



MODERNIZING ENTERPRISE SYSTEMS IN HEALTHCARE

An end-to-end testing approach for greater predictability and quality

Abstract

As digital technologies, smart wearables and remote monitoring capabilities penetrate healthcare, traditional healthcare companies are unable to keep up with end-user expectations. Under pressure to adopt rapid transformation, these organizations are looking for robust and end-to-end testing procedures. This paper explains various end-to-end testing approaches within the four main modernization techniques for healthcare companies. The analysis presented here acts as a guideline for healthcare leaders to make strategic and informed decisions on how to modernize their systems based on the needs of their end-users.

Introduction

Sustainability in healthcare is a looming challenge, particularly as the fusion of disruptive innovations such as digitization, Internet-of-Things and smart wearables enable remote and real-time health tracking, diagnosis and management.

To succeed in such an environment, healthcare organizations rely heavily on IT. Thus, using the latest end-to-end testing approaches becomes essential to:

- Ensure that all applications operate as a single entity with multi-module interactions
- Maintain performance/non-functional scenarios within the desired limit
- Identify bottlenecks and dependencies ahead of time so that the business can take appropriate actions



Testing challenges in healthcare modernization

Business transformation in healthcare is complex because of the challenges in maintaining integrity between different types of customer needs and health-related plans. Modernizing healthcare software applications mandates enabling multi-directional flow of information across multiple systems, which can complicate the entire healthcare workflow application.

Further, failures or errors in systems outside the enterprise environment can adversely affect the performance of applications with which they are integrated.

To address such challenges, it is important to determine the right method and types of end-to-end testing. This will optimize application performance by testing it across all layers from the front-end to the back-end along with its interfaces and endpoints.

Typically, most healthcare organizations use multi-tier structures with multiple end-users, making end-to-end testing very complex. Launching a new product in such a multi-directional business scenario requires extensive user testing. Thus, to enable end-to-end (E2E) testing, health insurance companies must first understand what customers expect from their healthcare providers and identify how they can meet these expectations in shorter timelines.

Pictorial Diagram of Multi-tier Healthcare Business

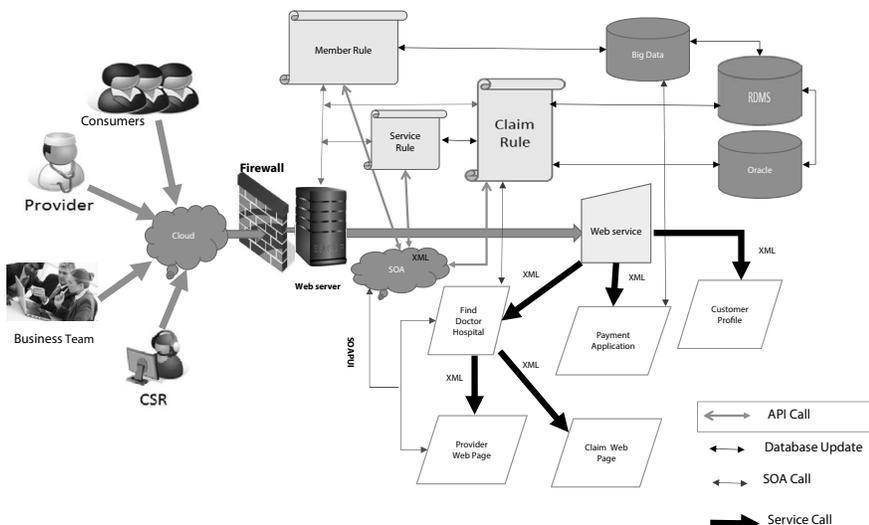
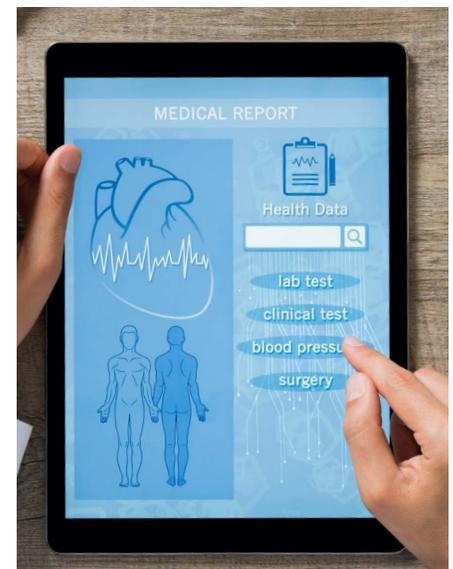


Fig 1: A typical multi-tier healthcare business



Infosys solution

End-to-end testing approaches for different modernization techniques

Infosys leverages four modernization techniques to help healthcare organizations enable end-to-end testing. These techniques are:

1. Re-engineering technique

Use-case: Best-suited in cases where companies need to digitize healthcare product marketing to different constituents of a state through online retail.

This modernization technique is useful when venturing into new markets or when retiring obsolete technologies due to high maintenance costs. It leverages the following end-to-end testing approaches:

- **Simulation testing:** User-centric interaction testing is performed based on different behavior events like usability testing, cross browser compatibility and mobile testing
- **Compliance testing:** This testing is needed for security protocols and financial boundary testing as per mandates defined by state and central governments

- **Blend testing:** This combines functional and structural testing into a single approach and is essential for any healthcare digitization transformation strategy Universal automation: This is a new approach that automates the acceptance of changed features in applications through browser recognition, payment gateways, etc.

- **Risk-based testing:** This focuses on testing a few components or critical defects that are identified as high-risk functions, have significant complexity in business operations and can impact key features

- **Continuous automation testing:** Based on continuous integration and continuous delivery, this testing is for real as well as virtual features that are proposed for projects in minor transition

- **Recognition accuracy testing:** This tests non-textual data like images, pictorial figures, feelings, fingerprinting, etc., using a virtual augmented framework

Benefits – This testing approach provides higher returns on investment by nearly 80-90% in short spans within 5-7 iterations. It also improves co-ordination, accuracy and reusability of data in ensuing runs, thus providing a robust and reusable testing option through cutting-edge technology.

Risks – Diversified technology exposure is critical to support such big bang transformation and limited technical knowledge may result in uncovering fewer quality issues. Further, rebuilding the enterprise framework can be costly.

2. Replacing or Retiring technique

Use case: Best-suited when one needs to remove contract and legal documentation from healthcare insurance and hospitals to a separate online portal.

This modernization technique is used when there is a need for more control and accuracy. Migratory functions are clustered as units and can be renovated easily without disturbing other applications. Here, end-to-end testing focuses on components that undergo gradual replacement or are retired as described below:

- **Plug-and-play testing:** This is usually executed when testing teams employ different types of tools for automation scripting or when different types of technologies are involved in testing

- **Web service-based testing:** This is a mechanism or medium of communication by which two or more applications exchange data, irrespective of the underlying architecture and technology

- **Neutrality testing:** This is typically used when the existing platform is replaced with a new one without altering the final business outcomes or end-user experiences

- **Parallel testing:** This analyzes several applications or sub-elements of one application simultaneously and in the same instance using agile or waterfall models in order to reduce test time

- **Assembly testing:** This reveals precise interactions among modules as per user requirements. It is used when functions

are grouped into a logical entity and alliances are needed

- **Usability testing:** Usability testing covers learnability, memorability, adeptness, and customer satisfaction indices to determine how easy to use the application is for end-users

Benefits – This modernization approach provides more structure and control to end-to-end testing with 15-20% effort reduction. It ensures effective application testing with the option of reverting to native state on-demand when needed. Further, it requires only 5-7% effort for automation changes during build.

Risks – Project overrun can occur without proper supervision. Additionally, it requires repeated testing of the same regression suite even for small deployments.

3. Re-fronting technique

Use case: Best-suited when adding encryption logic protocol is required for sensitive claim-related information passing through a web service.

This approach is used when end-users want to use the same data efficiently and quickly without investing in expensive infrastructure set-up. It covers virtualization, non-functional testing and regression testing as described below:

- Virtualization testing: This simulates multiple users to check the performance

of the new technology while it interacts with existing applications

- Non-functional testing: Certain features like technology compatibility, platform integrity, exception handling, help analysis, impact exploration, and application availability falls under the purview of non-functional testing
- Regression testing: Regression re-run approach is used when there is a slight change in functionality but the overall system behavior has not changed

Benefits – This approach simplifies localized defect resolution. Here, end-to-end testing is more stable as changes are limited and specific. Further, the cost of running E2E test cases is lower as the regression suite can be easily automated.

Risks – Frequent patch changes can lower productivity and increase maintenance cost. Further, repeated testing of the same emergency bug fix can reduce long-term RoI.

4. Re-platforming technique

Use case: Best-suited when upgrading billing/payments databases to recent versions is needed due to license renewals.

Re-platforming of application modernization is primarily done in areas where businesses aim to minimize maintenance costs with cost effective technology. This modernization technique uses migration, acceptance, intrusive, and volume testing approaches as described below:

- Migration testing: This is used when ensuring data integrity is the most important factor during technology upgrades
- Acceptance testing: Acceptance testing ensures that applications being moved to a new platform have the same recognition intensities as before
- Intrusive testing: Also used as negative testing, this approach determines the effect of hosting unexpected variables into the system or overall application
- Volume testing: This evaluates the stability of applications by ingesting a huge number of records

Benefits – This approach simplifies end-to-end testing as predicted business outcomes are achieved. It lowers testing cost, thus reducing total cost of operations and time-to-market and does not require additional infrastructure or specialized licensing tools. Further, it increases testing penetration by reusing scenarios, data and execution strategies.

Risks – Re-platforming may warrant additional testing of critical business flows to ensure functional defects are caught early to avoid cost impact. Also conducting testing in the new platform requires proper training.



Comparative analysis of various testing approaches

The following table depicts a matrix of end-to-end test approaches along with modernization techniques in healthcare. The matrix illustrates which E2E testing method is best-suited to the four different modernization techniques. While 'yes' and 'no' represent absolute outcomes, it is important to note that 'maybe' results depend on how critical the business needs are and whether the approach is actually cost-effective when considering the overall business operations.

Table 1: Comparison of different end-to-end test approaches and modernization techniques

Modernization techniques for end-to-end testing approaches				
Testing Approach	Re-engineer	Remediate or replace	Re-front	Re-platform
Simulation testing	Yes	Yes	No	No
Compliance testing	Yes	Yes	No	No
Blend testing	Yes	Yes	No	No
Universal testing	Yes	Yes	Yes	No
Risk-based testing	Yes	Yes	No	No
Plug-and-play testing	No	Yes	Yes	Yes
Web service-based testing	Yes	Yes	Yes	Yes
Agile testing	No	Yes	Maybe	No
Parallel testing	No	Yes	Yes	No
Virtualization testing	No	No	Yes	Yes
Usability testing	Yes	Yes	No	No
Recognition testing	Yes	No	Yes	Maybe
Regression testing	No	No	Yes	Yes
Migration testing	Yes	No	Maybe	Yes
Assembly testing	Yes	Yes	Yes	No
Volume testing	Yes	No	No	Yes
Intrusive testing	Yes	No	No	No
Acceptance testing	Yes	Maybe	No	Yes

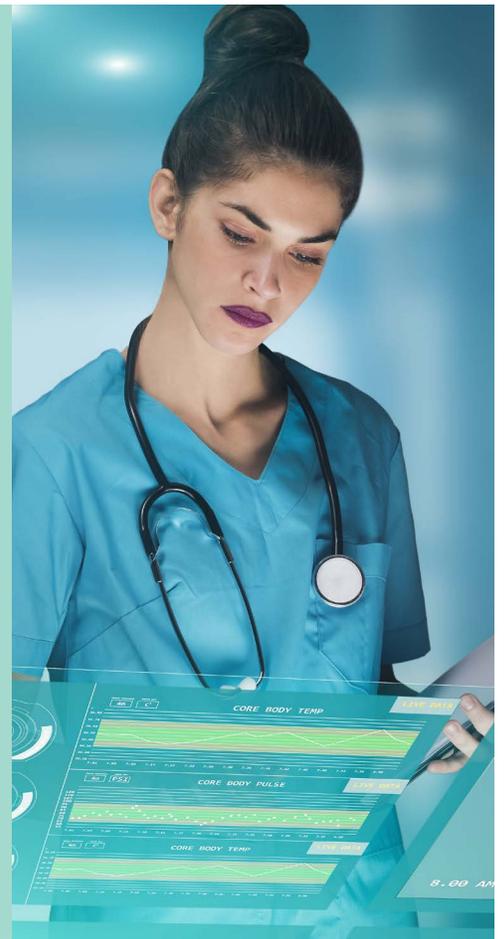
Case study

Business need – A healthcare company with three verticals for policyholders, doctors and claims wanted to remodel their business portfolio to adopt new technologies and meet customer demand. While the policy procuring system and the consumer premium collection system were digitized, the company decided to re-engineer the claims processing system from DB2 to a big data-based system. As the claims vertical interacted with doctors and hospital web portals, they also wanted to gradually transform the portals component-wise in order to give doctors sufficient time to acquaint themselves with the new digitized system.

Solution approach – To support the company's hybrid transformation project, we used an end-to-end testing strategy that leveraged complementary test approaches from different modernization

techniques across the three verticals, as described below:

- **For policyholders:** The customer procurement system was treated with a combination of re-front modernization and big bang transformation. Blend testing was used with continuous automation followed by web service testing and assembly testing
- **For providers (doctors/hospitals):** Here, we used a combination of assembly, regression rerun and agile testing to ensure gradual changes since agile testing methodology is best-suited for scenarios where constituents are deployed slowly over a period of time
- **For claims:** Claims is a crucial vertical. Thus, skeleton scripts, virtualization and migration testing methods were used for their stability and lower risk when migrating from DB2 to big data



End to End testing Approach		
Customer	Doctors	Claim
Blend	Assembly Testing	Skeleton scripts
Continuous Automation	Regression Re-run	Virtualization
Web Service Testing	Agile	Migration
Assembly Testing	Parallel testing	Regression Re-run
Non-Functional Testing for all the verticals		

Fig 2: End-to-end testing approach for the three verticals

As each vertical of the company has different business needs, different types of modernization were needed to suit various end-users.



The road ahead

In future, more consumers will embrace digitization and the uber connectedness of wearables and mobile devices that can track the user's health through in-built monitoring systems. Thus, as a higher number of service operators orchestrate multiple domains, we can expect to see greater challenges ahead for end-to-end testing. This makes it imperative to leverage DevOps and analytics-based testing capabilities along with modernization approaches.

Conclusion

Disruptive technologies are creating avenues for healthcare providers to issue virtual treatment, advice and services. However, this requires some degree of IT modernization for which end-to-end testing is crucial. There are various approaches that can be used to enable re-engineering, replacing, re-fronting, and re-platforming modernization techniques. Each testing approach has its benefits and risks and must be chosen based on the end-user expectations. Thus, it is important for business leaders to be aware of these in order to make the right decision for their IT modernization journey. The right approach can offer significant cost advantages, accelerate time-to-market and ensure seamless end-user experience.



Authors

Dipayan Bhattacharya
Project Manager
Infosys Validation Solutions

Amit Kumar Nanda
Group Project Manager
Infosys Validation Solutions



References

- Egui Zhu1, M., Anneliese Lilienthal1, M., Shluzas, L. A., Masiello, I., & Zary, N. (2015). Design of Mobile Augmented Reality in Health Care Education: A Theory-Driven Framework. *JMIR Medical Education*, 1-17.
- Flahiff, J. (2011, June 6). Integrating Agile into a Waterfall World. *The InfoQ Podcast*.
- Hewlett-Packard Development Company. (2012). *Survival guide for testing modern applications*. Hewlett-Packard Development Company.
- Infosys. (n.d.). <https://www.infosys.com/it-services/validation-solutions/white-papers/documents/end-test-automation.pdf>.
- Infosys. (n.d.). <https://www.infosys.com/it-services/validation-solutions/white-papers/documents/qa-strategy-succeed.pdf>.
- Karan Maini. (n.d.). *Legacy Modernization*. Retrieved from <https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwjSs5HssZnTAhUFSY8KHeKeDBoQFggiMAI&url=http%3A%2F%2Fdoczz.net%2Fdoc%2F5515301%2Flegacy-modernization&usq=AFQjCNFDRMLgmSlUWqaqhyCqR7XahprSBQ&bvm=bv.152174688,d.c2l>
- NetReach Technologies. (2010). *Legacy Application Modernization Balancing Risk vs. Reward*.
- Quality Thoughts. (2016, Oct 10). *Quality Thoughts*. Retrieved from <http://www.qualitythought.in/courses/webservices-testing/>
- Selenium Labs. (2016, Oct 10). *Web Services SOA Testing* . Retrieved from <http://www.seleniumlabs.in/web-services-soa-testing-in-bangalore.html>
- Slide share. (2016, Oct 10). *Webservices Testing*. Retrieved from <http://www.slideshare.net/AmitChanna/webservices-testing-a-changing-landscape>
- Transvive. (2011). *Migration Strategies & Methodologies*. Toronto: Transvive.
- (n.d.). Retrieved from <http://healthinsurancemedical.50webs.com/article1.html>
- (n.d.). Retrieved from https://www.google.co.in/search?hl=en-IN&biw=1280&bih=866&tbm=isch&q=3+pillars+of+financial+security&oq=&gs_l=#imgrc=uQhgXTjol03D8M%3A
- (n.d.). Retrieved from <https://www.google.co.in/search?q=computer+server+clipart&sa=G&hl=en-IN&biw=1280&bih=866&tbm=isch&imgil=W9NhyA0FXtloxM%253A%253B5cEEU2VgnYvcM%253Bhttp25253A%25252F%25252Fpublicdomainvectors.org%25252Ffen%25252Fpowerpoint-clip-art-server&source=iu&pf=m&fir>
- (n.d.). Retrieved from <http://www.google.co.in/imgres?imgurl=http://sacramentoappraisalblog.com/wp-content/uploads/2016/08/real-estate-market-future-sacramento-appraisal-blog-mage-purchased-and-used-with-permission-by-123rf-1.jpg&imgrefurl=http://paper.li/AngelaRunsAmuck/130652>
- (n.d.). Retrieved from https://www.google.co.in/search?q=virtual+reality+in+business&hl=en-IN&biw=1280&bih=866&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjP9MwOnM3QAhXIRY8KHbjBCwQ_AUICcgB#imgrc=VyCt_MfLFMQo0M%3A

For more information, contact askus@infosys.com



© 2018 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.