C. To get testimonials from the client stakeholders on the automation done.

Conclusion

Automation development processes need to be given importance to deliver the correct value to the end-users. Following an Agile model of development is essential to the success of the automation sprint cycle. In addition to the standard Agile practices like daily stand up call, sprint planning using poker games, defining user stories, show and tell to get user feedback, retrospection meetings, etc., it is essential to bring in more Agile principles into practice from various Agile frameworks to reap more benefits like, to improve the productivity of the automation team, to improve the quality and timeliness of delivery and add more value to the end-users/clients.

The persona centric use case definition, planning council for refining the use cases for sprint readiness, vertical slicing of use cases into user stories and test cases, ensuring frequent feedback loops, decoupling the use case deployment in staging from production release, end of sprint demo, etc. are some of the principles we have experimented and found to be very effective and delivering the benefits intended. Following the right processes to deliver quality is the key to success in any development cycle and applies to the automation delivery.

Author Profile

Asif Mapara

Associate Principal
Consulting, NWOW Automation (NAUT)

Asif is an established Agile coach and a certified SAFe® 5 Agilist (SA). He leads the scrum ceremonies of CAO (Custom Application Operation) automation delivery sprints of multiple delivery units within LTIMindtree. He has over 18 years of experience in IT service delivery, project/program management, consulting, Agile coaching, and transformation. He has successfully coached and mentored the CAO delivery teams to adopt an Agile way of developing and deploying automation use cases.

Asif was instrumental in leveraging some of the best practices from multiple Agile frameworks like Scrum, SAFe®, Nexus, XP, etc., and establishing a hybrid delivery model to enable rapid automation delivery by CAO. He has played the role of Agile coach for some global clients. He is passionate about coaching/mentoring the teams to adapt design thinking concepts, understanding personas through empathy maps, defining OKR based strategic themes, and portfolio Kanban.