VIEW POINT

DELIVERING LARGE IT PROGRAMS – CHOOSING THE RIGHT IMPLEMENTATION APPROACH

Abstract

The odds of large IT programs failing remain high (estimates from different studies range from 45% to 70%). Despite countless program execution methodologies and best-practices, why do these programs still fail? What should organizations do to ensure successful delivery of their large IT programs?

In this paper, Jayanta Ghosh, who has over 18 years of experience in planning and execution of large programs across industries, outlines various factors that influence success of a large IT program with a particular focus on the importance of getting the implementation approach right at the outset.

Infosys®
Public Services

Many factors influence the success of a large IT program - technology, implementation approach, governance, program management, change management, and communication. Quite a few studies have stressed the importance of getting program management and governance right. While both factors are important, the pace of change driven by the digital imperative has made selection of an implementation approach equally, if not more, important.

In addition to being complex due to their sheer size and scope, large IT programs involve a lot of people and have a long execution timeframe. The world and the organization do not sit still while the program is underway. Initial assumptions may not hold true, requirements may evolve, or key stakeholders may change. The chosen implementation approach should enable an organization to adapt to these changes and ensure that the program gets executed on-time and on-budget.

More often than not, a large IT program involves modernization of legacy systems. It is not uncommon for agencies to face serious challenges while modernizing their system or realize that the system they built, after investing a lot of time, effort and money, is not fully adopted by their business users. Here is a real life example of challenges faced by an agency while executing a large legacy modernization program:

The agency opted to implement a leading commercial off the shelf (COTS) product to support its legacy modernization program. However, the COTS vendor updated product's roadmap and this delayed the rollout of a key product feature. The agency and the system integrator had already deployed people to implement functionalities built on that feature and were incurring significant costs to keep them engaged.

Halfway into the development phase, the team found a major integration issue between the product that they were implementing and an ancillary system. A key data element which was required by the interfacing product was missing from the COTS product.

Scope, which was initially fixed, changed as some design assumptions related to customization of the chosen COTS product were found to be incorrect.

Four weeks before the new system was supposed to go live, the team identified a few technical and functional gaps which had to be plugged quickly to meet the go-live date.

User acceptance testing (UAT) was not progressing as planned since user testers didn't know how to use the new system. Also, one of the partner departments/agencies was reluctant to sign-off on user acceptance testing (UAT) results since their requirements were not incorporated into the system.

All these challenges would have certainly derailed and delayed the program. However, the agency had adopted the 'right' implementation approach which helped them address all these issues and ensured that their system went live successfully as planned.

Choosing the right implementation approach

The right implementation approach balances predictable and timely execution with agility to adapt to changes. It ensures effective management of multiple interrelated tracks, breaks down the entire program into small, manageable pieces, keeps all the stakeholders engaged, and ensures successful implementation and adoption of the system by the users.

The right implementation approach includes the following key elements:

- **COTS/solution roadmap:** A basic minimum scope (also called minimum viable product in the product development world) should be defined for each release. This makes decision making easier and faster. If using a COTS product, the product roadmap of underlying platform needs to be incorporated into release planning.
- **Constraints:** Organizations should identify all the constraints – regulatory, funding, skillsets, etc. – early on in the planning cycle and create remediation strategies that minimize the impact of these constraints on each release.
- **User adoption:** Organizations must allow for adequate time between releases to manage the learning curve of users, encourage feedback, and facilitate adoption.
- **Post release support:** An elevated level of support is required during and after any production release. Business subject matter experts must work hand in hand with the technology team during the stabilization period.
- **Data migration approach:** Minimizing the need to maintain data in multiple systems is key to successful execution of many large programs. When it cannot be avoided, organizations should be aware of and should have a strategy to de-risk data corruption issues.
- **Methodology:** The methodology for design, development, testing and implementation may be either waterfall or agile options described in the next section.

Both the methodology and the implementation approach should incorporate an adequate level of verification and validation to mitigate errors/risks as well as have clear visibility into the progress of the project.

During the system development methodology stage, reviews of artifacts (verification) and testing of products at various levels of development and integration (validation) ensure that defects or deficiencies are identified and eliminated as the project progresses.

This is an important activity that should not be ignored or sacrificed for a crashed schedule.

Also, project executives should ensure that the testing process and practices are independent and not influenced by the development team (or any other interests), and that they have a separate and controlled test environment to test the work product. The real value one gets from an independent and effective testing function is the view of the product and prioritization of the activities before go-live; for example, defects to fix, workarounds to develop, and additional communications needed. Exit gate reviews are critical governance mechanism to ensure the project is ready to move to the next phase.

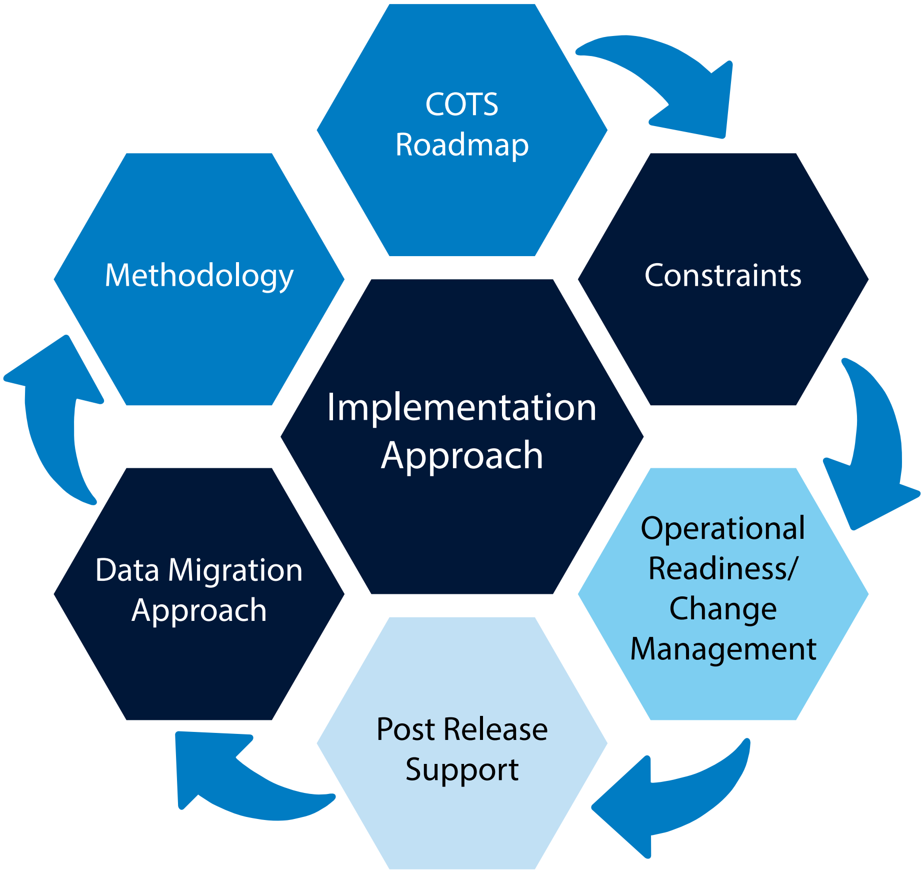


Figure: 1 Considerations for selecting an implementation approach

Waterfall vs. agile: Which approach should an agency choose?

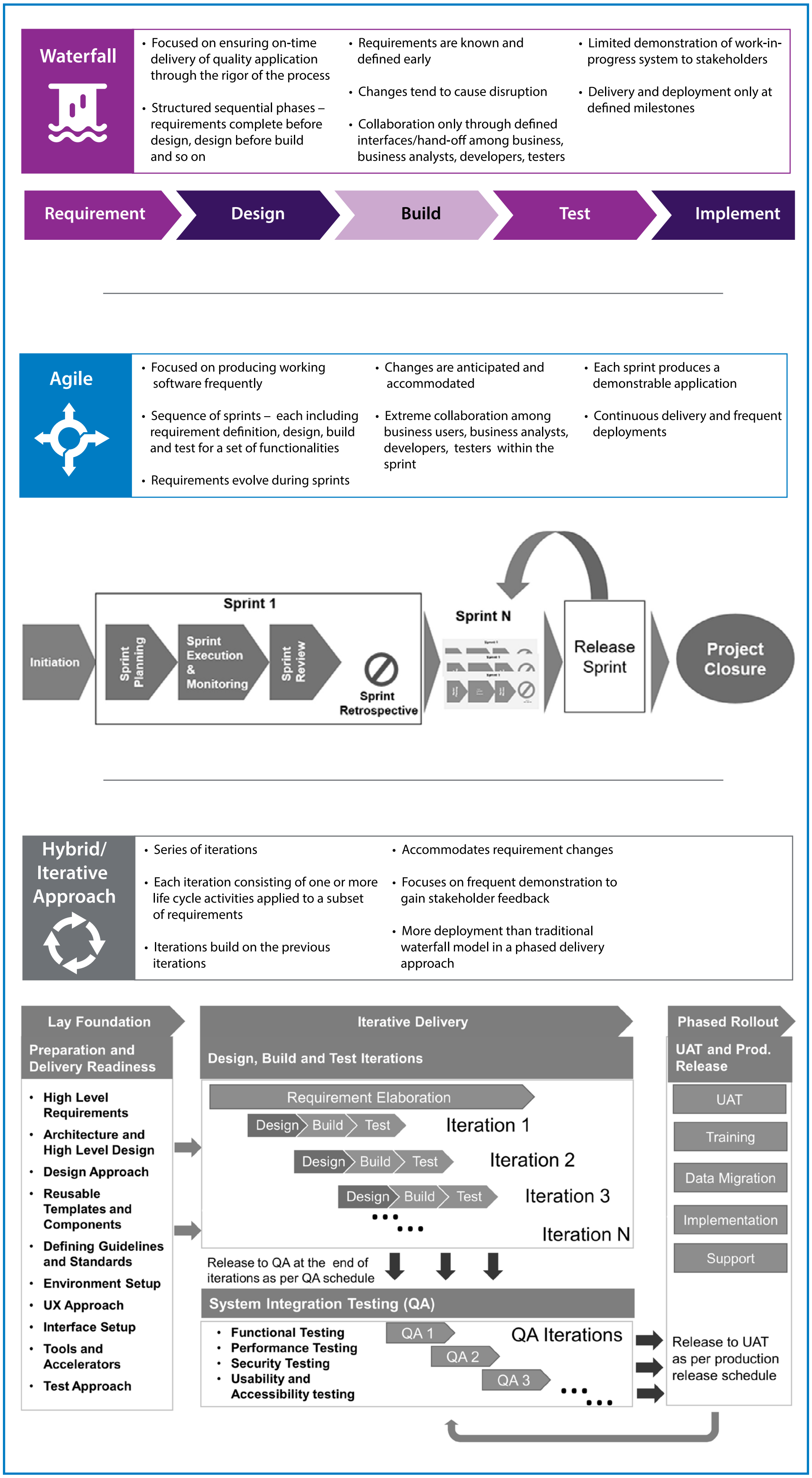
We often see public sector organizations deliberating between two approaches to execute their large IT program – waterfall and agile.

Traditional waterfall methodology provides more control but increases the risk of late surprises when it comes to meeting stakeholder expectations. Many proponents of the agile world believe that the trade-off of speed versus process rigor is worth it.

Agile bridges the gap between the requirement/client expectation and a solution that evolves through rapid and frequent feedback. True agile, however, requires significant investments from an organization in tools and user training. It also requires a cultural shift in the way projects are executed which, more often than not, is the most difficult challenge to its adoption. So which implementation approach is right for organizations, especially those not fully geared up for the agile world?

The iterative approach offers the best of both the waterfall and agile worlds – it offers speed of execution and at the same time provides enough headroom to incorporate changes and get stakeholders' buy-in.

The iterative approach consists of a series of mini-waterfalls that are executed rapidly. Given the change impact of any large system deployment, especially in case of public sector organizations, it is best that the rollouts happen in phases in order to minimize the impact on operations and provide sufficient time for users to become comfortable with the new system. Figure 2 highlights how the iterative approach scores over traditional waterfall or the new-age agile methodologies.



Waterfall

- Focused on ensuring on-time delivery of quality application through the rigor of the process
- Structured sequential phases – requirements complete before design, design before build and so on
- Requirements are known and defined early
- Changes tend to cause disruption
- Collaboration only through defined interfaces/hand-off among business, business analysts, developers, testers
- Limited demonstration of work-in-progress system to stakeholders
- Delivery and deployment only at defined milestones

Agile

- Focused on producing working software frequently
- Sequence of sprints – each including requirement definition, design, build and test for a set of functionalities
- Requirements evolve during sprints
- Changes are anticipated and accommodated
- Extreme collaboration among business users, business analysts, developers, testers within the sprint
- Each sprint produces a demonstrable application
- Continuous delivery and frequent deployments

Hybrid/ Iterative Approach

- Series of iterations
- Each iteration consisting of one or more life cycle activities applied to a subset of requirements
- Iterations build on the previous iterations
- Accommodates requirement changes
- Focuses on frequent demonstration to gain stakeholder feedback
- More deployment than traditional waterfall model in a phased delivery approach

Maturity matters

Often the capability and process maturity of the project and the organization make all the difference when it comes to a successful implementation. Mature projects and mature organizations offer much more predictable outcomes.

Agencies can leverage the CMMI maturity level to assess the capabilities of the implementing organization which is a direct indicator of the performance, quality and cost-effectiveness.

Leveraging iterative approach for a large IT program – an example

Let's see how an agency used an iterative approach to deliver a large IT program – the modernization of a legacy eligibility system for health and social programs.

After due deliberation, the agency selected a COTS product to modernize its legacy integrated eligibility system, however, the COTS product itself was continuously evolving to address changing government regulations and mandates. The product roadmap indicated that some of the key features might not be available before the defined go live date.

The go-live date was a hard constraint that had been communicated to the citizens. Since the requirements were driven by evolving federal requirements and standards, the interpretation of the mandates/rules and corresponding business processes also kept changing. There were additional constraints around availability of skilled staff, mandated gating reviews, and the target technology environment. The first release was to go-live in less than a year from the project start date. It was going to be an uphill task to meet that deadline and build the required system using a traditional implementation approach.

Given the strong possibility of changes being introduced while the execution was underway, the project team worked with the client stakeholders and product vendors to develop and execute the program using an 'iterative' approach:

- **Implementation planning and prioritization:** The teams prioritized only those functionalities that were absolutely necessary for Day 1 operations post go-live. For example, application intake through public facing portal, rules for determining eligibility and benefits, caseworker's workflow, plan selection, communication / letters for the first month of operations, etc. The entire program and releases were aligned with the product roadmap – each release was planned to take advantage of the latest COTS version and functionalities.
- **Methodology:** The first release was delivered in multiple iterations each focusing on a set of functionalities like intake, eligibility, plan management and shopping experience, notices etc. Each iteration followed the life cycle stages of design, development and testing. The functionalities for the iterations were sequenced in a spiral fashion to progressively build the functional and technical foundation. For example, initial iterations focused on developing

the external interfaces which provided adequate time to correct all interface issues with external agencies.

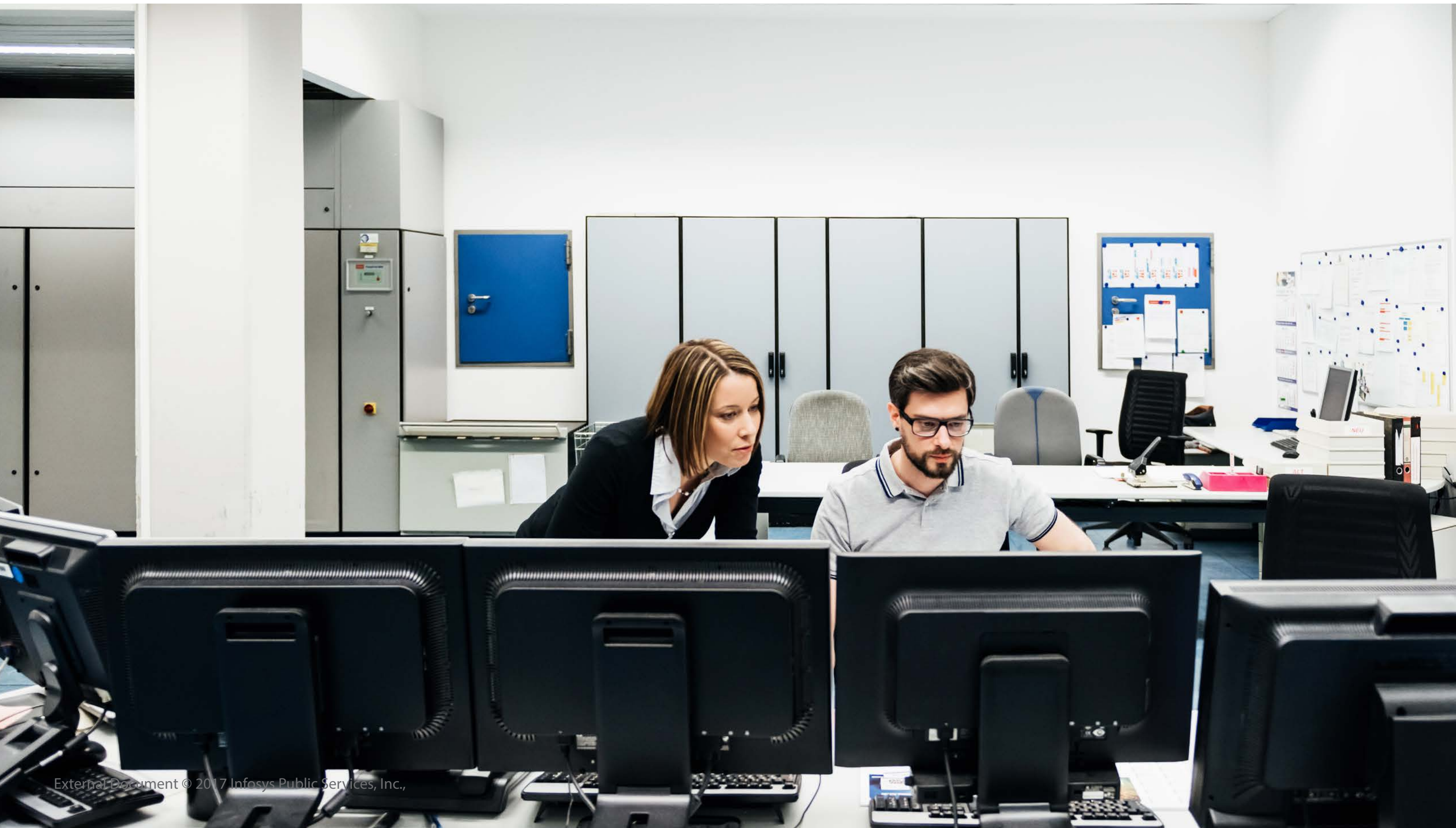
- **Testing and validation:** An independent team tested the output from each iteration and the integrated product as it evolved. Defects or deficiencies were added to the backlog and triaged frequently. An end-to-end testing phase helped eliminate cross-functional issues not identified during the iterations. Defect backlog and prioritization was an important tool for risk management as well as preparation for change management and operations post go live. Workaround and job-aids were developed for unresolved deficiencies to reduce operational impact.
- **Operational readiness:** As part of operational readiness, users were trained in a dedicated training environment. Training activities started before the release using a train-the-trainer approach. Each subsequent release was supported by a proactive set of training activities starting with a training plan and followed by content development, data setup, and training delivery. In addition, necessary backup plans were developed to address any major system issues that might arise.
- **Post-go live support:** In the stabilization phase, a command center was set up to provide hyper-care. Issues that had significant impact on the users and operations were planned and executed separately as minor projects.
- **Data migration approach:** Data migration for the legacy cases was planned and implemented as a separate work stream that followed the system development in order to utilize a more stable version of the target application. Additionally, there was a strong focus on maintaining data integrity between releases.

Conclusion

Successful execution of large IT programs requires careful consideration of various factors like technology, implementation approach, governance, program management, change management, and communication. In this paper, we discussed the importance of the implementation approach and the key aspects that an organization should consider while adopting the right implementation approach.

The iterative approach – a combination of 'traditional waterfall' and 'new age agile' – balances predictability and agility to adapt to new requirements or regulatory changes. In our opinion, this approach is the right approach for most of the agencies today. It can help agencies break down large IT programs into interconnected pieces and implement the program as a series of releases.

Future papers will explore other factors in detail, highlight the interplay between them, and discuss their influence on the chances of success.



About the author

Jayanta Ghosh has over 18 years of experience in delivering technology solutions for healthcare, public sector and insurance organizations. He has managed large, complex IT programs and service delivery for healthcare payers, insurance and government organizations, managed validation for health benefit exchange implementation, and also led the pre-sales team. Currently, he leads the Quality Assurance practice for Infosys Public Services which provides an independent verification and validation of the large system implementation programs, provides program health visibility to executive stakeholders, and enables implementation of standards and best practices.

Infosys®
Public Services

For more information, contact askus@infosyspublicservices.com

Infosyspublicservices.com

Stay Connected