Department of Healthcare Management & Health Economics

ASSIGNMENT REPORT

MANAGEMENT OF NON-HEALTHCARE AREAS: OPTION 1 - LEGACY MODERNISATION, CLOUD ARCHITECTURE, AND ENTERPRISE AGILITY

Student Name:	Vaghani Paras Hareshbhai
Student ID:	DE240399
Programme:	Healthcare Management & Health Economics
Module:	Management of Non-Healthcare Areas (M07)
Submission Date:	19 / 10 / 2026
Institution:	EIIET (European Institute of Innovation)

Table of Contents

1. Abstract Page 2

2. Introduction Page 3

3. Literature Review Page 3

4. Methodology Page 4

5. Findings & Technical Analysis Page 5

6. Discussion Page 6

7. Strategic Governance & Migration Framework Page 7

8. Conclusion & Recommendations Page 7

9. References Page 8

10. Appendices Page 8

1. Abstract

This report examines the strategic and operational challenges that modern enterprises face when updating their old technology systems to achieve enterprise agility. In today's digital business environment, reliance on legacy IT infrastructure creates severe technical debt, data silos, and structural rigidity. These limitations prevent organisations from responding quickly to market changes and customer demands.

The paper evaluates the technical shift from monolithic system architectures to modern cloud-native, microservices-based environments. Through real-world case studies, such as Netflix's successful transition to the cloud versus public technology failures like the TSB Bank migration outage, this report highlights critical factors that lead to success or failure in IT transformation projects. Furthermore, architectural governance models and organizational change management methods, including Kotter's 8-Step Model, are discussed to provide senior executives with actionable strategies for guiding major digital modernization initiatives.

2. Introduction

In the modern corporate world, digital technology is no longer just a support tool for operations; it has become the main driver of competitive advantage and strategic growth. Organizations across global industries must adapt quickly to changing customer expectations, emerging market threats, and new digital business opportunities (Bharadwaj et al., 2013). However, many established enterprise organizations still rely heavily on legacy technology ecosystems built several decades ago.

Legacy IT systems often create significant operational challenges. They generate accumulation of technical debt, maintain isolated data silos, and suffer from architectural rigidity that limits business speed (Ward & Peppard, 2016). When business leaders attempt to launch new digital services, these older systems make changes slow, costly, and risky. Therefore, modernizing legacy systems and adopting cloud-native microservices architectures have become critical strategic priorities for senior executives. This project report explores the architectural transition from legacy monoliths to modern cloud environments, examines corporate case studies of success and failure, and presents a governance framework to manage large-scale technology transformations.

3. Literature Review

Academic literature on information systems management emphasizes the concept of Enterprise Agility—the ability of an organization to sense environmental changes and respond quickly and effectively (Overby et al., 2006). Achieving agility requires alignment between business strategy and underlying IT enterprise architecture.

Technical Debt and Legacy Systems: Technical debt represents the implied cost of future rework caused by choosing an easy, short-term solution instead of using a better, long-term technical approach (Kruchten et al., 2012). Over time, as legacy systems receive temporary patches without proper re-engineering, technical debt increases. This leads to system brittleness, where small code modifications cause unexpected failure in unrelated parts of the software.

Monolithic vs. Microservices Architecture: Traditional enterprise software relies on monolithic architecture, where all business functions, database access, and user interfaces are bundled into a single tightly-coupled codebase (Newman, 2015). While monolithic systems are simpler to build initially, they become difficult to scale and maintain as the organisation grows. In contrast, cloud-native microservices architecture breaks applications into small, independent services that communicate via standardized Application Programming Interfaces (APIs). Each service can be developed, deployed, and scaled independently, enabling continuous deployment and operational agility.

According to Lewis and Fowler (2014), microservices allow organizations to align IT teams with specific business capabilities. This alignment speeds up software release cycles from months to days, reducing time-to-market for digital products.

Architectural Feature	Monolithic Systems (Legacy)	Microservices Architecture (Cloud-Native)
System Coupling	Tightly coupled; single unified codebase	Loosely coupled; independent functional services
Deployment	Risk of full-system downtime during updates	Independent service deployments with zero downtime
Scalability	Monolithic scaling (entire system must scale)	Granular scaling (only heavy-demand services scale)
Data Management	Centralized monolithic database (data silos)	Decentralized data stores per microservice
Release Frequency	Slow releases (monthly or quarterly)	Fast releases (multiple times per day via CI/CD)

4. Methodology

This project report uses a secondary research design based on qualitative literature review and corporate case study analysis. Data were gathered from leading peer-reviewed business and technology journals (e.g., *MIS Quarterly*, *Harvard Business Review*, *IEEE Software*) and enterprise technology industry reports from organizations like Gartner and McKinsey & Company.

The evaluation process involved comparing architectural frameworks (Monolithic vs. Microservices) and analyzing global enterprise case studies representing both successful modernization (Netflix) and failed migration executions (TSB Bank). Findings were synthesized to identify core success factors and operational risks in enterprise technology transformations.

5. Findings & Technical Analysis

The research findings show that legacy modernization is not merely a technical software upgrade; it is a fundamental shift in enterprise operational capability. Organizations that retain legacy systems face rising maintenance costs, as up to 75% of IT budgets are spent maintaining existing systems rather than building new innovations (Gartner, 2021).

Structural Impediments of Legacy IT Infrastructure:

1. *Data Silos:* Legacy systems store business data in proprietary, isolated databases, preventing cross-departmental analytics.
2. *Architectural Rigidity:* Tightly coupled components make updating one feature dangerous for the entire system.
3. *Skills Scarcity:* Older programming languages (e.g., COBOL, Fortran) suffer from a shrinking labor pool as experienced engineers retire.



IMPACT OF DATA SILOS ON CORPORATE PERFORMANCE

When enterprise data remains trapped in legacy databases, executives cannot access real-time operational metrics. This slows down strategic decision-making, increases reporting error rates, and prevents the deployment of automated customer service platforms.

6. Discussion

To evaluate the practical outcomes of IT modernisation, it is useful to compare real-world cases of enterprise cloud adoption and migration execution.

Case Study 1: Netflix - Successful Cloud Migration

In 2008, Netflix experienced a major database corruption that stopped its DVD delivery service for three days. Following this incident, Netflix leadership decided to move its entire IT infrastructure from on-premise data centers to AWS public cloud (Izrailevsky et al., 2016). Over seven years, Netflix transformed its monolithic application into hundreds of cloud-native microservices. They built tools like "Chaos Monkey," which intentionally broke production services to test resilience. This migration allowed Netflix to scale its streaming platform globally, handle millions of concurrent users, and maintain high system reliability.

Case Study 2: TSB Bank - Failed Migration Incident

In April 2018, UK-based TSB Bank attempted to migrate 5.2 million customer accounts from its former parent company's (Lloyds Banking Group) legacy platform to a new banking system called SABRE, developed by its new parent company, Sabadell (Treasury Committee, 2019). The migration was executed over a single weekend ("big bang" approach) without sufficient testing. The transition failed severely: millions of customers were locked out of online accounts, saw incorrect balances, or experienced fraudulent transactions. The outage lasted for weeks, costing TSB over £330 million in regulatory fines and customer compensation, and causing severe reputation damage.

Analysis of Organizational Failures:

The contrast between Netflix and TSB Bank shows that technical strategy and migration style directly determine outcome. Netflix used a gradual, iterative migration strategy, while TSB Bank used a high-risk "big bang" cutover. TSB Bank's failure was caused by lack of testing, poor executive governance, and inadequate change management among technical teams.

7. Strategic Governance & Change Management Framework

Major technology transformations require strong executive governance and structured organizational change management. Senior leaders must combine technical refactoring with cultural transformation to ensure business alignment and minimize risk.

THE 4-STAGE ENTERPRISE MODERNISATION GOVERNANCE FRAMEWORK

Stage 1: Technical & Architectural Assessment

Audit all existing legacy assets to identify technical debt. Classify applications based on strategic value and migration complexity (e.g., Re-host, Re-platform, Refactor, or Retire).

Stage 2: Iterative Migration (Strangler Fig Pattern)

Avoid "big bang" migrations. Use Fowler's "Strangler Fig" pattern to gradually replace legacy monolithic features with microservices one service at a time, keeping risk low.

Stage 3: Organizational Change Management (Kotter’s Model)

Apply Kotter’s 8-Step Change Model to address employee resistance. Communicate a clear strategic vision, create short-term wins, and upskill existing staff in cloud technologies.

Stage 4: Continuous Architectural Governance

Establish an Enterprise Architecture Review Board to ensure all new software development follows cloud-native principles, security standards, and API management guidelines.

8. Conclusion & Recommendations

Modernizing legacy IT systems is essential for enterprises seeking operational agility and sustained competitive advantage. Retaining rigid monolithic architectures creates severe technical debt and exposes organisations to disruption from digital-first competitors.

Strategic Recommendations for Senior Executives:

1. Adopt an incremental migration strategy (Strangler Fig pattern) rather than high-risk single-event deployments.
2. Invest in workforce reskilling programs to transition legacy developers to cloud-native DevOps practices.
3. Establish unified architectural governance to prevent new technical debt from accumulating.
4. Align IT transformation milestones directly with measurable business outcomes, such as reduced feature release time and increased system availability.

9. References

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital business strategy: toward a next generation of insights. *MIS Quarterly*, 37(2), 471-482.

Gartner. (2021). *IT Key Metrics Data: Executive Summary*. Gartner Research, Stamford, CT.

Izrailevsky, Y., Meshenberg, A., & Becker, A. (2016). Completing the Netflix Cloud Migration. *Netflix Technology Blog*.

Kruchten, P., Nord, R. L., & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *IEEE Software*, 29(6), 18-21.

Lewis, J., & Fowler, M. (2014). Microservices: A definition of this new architectural term. *ThoughtWorks Architectural Papers*.

Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, Sebastopol, CA.

Overby, E., Bharadwaj, A., & Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, 15(2), 120-131.

Treasury Committee. (2019). *IT Failures in Financial Services*. House of Commons, London: The Stationery Office.

Ward, J., & Peppard, J. (2016). *Strategic Planning for Information Systems* (4th ed.). John Wiley & Sons, Chichester.

10. Appendices

Appendix A: Legacy System Modernization Risk Assessment Matrix

Migration Strategy	Technical Complexity	Operational Risk	Time to Value	Recommended Scenario
Re-host (Lift & Shift)	Low	Low	Fast (< 6 months)	Non-critical legacy workloads
Re-platform	Medium	Medium	Moderate (6-12 months)	Database upgrades & minor optimizations
Refactor (Microservices)	High	Medium (if incremental)	Long (12-24 months)	Core strategic business applications
Big Bang Replacement	Very High	Extreme	High Initial Risk	Discouraged due to high failure rate