

TECHNOLOGY TRANSFORMATION: Conclusions from an Iterative Learning Process



Table of Content

No.	Content Title
-----	---------------

01.	Abstract
-----	----------

02.	Introduction
-----	--------------

03.	The Pitfalls of Transformation
-----	--------------------------------

04.	The Transformation Model
-----	--------------------------

05.	The Execution Approach
-----	------------------------

06.	The Cybage Experience
-----	-----------------------



Abstract

Technology transformation is a crucial decision in the lifespan of any business and comes with several complexities arising from diverse sources and circumstances. In this whitepaper, we uncover the process of transforming the technology component of a running business and the associated pitfalls that an organization needs to dodge to ensure a path to success.

We also discuss how successfully addressing these complexities requires the use of a mature model that has been perfected over time by implementing it for 200+ clients. This whitepaper concludes with insights into the model adopted and details how it helped in turning large technology transformation projects into result-driven processes rather than ad-hoc tangles.

Introduction

Change is the only constant!

Today, any successful company needs to keep updating its tech investments to sustain itself, stay competitive, and continue scaling.

Every year the IT spend of enterprise companies is increasing with a core focus on significantly digitalizing their internal and external operations. Many companies are already searching for bleeding edge technology to solve disruptive problems for their customers, while others are just looking to optimize their incumbent processes and costs.

Even for companies whose businesses are IT-driven, the growing technical debt due to the existing software, together with the need for seamless integration of external services, are major factors why many software components need constant transformation. Whatever the case, it is plain as day that adopting the latest technology is going to be a key focus area

The Pitfalls of Transformation

Every sizeable enterprise often has at least some technology that fits within the “legacy” category, which means the technology used to build it is close to the end of life (EOL). Software engineering teams often find it challenging to maintain or scale better growth in such cases. Such enterprises and ISVs have often been part of mergers/acquisitions, and as a result, they are most likely dealing with technologies of various levels on maturity and automation thrown into a common mix.

Most commonly, businesses realize a specific set of challenges amid the transformation exercise, but certain factors form a typical pattern for hindrances and often fall under these categories:

Widely observed transformation pitfalls

Ignoring interdependencies

Transformation exercises happen on already running large systems and are deeply ingrained in an enterprise's culture. Certain aspects, such as migrating to the cloud without considering cybersecurity or DevOps, end up being taken for granted. The resulting gap from such instances limits flexibility.

Architecture modernization changes how the development team deploys solutions. If not in sync, this causes delays and errors.

A missing governance model

Enterprises & ISVs must keep business objectives in mind and ensure every initiative is P&L-oriented (with KPIs in place for tracking). We often observe a strong emphasis on engineering strategies compared to the governance model, which impacts the KPIs.

Proper governance empowers better coordination and sharing of digital initiatives.

Added costs

Technology transformation initiatives are expensive, and not targeting low-hanging fruit initially can lead to a lack of demonstrable value.

Well-planned/successful transformation should unlock value quickly.

Not identifying business value

The growing ocean of solutions leads to a lack of precise business value derived from transformations. For example, the time for completing a workflow or clicks is reduced. But converting this saved time into bottom-line revenue is difficult.

Identifying business value at the beginning of initiatives and maintaining objectivity is vital to success.

The Transformation Model

The Cybage-designed model considers interdependencies between business activities while focusing on three main phases.

Phase I Envision the Role of Technology

Keeping a technology-centric model when making any business decisions is the key. Organizations first need to think through a tech forward business strategy, user experience, and integrated products and platforms. It is vital to evaluate the impact of technology across different business functions upfront, as this will help define how technology can drive business value.

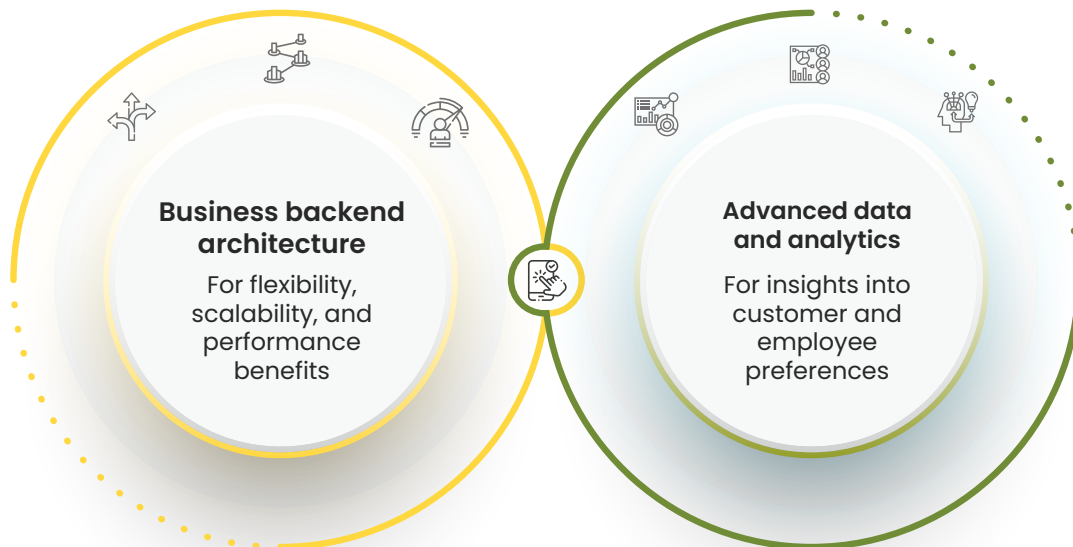
Phase II Outline A Technology Delivery Model

Every organization's delivery model should be set up to cater to the ever-changing technological needs of its business and consumers. An Agile @ Core culture can ensure that activities with potential are prioritized accordingly. Next-gen infrastructure services (e.g., cloud-based infra) can drive the stability, flexibility, and scalability required for future needs. In addition, a strong product engineering background and the right skillset are necessary to bring in quality and efficiency and to carry out transformation successfully.

“ Having the right set of partnerships brings in the right set of capabilities to carry out transformation successfully—such an expertise is more challenging to develop from scratch or scale. ”

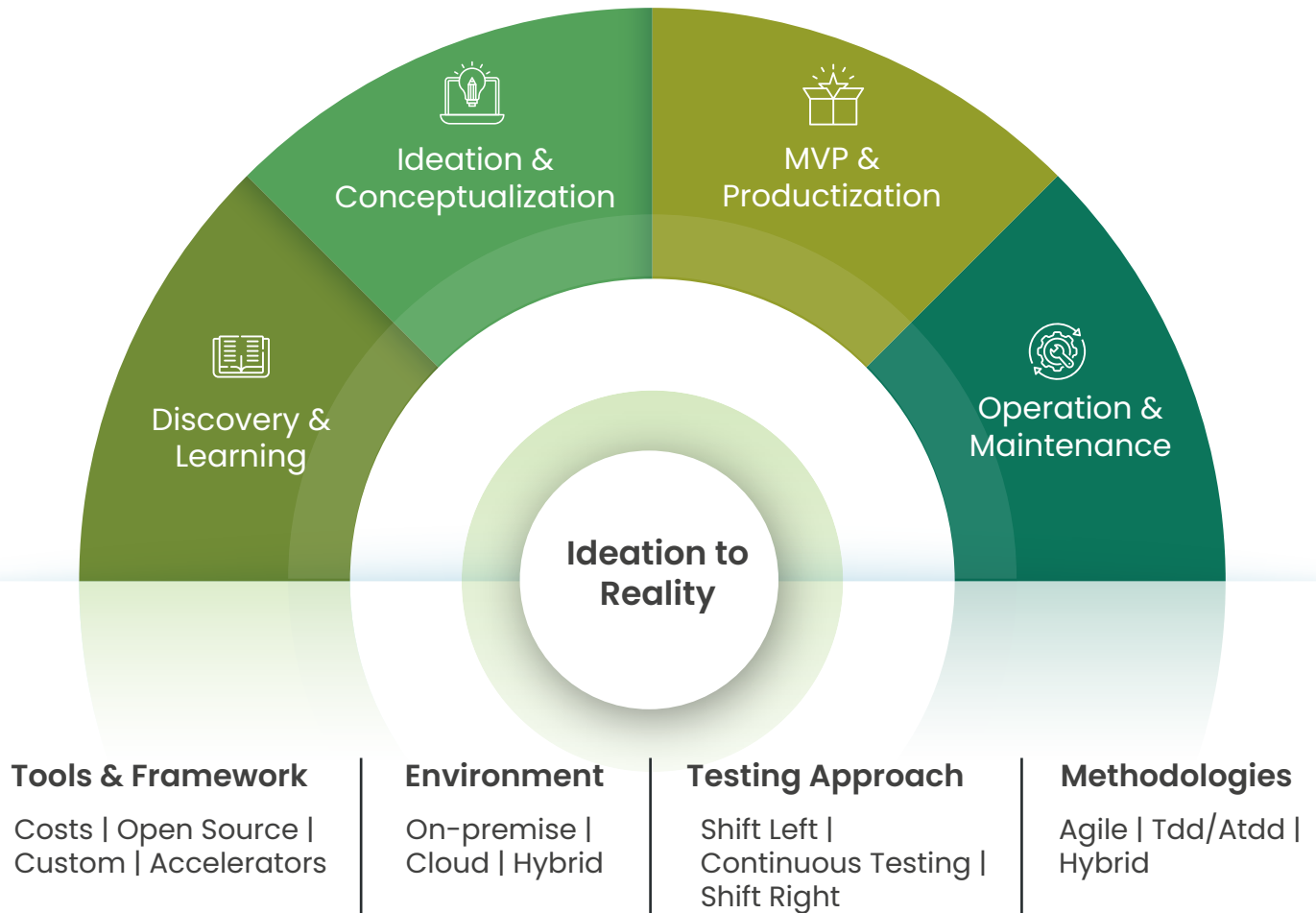
Phase III Build A Robust, Future-Proof Foundation

The Elements of an Enhanced User Experience



Enhanced user experience empowers teams to enable and integrate security measures while implementing new systems rather than having security as an add-on feature. This approach further enhances information security and accelerates delivery. The success of this transformation model lies in coordinating or managing only interdependencies mentioned above while adequately sequencing their initiatives to derive business value quickly. The organization needs to unambiguously know two things: (1) how it wants to be positioned in the future and (2) which existing capabilities can deliver successful transformation.

The Execution Approach



Through its vast experience in delivering technical transformations, Cybage has developed a reliable execution approach, the key to which is identifying suitable personnel to start with. Once the stakeholder & engineering teams are in place, we follow this four-step approach:

Discovery & Learning

- ▶ Requirements gathering is the primary step of the software development lifecycle. While some requirements are evident and known to customers beforehand, others are often discovered after discussions and interviews with relevant stakeholders.
- ▶ In a **transformation exercise**, it is critical to baseline the existing systems' current state (aka the "**AS IS**" state). Having a vision of the "**TO BE**" state is also vital when plotting a roadmap to identify applicable milestones.
- ▶ Discovery comprises **Functional** and **Non-Functional Requirements (NFR)** that help cover future scaling goals.
- ▶ The **tools & technologies** also have to be identified by keeping these goals in mind.
- ▶ From the engineering teams, the appropriate **technical architects and functional experts** from the core group are required to carry out this phase.
- ▶ Specific use cases demand the creation of **clickable prototypes**. For example, if the user interface of existing systems needs modernization, dummy screens can help the stakeholders confirm any usability elements.

Ideation & Conceptualization



- ▶ In the second step, solutioning views for design blueprints need to be created. These views can be architectural or business function-specific and should be used as team guidelines.
- ▶ **Architectural diagrams and descriptions** of the framework for development are the key artifacts from this step.
- ▶ At this point, apart from **implementation, testing and deployment strategies** should also be designed.
- ▶ To prove the concept and effectiveness of the solution, Cybage recommends the **Tracer Bullet approach**. This helps create a workable "Hello World" state of the new system to its feasible extent. This also demonstrates how the chosen implementation & deployment stack can bring the architecture on paper to life.
- ▶ Another critical step here is identifying the KPIs to measure the success and progress of a transformation exercise. This also helps quantify the success of the "TO BE" state after it is built.

MVP & Productization



- ▶ The entire team is assembled at this stage as an extension of the Tracer Bullet exercise. An **MVP (Minimum Viable Product)** should ideally be built before the implementation/transformation of the "TO BE" state. The requirements of an MVP will be a subset of the original conditions determined in the discovery stage, but all stakeholders will also need to agree on a product's essential features.
- ▶ This is an iterative step that requires multiple iterations of sprint planning. So, the assembled team focuses on **building the most critical parts** of the system, not on nice-to-haves. Feedback from relevant stakeholders must be gathered periodically in demos and retrospective meetings.
- ▶ There must be a strong emphasis on NFRs to avoid any unwanted surprises at the end. Adopting correct **observability tools** by using **appropriate logging and monitoring tools** should also be done at this stage, so when features get added, there are ways to figure out issues and solve them in a short turnaround time.
- ▶ Once MVP is ready and signed off, teams can add the relevant features to it with a focus on quality.
- ▶ **Knowledge management** plays a vital role throughout this stage. Appropriate tools for documentation of functional and non-functional parameters of the system, backed by a robust governance model, can help ensure that all teams follow the process.



- ▶ No software project, be it greenfield or transformation, can be successful without governance. Appropriate tools must be in place for significant enterprise system transformations to avoid manual overlooks.
- ▶ When the "AS IS" system lacks monitoring tools, the first discovery step will highlight the need to develop a relevant toolset for **continuous monitoring**. Systems (hardware & applications), people, and project progress will all need to be observed and must follow the progress of the exercise.
- ▶ Automating quality assurance, reporting & release processes via continuous integration & delivery are critical aspects of optimizing the operations. As a result, focus teams need to be deployed.
- ▶ Lastly, feedback on reports containing key metrics will be presented to the appropriate stakeholders. For example, Software Architects should see NFR status. Program Managers should see burn-down charts, bug fix rate, resource bandwidth & sprint velocity reports. And the higher-level executives should see expenditure reports-Cap-Ex/Op-Ex, ROI on IT spend, the status of quarterly goals, etc.

The Cybage Experience



Cybage has been successfully undertaking and implementing technology transformation projects for customers from diverse areas of interest. As a result, we have developed and improved this transformation model, which has proved effective across multiple use cases, irrespective of the domain.

These assignments are generally complex as we have to work on already running businesses and can't afford any downtimes or disruptions to existing customers. This is similar to performing an open-heart surgery! Hence attention to detail, custom strategies, and time-bound tactical decision-making are crucial to success.

Some examples of Cybage's extensive, complex and successful technology transformation assignments:

Re-architected & rewrote the core software piece for a leading Southeast Asian bed bank in Travel and Hospitality by



Empowering worldwide distribution with an additional NFR improvement to meet higher requests per second in the future

Completing organization-wide security standards such as OWASP compliance

Building a robust DevOps culture

Converged two different technology stacks on the UI layer for a large customer in Media & Entertainment by



Completely transforming the platform UI design & visualization to bring a seamless experience for an acquired business

Utilizing the best possible UI tech to help improve the existing experience

Modernized & transformed technology for a leading Biometric Solutions Provider by



Improving the UX and performance parameters to support multiple devices

Driving various third-party integrations to assist subscription-based licensing & packages for their solution offering

Transformed a legacy Manufacturing Management Software from old client-server architecture to a cloud-based SaaS model by



Developing a scalable application for multi-client businesses installed on a single instance

Serving all clients from that instance to reduce the TCO (Total Cost of Ownership) & lower system maintenance needs