JOURNAL OF DECISION MAKING AND HEALTHCARE
Volume 3 (2026), No. 1, 32–38
https://doi.org/10.69829/jdmh-026-0301-ta03
Tulipa Opera Scholarum

AN OPTIMAL INTER-UPGRADE TIME: A CASE STUDY OF BUSINESS INTELLIGENCE SOFTWARE

INDRIATI NJOTO BISONO[1], MARTINUS MARIYANTO[2], AND HANIJANTO SOEWANDI[3,*]

[1] *Industrial Engineering Department, Petra Christian University, Surabaya, Indonesia*
[2] *International Business Engineering Program, Petra Christian University, Surabaya, Indonesia*
[3] *MicroStrategy, Tysons Corner, VA, USA*

Abstract. This study investigates the strategic timing of software upgrades within large-scale organizations, specifically focusing on Business Intelligence (BI) systems. By analyzing a major industrial case study in Indonesia, we examine the transition from legacy web-based systems to mobile-first (tablet) architectures. We propose a mathematical model to determine the optimal inter-upgrade interval, balancing setup and testing costs against operational benefits. Assuming planning horizons of 10 years, our model suggests an optimal upgrade frequency of 2.5 years. These findings provide a quantitative framework for Information Systems (IS) executives to balance fixed infrastructure setup costs against the variable human capital costs of testing, facilitating more informed and cost-effective technological lifecycle management.

Keywords. Inter-upgrade Time, Decision Support Systems, Business Intelligence, Software Lifecycle Management, Optimization Modeling.

1. Introduction

In the contemporary enterprise landscape, data visualization serves as a critical catalyst for organizational performance. For large-scale entities such as PT. XYZ, the effective visualization of complex datasets is essential, as it empowers decision-makers to obtain a comprehensive situational overview without the cognitive burden of manual numerical processing. By synthesizing vast quantities of data into interpretable visual formats, users can achieve rapid information comprehension and accelerated interpretation. Furthermore, advanced visualization techniques enable senior management to detect nuanced trends and spatial-temporal patterns tailored to specific organizational objectives. Beyond pattern detection, Petrini and Pozzebon [7] argue that the successful management of BI involves more than technical implementation; it requires a strategic alignment between the analytical tools and the organizational culture to ensure data-driven insights translate into measurable performance.

The current BI infrastructure at the subject organization (PT. XYZ) relies on web browser access from personal computers. However, the increasing mobility of executives necessitates a transition to tablet-based dashboards. Research suggests that while performance metrics between tablets and PCs are comparable for direct screen manipulation [6], tablets offer superior accessibility in "on-the-go" executive environments, such as meeting rooms. This transition is supported by the work of Basole [2], who characterizes 'Enterprise Mobility' as a fundamental paradigm shift. Basole highlights that mobile access to enterprise data is no longer an ancillary feature but a strategic necessity that enhances the 'information reach' and agility of top-level management in dynamic environments.

It is not surprising that many companies are also trying to adapt and migrate their business applications from web to mobile platforms. In particular, this article focuses on our experience assisting

*Corresponding author.
E-mail address: mlindri@petra.ac.id (I. N. Bisono), martinus@gmail.com (M. Maryanto), and hsoewandi@microstrategy.com (H. Soewandi)
Accepted: February 25, 2026.

the company in migrating its web-based Business Intelligence application to a mobile-based solution. Unfortunately, we encountered challenges related to software upgrade timing because PT. XYZ was using an older version of the MicroStrategy Dashboard.

The migration process is frequently complicated by software versioning and the technical limitations of legacy objects. Within the MicroStrategy (MSTR) ecosystem, dashboards are typically constructed as either "Report Service Documents" (optimized for pixel-perfect formatting) or "Dossiers" (optimized for responsive analysis). Although MSTR 2019 introduced "Responsive Design" to facilitate multi-device compatibility, PT. XYZ continues to use legacy Report Service Documents—likely developed in MSTR 10.x or earlier—lacking the agility of modern Dossier objects. By the time of this study, the release of MSTR 2021 placed the organization two versions behind the current software iteration, presenting a significant barrier to optimization. While upgrading to MSTR 2020 or beyond would provide access to "Free-form Layouts" and superior responsive features, such a transition is often precluded by rigid organizational policies. At PT. XYZ, the institutional policy dictates a three-year upgrade cycle; having last updated in 2019, the next authorized system overhaul is scheduled for 2022. This temporal lag forces a reliance on less efficient legacy formats for mobile deployment. Software upgrades represent substantial investments, involving complex testing protocols and extensive User Acceptance Testing (UAT) that demand significant time and personnel resources. Consequently, a quantitative evaluation is required to identify a "good compromise" between technological currency and cost-efficiency. This paper proposes a mathematical model to resolve this dilemma, providing a framework for determining the optimal inter-upgrade interval for enterprise BI systems

2. Literature Review

The extant literature regarding the optimal temporal intervals for enterprise software upgrades remains limited. Existing research primarily focuses on user perceptions, pricing strategies, or qualitative organizational motives, leaving a significant gap in quantitative decision-making frameworks for lifecycle management.

Vaniea and Rashidi [8] conducted an empirical survey of 307 respondents to examine user behavior during software updates. Their findings indicate that stakeholders perform a complex cognitive balancing act, weighing the potential functional benefits of an update against the inherent risks and implementation costs. The authors identified three critical decision-making factors: (1) the transparency of information regarding the new version, (2) the volume of resources required for deployment, and (3) the availability of a robust recovery path in the event of failure. While providing a valuable psychological framework for update adoption, their study does not offer a specific mathematical interval for enterprise-wide upgrades.

From an economic perspective, Bala and Carr [1] developed mathematical models to analyze the relationship between software upgrade timing and pricing strategies. Their research aimed to determine the efficacy of special upgrade pricing in incentivizing technology adoption. They concluded that discounted pricing is most effective when a new version either introduces substantial functional improvements or represents a very minor incremental change. Conversely, for moderate upgrades, discounted pricing may not be the optimal strategic choice. However, their model focuses predominantly on consumer-facing personal software and price elasticity rather than the operational timing requirements of enterprise-level infrastructure.

Khoo and Robey [5] provided one of the few qualitative studies specifically addressing "packaged" enterprise software, such as SAP and Windows. Through a case study of a large-scale organization, they characterized software upgrades as a unique category of Information Systems (IS) projects, distinct from routine maintenance or initial system adoption. Their findings suggest that "sunset dates"—the points at which vendors cease support for legacy versions—often serve as the primary external driver for upgrade decisions. While Khoo and Robey [5] highlight the necessity of maintaining support, their

qualitative approach does not address the quantitative balance between various operational costs and technological currency. Integrating reliability into the upgrade cycle is further addressed by Lad and Kulkarni [4]. They propose a Life Cycle Cost (LCC) approach to maintenance, emphasizing that the optimal schedule must account for both fixed setup costs and the increasing variable costs of system degradation.

Despite these contributions, there remains a lack of specific guidance for organizations—particularly those in high-stakes environments like healthcare or large-scale manufacturing—on how to optimize the frequency of upgrades to maintain performance while minimizing fiscal waste.

To address this gap, this study proposes a quantitative decision model inspired by seminal research in inventory control. By adapting the principles of the Economic Order Quantity (EOQ) model, which balances setup costs against holding costs [3], we develop a framework that balances "setup" (testing and implementation costs) against the "holding" costs of technical debt and system stagnation. This approach allows decision-makers to move beyond arbitrary, policy-driven intervals toward a data-driven temporal optimization.

3. Mathematical Model

To analyze the decision-making logic behind enterprise software lifecycles, we propose a mathematical optimization model. This model aims to determine the optimal temporal interval for Business Intelligence (BI) upgrades by balancing the fixed costs of infrastructure preparation against the variable costs of system testing.

3.1. **Model assumptions and parameter definition.**

- Planning Horizon (T): The organization operates within a fixed longitudinal planning window (e.g., 10, 20, or 30 years).
- Annualized Setup Cost (K/year): This represents the fixed capital and operational expenditure required to maintain a dedicated testing environment, including server hardware (RAM, storage) and Relational Database Management System (RDBMS) licenses for data staging.
- Testing Cost (h): Software validation requires significant human capital. We assume a constant cost (h) per time period, representing the cumulative salary and opportunity costs of personnel engaged in User Acceptance Testing (UAT).
- Continuous Release Cycle: While vendors like MicroStrategy release updates periodically, the model assumes a continuous time scale to facilitate the derivation of an optimal point.
- Dual-System Architecture: The model assumes the parallel operation of a stable "Production" environment and a "UAT/Development" environment for version validation.

Decision Variable:

- t: The inter-upgrade time (the interval between consecutive software updates).

3.2. **The primary optimization model.** The Objective Function aims to minimize the Total Cost (TC) over the planning horizon. TC is the sum of the amortized setup costs and the cumulative testing costs. Therefore, the total cost (TC) for a cycle will be given by:

$$TC(t) = \frac{K \cdot T}{t} + h \cdot t \quad (\forall t \in \mathbb{R}^+)$$

Given that both terms in the summation are convex, the function $TC(t)$ is strictly convex. To find the optimal inter-upgrade interval (t^*), we take the first derivative of TC with respect to t and set it to zero:

$$\frac{dTC}{dt} = -\frac{K \cdot T}{t^2} + h = 0$$

Solving for t yields the Optimal Inter-Upgrade Time:

$$t^* = \sqrt{\frac{K \cdot T}{h}} \tag{3.1}$$

3.3. **Model calibration and policy implications.** Applying this model to PT. XYZ's current policy—a fixed three-year upgrade cycle ($t = 3$)—and assuming a 10-year planning horizon ($T = 10$), we can infer the organization's implicit valuation of its resources. Under these parameters, the ratio of setup costs (K) to testing costs (h) is approximately 0.9 (i.e., $K \approx 0.9h$). This suggests that for a three-year cycle to be mathematically optimal, the annualized cost of maintaining testing infrastructure must be nearly equal to the annual cost of human capital dedicated to testing. If the organization reduces setup costs (e.g., through cloud-based automation), the optimal interval t^* would decrease, justifying more frequent updates to leverage newer software features.

3.4. **Model extension: incorporating a stabilization period.** In many healthcare and mission-critical industrial environments, organizations require a fixed stabilization period (t_1) following an upgrade before the next testing cycle begins. In this refined scenario, we redefine t as the duration of the testing phase. The total cycle time then becomes $t + t_1$. The modified Total Cost function is:

$$TC(t) = \frac{K \cdot T}{t + t_1} + h \cdot t \quad (\forall t \in \mathbb{R}^+)$$

Maintaining the convexity of the function, we derive the Optimal Testing Duration:

$$t^* = \sqrt{\frac{K \cdot T}{h}} - t_1 \tag{3.2}$$

Example Application: Using the previous assumptions ($T = 10$, $K = 0.9h$) and introducing a mandatory six-month stabilization period ($t_1 = 0.5$ years), the optimal active testing duration is calculated as 2.5 years. This brings the total inter-upgrade interval back to 3 years ($2.5 + 0.5$), demonstrating the mathematical consistency of the model with organizational constraints.

4. Sensitivity Analysis

To further examine the robustness of the proposed model and its practical utility for organizational decision-making, we conducted a sensitivity analysis. This analysis evaluates how variations in the cost ratio (K/h), planning horizons (T), and stabilization periods (t_1) influence the optimal inter-upgrade interval (t^*).

4.1. **Impact of cost ratios (K/h) and planning horizons (T).** We evaluated the ratio of setup costs to testing costs (K/h) across a range of 0.10 to 2.50, considering three distinct planning horizons: 10, 15, and 20 years. The results, detailed in Table 1 and Figure 1, reveal a direct correlation between the cost ratio and the optimal upgrade interval.

Table 1. Various values of $\frac{K}{h}$ and T and their corresponding t^*

$\frac{K}{h}$	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4
$T = 10$	1.41	2.00	2.45	2.83	3.16	3.46	3.74	4.00	4.24	4.47	4.69	4.90
$T = 15$	1.73	2.45	3.00	3.46	3.87	4.24	4.58	4.90	5.20	5.48	5.74	6.00
$T = 20$	2.00	2.83	3.46	4.00	4.47	4.90	5.29	5.66	6.00	6.32	6.63	6.93

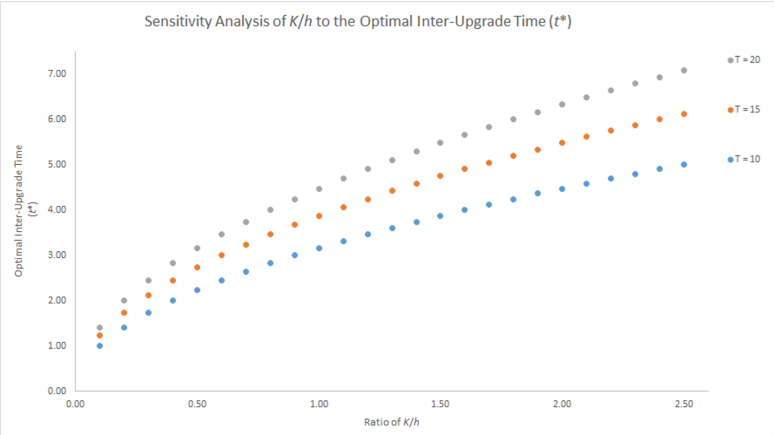

FIGURE 1. Sensitivity Analysis of various values of $\frac{K}{h}$ and T and their corresponding t^*

- Fixed Horizon Effects: As the setup cost (K) increases relative to the testing cost (h), the optimal interval (t^*) lengthens to amortize the high initial investment over a broader temporal range.
- Non-Linear Temporal Impact: The planning horizon (T) exerts a non-linear effect on t^* due to its position within the square-root function of the primary model (3.1). For instance, at a constant K/h ratio of 1, increasing the planning horizon from 10 to 20 years extends the optimal upgrade interval from 3.16 years to 4.47 years.

4.2. **Influence of the stabilization period (t_1).** In many high-stakes environments, such as healthcare or industrial manufacturing, a mandatory stabilization period (t_1) is required post-upgrade to ensure system reliability before the next testing cycle commences. We conducted a comparative sensitivity analysis for t_1 values of 0.5, 1.0, and 1.5 years, utilizing a fixed 15-year planning horizon ($T = 15$).

TABLE 2. Various values of $\frac{K}{h}$ and t_1 (for $T = 15$), and their corresponding t^*

$\frac{K}{h}$	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2.0
$T = 15, t_1 = 0.5$	1.23	1.95	2.50	2.96	3.37	3.74	4.08	4.40	4.70	4.98
$T = 15, t_1 = 1.0$	0.73	1.45	2.00	2.46	2.87	3.24	3.58	3.90	4.20	4.48
$T = 15, t_1 = 1.5$	0.23	0.95	1.50	1.96	2.37	2.74	3.08	3.40	3.70	3.98

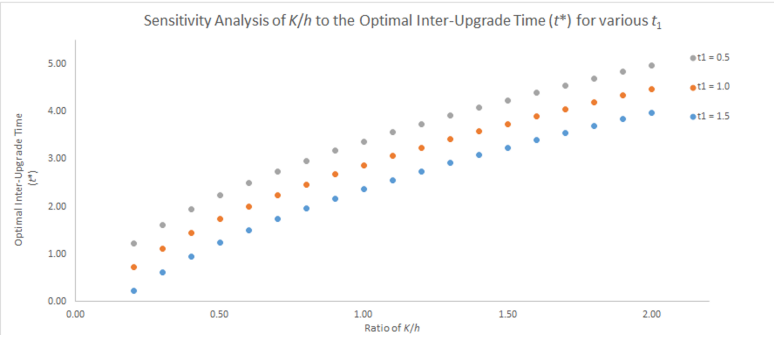

FIGURE 2. Sensitivity Analysis of various values of $\frac{K}{h}$ and t_1 (for $T = 15$), and their corresponding t^*

- Linear Downward Shift: Unlike the non-linear impact of T, the introduction of a stabilization period results in a linear downward shift of the optimal testing duration (t^*), as demonstrated by the relationship between Equations (3.1) and (3.2).
- Active Testing Contraction: As the stabilization period t_1 increases, the time available for active software testing within a cost-optimized cycle decreases proportionally. For a K/h ratio of 1 and $T = 15$, increasing t_1 from 0.5 to 1.5 years reduces the optimal active testing time from 3.37 years to 2.37 years.

4.3. **Strategic decision-making implications.** These findings offer critical insights for executives managing enterprise BI systems or Healthcare Information Systems (HIS). To achieve more frequent upgrades and leverage the latest technological features (such as responsive "Free-form Layouts"), organizations must focus on reducing the setup cost (K). In a healthcare context, automating the deployment of testing environments or utilizing cloud-based staging areas can effectively lower K, thereby reducing the optimal inter-upgrade interval. This allows for the rapid integration of advanced clinical decision-support tools without incurring the prohibitive costs traditionally associated with legacy three-year cycles. By calibrating their upgrade policies according to these mathematical parameters, decision-makers can ensure that technological renewal is driven by cost-efficiency rather than arbitrary institutional mandates.

5. Conclusion

Effective data visualization is no longer a luxury but a fundamental requirement for strategic decision-making in complex organizational environments. As enterprises pivot toward mobile-first BI architectures to support the mobility of executive leadership, the limitations of legacy software versions—such as the reliance on "Report Service Documents" over modern "Dossiers"—become a significant barrier to operational efficiency. This research has demonstrated that relying on arbitrary, policy-based upgrade cycles (e.g., the three-year mandate at PT. XYZ) can lead to significant missed opportunities for technological optimization. Our proposed mathematical model offers a rigorous, data-driven alternative to conventional policy-making. By applying the formula $t^* = \sqrt{KT/h}$, decision-makers can identify the precise temporal point where the cost of maintaining obsolete infrastructure outweighs the investment required for implementation and User Acceptance Testing (UAT). Broader Applicability to Healthcare While the empirical basis of this study is situated within the automotive industry, the model's implications for the healthcare sector are profound. In healthcare informatics, where the timely delivery of clinical data can impact patient outcomes, the decision to upgrade Hospital Information Systems (HIS) or clinical dashboards must be calculated with precision. The high cost of clinical staff participation in testing (variable cost h) must be balanced against the necessity of maintaining state-of-the-art diagnostic and analytical tools (setup cost K). In conclusion, shifting from a policy-driven to a model-driven upgrade strategy allows organizations—whether industrial or clinical—to maintain technological currency while minimizing fiscal waste. Future research should explore the inclusion of "risk-of-failure" variables within the model to further refine the optimal interval for mission-critical healthcare environments.

Statements and Declarations

The authors declare that they have no conflict of interest.

References

[1] R. Bala and S. Carr. Pricing software upgrades: The role of product improvement and user costs. *Production and Operations Management*, 18(5):560–580, 2009.

[2] R. C. Basole. Enterprise mobility: Researching a new paradigm. *Information Knowledge Systems Management*, 7(1-2):1–7, 2008.

[3] D. Erlenkotter. Ford whitman harris's economical lot size model. *International Journal of Production Economics*, 155:12–15, 2014.

[4] B. K. Lad and M. Kulkarni. Integrated reliability and optimal maintenance schedule design: a life cycle cost based approach. *International Journal of Product Lifecycle Management*, 3(1):78–90, 2008.

[5] H. Min Khoo and D. Robey. Deciding to upgrade packaged software: a comparative case study of motives, contingencies and dependencies. *European Journal of Information Systems*, 16(5):555–567, 2007.

[6] A. A. Ozok, D. Benson, J. Chakraborty, and A. F. Norcio. A comparative study between tablet and laptop pcs: User satisfaction and preferences. *International Journal of Human-Computer Interaction*, 24(3):329–352, 2008.

[7] M. Petrini and M. Pozzebon. Managing sustainability with the support of business intelligence: Integrating socio-environmental indicators and organisational context. *The Journal of Strategic Information Systems*, 18(4):178–191, 2009.

[8] K. Vaniea and Y. Rashidi. Tales of software updates: The process of updating software. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3215–3226, San Jose, USA, 2016. Association for Computing Machinery, New York.