Best practices for testing AI solutions

Mr. Sojan George
Business Development Manager, AI Lab, TCS, Kochi
sojan.george.316@gmail.com

Mr. Rajeev Mullakkara Azhuvath
Enterprise Architect, AI Lab, TCS, Kochi
ma_rajeev@yahoo.com

AI is slowly increasing its presence in all aspects of one's daily life and the beauty of it is that not many are even aware of its presence. For those in the software field, it is likely that everyone will encounter an AI application sooner than later. While AI weaves its "magic" and spell bounds us, one key question that stumps software engineers, especially the quality assurance engineers is, *"How do we ensure "magic" is working as expected?"*

To answer this tricky question we need to comprehend the underlying assumptions and dependencies in artificial intelligence solutions. Moreover, we need to understand why traditional testing methods will not work in most AI projects. This article aims to give a point of view (PoV) on these points along with some strategies and testing best practices that one can adopt while building AI solutions.

The Rise of AI – Changing the status quo in the testing domain

For decades, in the pre-AI era, software professionals build system that they could control in almost every aspect. Data was limited, engineers knew what to expect in all scenarios and solutions where "hand-engineered" to behave as expected. The systems were built with great precision, covering most scenarios. Quality assurance testers knew exactly what needed to be tested and more importantly knew what the expected output is. However, all that changed in the AI era and many assumptions in the pre-AI era suddenly became obsolete.

- **Solutions need not be 100% accurate to move into production:** One of the biggest shift in the AI era is the acceptance of a solution that is not 100% accurate all of the time. For quality assurance professionals, it might be difficult to comprehend this detail. The reason why a 80% or even a 70% accurate solution might be out into production is that the benefits, be it an increase in efficiency or reducing operational cost or even improving customer experience, more than compensates the effect of having an imperfect solution in production. Moreover, AI solutions are expected to improve over time via "Self learning" and feedback mechanisms.
- **Testing everything was difficult before but impossible in the AI era:** One of the key differentiators of the under-lying principles of building AI solutions are the dependencies on data and the critical role it plays on the overall solution. In this data intensive era, this critical component of the solution continues to be unpredictable in production. Just take the example of a chat-bot or a document classifier. The permutation of possibilities increases exponentially as we move from words to sentences, from sentences to paragraphs and from paragraphs to documents (as depicted in Figure 1 below). As such, there is no practical way to test every permutation and combination.

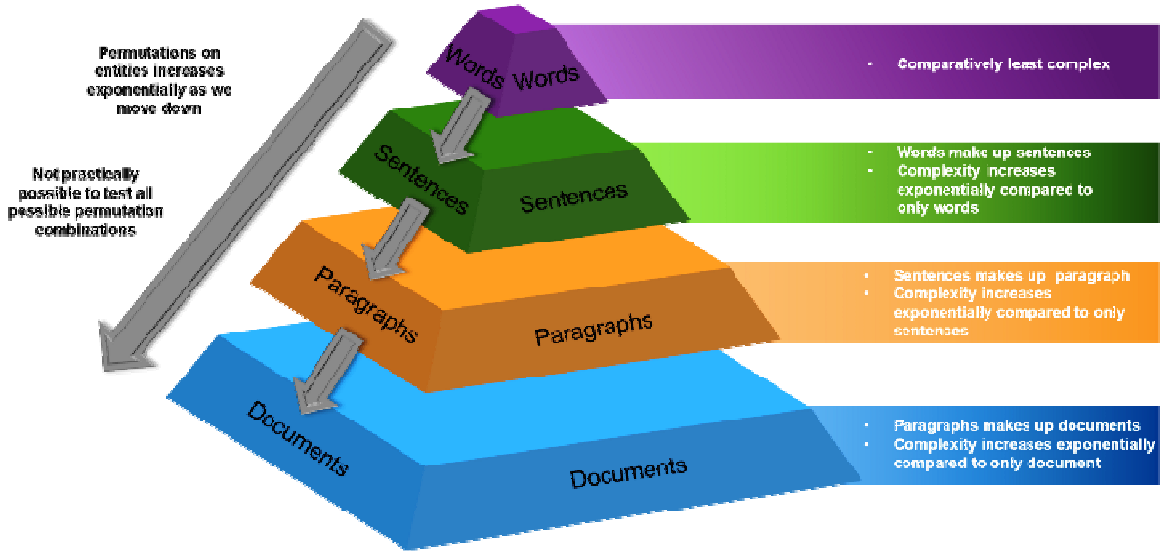

Figure 1: Complexity Increases Exponentially

- **Deployed solution evolves in production:** One assumption that quality assurance engineers used to make was that the solution that they sign off would remain constant until the next release. However, most AI solutions have a feedback loop that constantly evolves over time based on incoming data feeds. The same input need not give the same output always. We cannot do away with this "self-learning" feature for the sake of stability in testing, because in most cases, it is what makes AI "magical".
- **The technology and data also drives the solution:** Previously, business was the main component that drove solutions. As such, quality engineers needed to know the business to ensure that the solution being built served the end goal. However, today, apart from business, technology and data are also key drivers. As such, QA engineers of today, not only need to understand the business but also the technology and data behind it. This is a shift from the earlier expectations of QA engineers.
- **End users can influence the way the solution performs via Information Poisoning:** End users can influence the way an AI solution performs in the long run, especially in un-monitored self-learning systems. For example, in order to popularize any item in an e-commerce site, one can use bots to retweet, like or share to increase the items ranking unethically. Similarly, biases in sample training data could cripple a solution in the cradle. Biases are often amplified in feedback loops, leading to biased decisions. Human need to continue playing a critical role here. Today, most systems continue to have humans in the loop to ensure that the AI solutions are progressing in the right direction. These concepts would be new to QA engineers who have only worked on traditional applications.
- **Explain-ability and reproducibility of bugs not so straightforward:** In traditional applications, non-reproducible bugs are rare. One of the expectations from the QA engineer when logging a bug will be 'steps to reproduce the defect' and the development team is accountable to explain the reasons for the deviation. However, this is not so easy in AI solutions. The end solution is often a "black box" that comes out with the most probable answer. Expecting an audit trail is not so easy in AI based solutions leveraging technologies like deep learning.

Best practices for testing AI applications

Technology has moved on. Business processes has moved on. Methodologies have moved on. It is time for testing practices to move on as well. Over the last few years, AI solutions have challenged the status quo of existing QA processes. While many companies have come out with their own AI testing process, there seems to be no global AI testing methodology accepted across enterprises. As such, rather than come up with another AI testing process, we have highlighted some best practices and points to take care of while testing AI applications.

- **Change in the testing mind-set, from Determinist to Stochastic:** This is perhaps the biggest change that any tester needs to undergo to be good at testing AI applications. Traditional applications in the pre-AI era promoted a deterministic mind-set that expected QA engineers to know what to expect for every input. Every test case is either a 'Pass' or a 'Fail'. Every 'Fail' has the potential to delay moving into production environment. However, in the AI era, the results are based on probability and statistics. A 'Fail' in the testing environment can very well be a 'Pass' in the production environment in future, as more data is available. Any deviation from "as expected results" is not necessarily a failure, but rather a path for the system to improve and evolve.
- **Understanding how critical data is in AI solutions and building test cases to test them:** Unlike in normal non-AI applications, data plays a very critical role in AI applications. They can make or break your solution. However, chances are that testing teams will not have access to actual data and would need to prepare data as closely as possible to actual production data. Here business knowledge and changing trends becomes even more important. Test data preparation would require QA engineers to have an in-depth business knowledge of the use case and understand the mind-set of the end user. Moreover, it is important for them to evolve the testing data sets as well to ensure that they are testing based on the changing trends. Apart from this, teams need to monitor continuously to see if the dataset used to train the model is biased. If a model is trained on a data that is already biased, the AI solution will also be biased. For example, an AI solution trained to predict the acceptance or rejection of a candidate based on historical data could most probably be biased in favour of a particular gender. As quality assurance experts, it is imperative to understand the business and possible implications to ensure these bias scenarios too are tested as part of the QA process.
- **Adopting continuous testing & monitoring:** Testing never ends in an AI project. Unlike normal projects, the system evolves over time based on feedback. As such, it remains critical to have a testing process that compliments a typical AI project. As the system evolves continuously, we need to ensure that the AI model is evolving as dynamically as the external environment. Based on the use case, testing needs to adopt to ensure the various metrics like precision, recall, and f1 score are met to avoid adverse effects to the business brand, performance and compliance. As business gets new data, it becomes imperative that the model be re-trained to adapt to the new data trend.

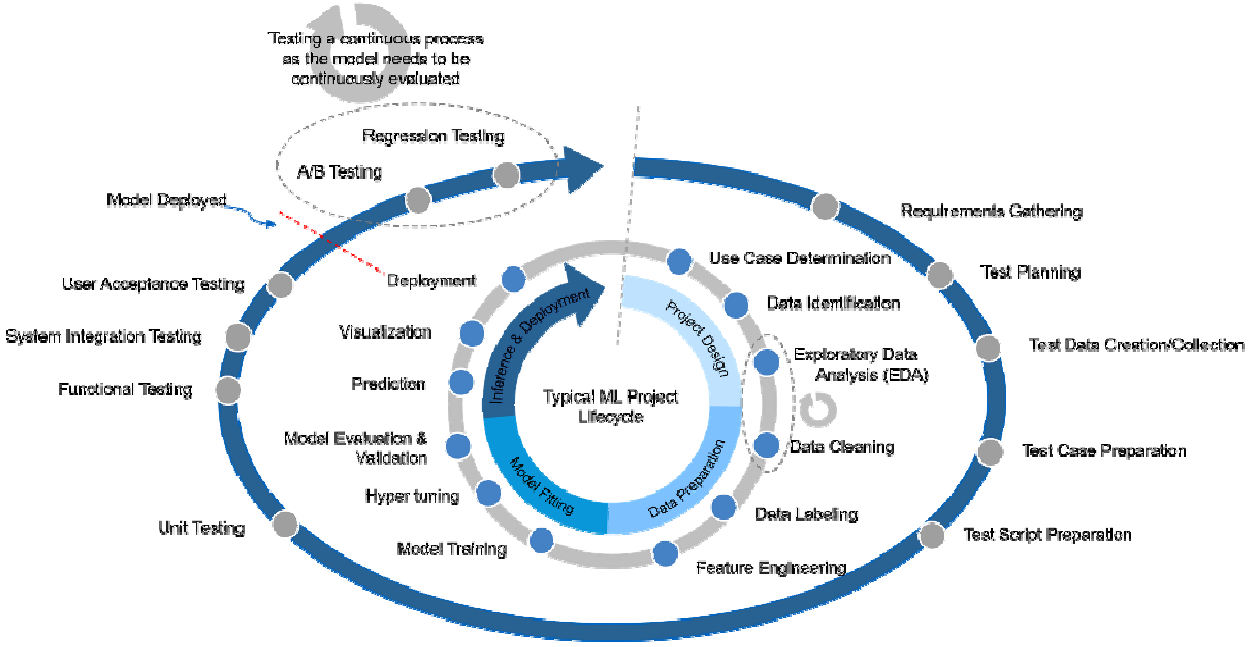

Figure 2: Continuous Testing Post Model Deployment

As highlighted in the above figure 2, even after deployment, the AI lifecycle does not end in most cases. This is especially true for solutions that depend heavily on data. Just as an AI solution evolves over time, it is important to have a continuous testing phase to ensure the model evolution is in the right direction. Post deployment, A/B testing and regression testing plays a critical role in the continuous testing phase.

- **Document the exit criteria precisely early on in the SDLC:** In AI solutions, as highlighted earlier, a 80% success criteria would be sufficient to move from one stage to another. However, how can one measure this 80%? This criteria needs to be grilled down very early on in the SDLC. If the output consist of many components like in, for example, information extraction solutions, we need to understand what the acceptance criteria is for each of the output component. Should some output component be 100% accurate always or are there components that even a 50% accuracy is acceptable? This acceptance threshold needs to be defined early on and the expectation be set (and documented) with all stakeholders in the design phase itself.
- **Pilot first within organization or beta crowd before going to complete public:** As AI solutions are so heavily dependent on data; solutions built for different organization will be different from one another (based on the underlying data). Therefore, one's earlier experience would not be an appropriate indication of future success. Couple that with the fact that the solution will not be having 100% accuracy from day 1 and we can understand the uncertainty one needs to deal with. Today social media exposes technology failure and these uncertainties could lead to bad marketing and branding exercise. Hence, planning to first expose the AI service to a controlled group provides the QA team with an opportunity to test the solution with actual data and identify fragile points, if any. As such, if one's team is closely linked with the customers in defining the roadmap, one should certainly encourage this practice across.
- **Educating customer and managing customer expectations:** AI is undoubtedly in a hype phase. Perhaps the most important step that any team building AI solutions should ensure is to maintain a realistic expectation with the customer. Committing to 100% accuracy in requirement phase (without analysing the data) could provide a false narrative to the customer that AI is not a black box and we are completely in control. Unfortunately, that is not true today. AI is in fact still very much a black box that we can control only to a certain extent. Model accuracy is highly dependent on underlying data. Hence educating the customer of these dependency forms a critical first step for most AI projects. Moreover, it is important to take into account the output deviations that exists in AI projects and educate the customer of these deviations
- **Ensure a smooth handling of exceptions:** Testing every permutation and combination in an AI project is often not possible. Hence, it is important to ensure that an exception-handling scenario exists to ensure that in the worst scenario, the BAU process is followed. Handling of such exceptions needs to be done in a seamless manner to ensure that customer experience is not lost in the whole process.
- **Risk based testing and the need for QA engineers to understand AI technology:** It is not possible to test every scenario in an AI solution. Furthermore, in order to break something you need to understand the foundations on which the code is built on. Traditional application built their code/use case based on business. As such, traditional QA engineer needed to understand business. However, in the AI world, the AI application is built on data and underlying probability algorithms. Hence, QA engineers with an understanding of the underlying data (basic data analytic skills) and some deep learning/machine learning principles would have a better intuition than those QA

engineers with no AI background. With limited time available, this intuition would play a key role in capturing critical bugs early on in the testing phase.

- **Training datasets needs to evolve as well:** In traditional projects, creating data sets is not often a continuous process. It's built during the initial phases of the projects and is used during the testing phase. However, in AI projects, data plays a critical role. As the trend of the incoming data changes over a period, it become imperative for the testing to capture these trend changes in their testing data as well. Teams needs to incorporate this change in their testing strategy.

Conclusion:

QA teams in traditional projects had a mind-set to see things as black and white. However, in the AI era, solutions outputs are not perfect. Seventy percent accuracy with a feedback mechanism might be an acceptable metric going forward. QA activities will undergo a dramatic shift in the AI era. QA engineers need to change their mind-set, learn AI technology concepts, bring changes to existing processes and manage customer expectations. QA engineers certainly have their work cut out in the AI era!

About the authors

Mr. Sojan George has over 12 years' experience in the IT industry and has been predominantly associated with the Artificial Intelligence domain. He currently works as a Business Development Manager at Tata Consultancy Services for the Artificial Intelligence Practice. Over the last 6+ years, he has interacted with multiple customers, across domains, in solving their pain points leveraging AI techniques (like Deep Learning, Shallow Learning, Natural Language Processing) and has helped shape their AI journey. He has completed his BTech from Mar Athanasius College of Engineering, Kothamangalam, Kerala and his MBA from Leeds University Business School, United Kingdom.

Mr. Rajeev M Azhuvath is a hands-on technologist with 19 years of experience. Presently he is part of the Artificial Intelligence (AI) Program in TCS. Primary responsibilities include delivery of architecture focused on AI and building capabilities around shallow learning, deep learning, & natural language understanding. The right mix of consulting experience, delivery experience, servicing experience, research experience, & futurism gives him the unbiased perspective of technology and its impact. Additional areas of interest include advances in Nano Technology, Bio Technology, Information Technology, & Cognitive Science (NBIC). Special interest in Convergence of Technologies & Technological Singularity and its impact to humanity.

IEEE Computer Society's Top 12 Technology Trends for 2020

AI@Edge, non-volatile memory products, and digital twins lead the disruptive 2020 technology outlook

The top 12 technology trends predicted to reach adoption in 2020 are: (More at http://bit.ly/35IejA5)

1. Artificial Intelligence (AI) at the edge (AI@Edge).
2. Non-volatile memory (NVM) products, interfaces and applications.
3. Digital twins, including cognitive twins.
4. AI and critical systems.
5. Practical delivery drones.
6. Additive manufacturing.
7. Cognitive skills for robots.
8. AI/ML applied to cybersecurity.
9. Legal related implications to reflect security and privacy.
10. Adversarial Machine Learning (ML).
11. Reliability and safety challenges for intelligent systems.
12. Quantum Computing.0