



SaaS vs. Traditional Software: A Comparative Analysis

Goutham Sabbani

MSc FinTech (UK)

ABSTRACT

The origin of Software as a Service has notably impacted several factors in human life and also business operations. This is evident by the 73% increase in global SaaS spending over the past five years. SaaS emerged in the 1990s as a cost-effective, scalable, and flexible alternative solution. Historically, traditional software has dominated the world, requiring upfront infrastructure investments and long-term maintenance [3].

This paper will discuss a comparative analysis of traditional software and SaaS, mainly focusing on their critical features like cost-efficiency, deployment speed, scalability, security, and user accessibility. We will examine how these variances impact the business, operational efficiencies, and user experience.

This study will focus on a comprehensive understanding of the advantages and barriers of comparing both software, offering ideas to help individuals navigate the evolving software landscape effectively.

Key words: Software as a Service (SaaS), Traditional Software, Cost Efficiency, Scalability

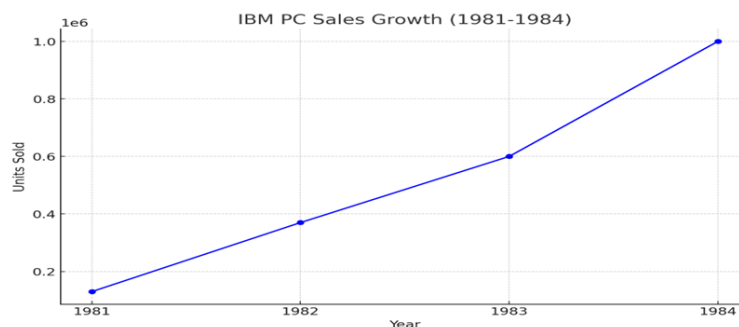
INTRODUCTION

The advent of traditional software systems took place in the 1950s, which are machine code or assembly languages. The higher languages then were Fortran in 1957 and COBOL in 1959, making significant key software developments that were more accessible and efficient. The origin of this software laid the groundwork for more sophisticated and user-friendly software. In the late 1980s, the personal computer was invented. Companies like IBM and Apple introduced personal computers (PCs), which gave power to smaller companies and individuals [4].

After this, the SaaS (Software as a Service) model took place in the 1990s and 2000, transforming how software was delivered and consumed. SaaS provides the facility to access data over the internet, unlike traditional software, which requires to install upfront. The high-speed internet available has driven the growth of the SaaS model [2].

Software engineering has been profoundly impacted by the rise in the SaaS model due to higher programming languages and operating systems in the early days.

For example, a significant company, IBM, shows exponential sales growth, depicting the importance of SaaS. Here is the bar chart for the same.



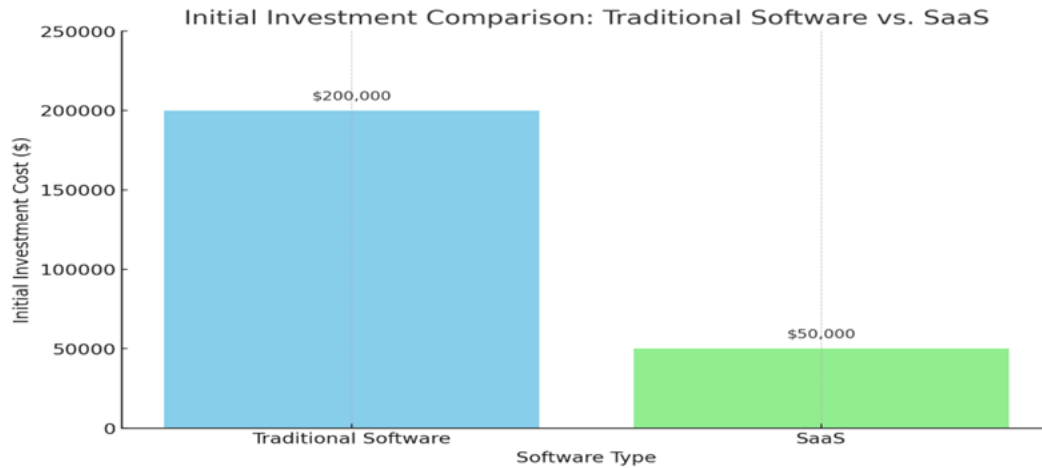
Source: IBM archives [5]

COST EFFICIENCY OF TRADITIONAL SOFTWARE V/S SAAS

Traditional software has ample upfront costs because running this software includes purchasing a software license, hardware infrastructure, and often substantial investment in implementation service. It costs companies thousands of dollars to buy enterprise software and the necessary servers to run it.

In contrast to traditional software, the upfront costs are comparatively low. SaaS is typically subscription-based, meaning businesses pay a monthly or annual fee to use the SaaS software. This model perfectly eliminates the significant upfront investment for software licenses and hardware. This spreads their investments over time, making it more accessible for companies to manage their budget and decrease initial capital expenditure [6].

Here is a bar chart comparing investment in traditional software and SaaS



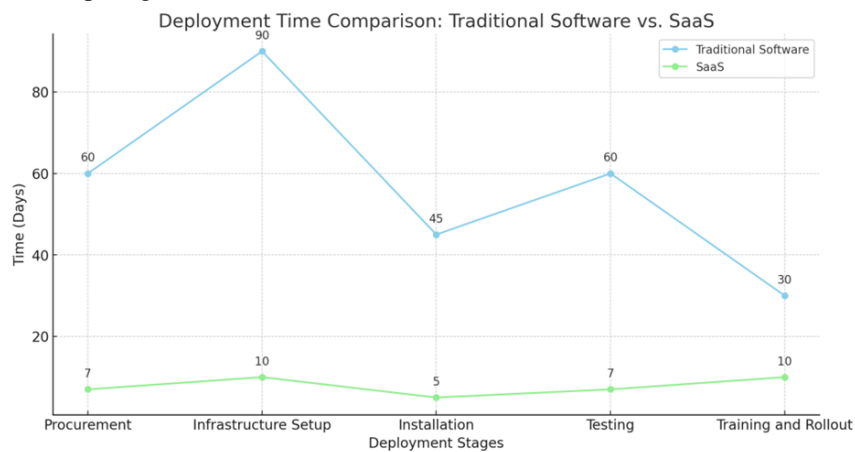
Source: SaaS casestudies [5]

DEPLOYMENT OF SAAS AND TRADITIONAL SOFTWARE

The traditional deployment will take a lot of time because it has several stages. First is the procurement phase, which involves buying the necessary software licenses and hardware infrastructure. This depends on the vendor they are seeking. This can go up to weeks or months, depending upon the procurement stages. Traditional software's next phase is the infrasound setup phase, which involves installing and configuring servers, databases, and network equipment. Traditional software requires compatibility and optimal performance, which requires specialized skills. After this installation phase, the software and existing company servers are installed on company or organization servers.

Compared to traditional software, SaaS is significantly faster and easier to deploy. Firstly, it starts with creating an account on the SaaS application that can often be completed within a few hours or days. The next step is configuring the SaaS model with previous software according to business software. SaaS usually has a user-friendly interface and predefined templates, making configuration faster and easier. Applications with existing systems and data sources, facilitated by API and pre-built connectors, streamline the process.

Here is a line chart comparing both timelines



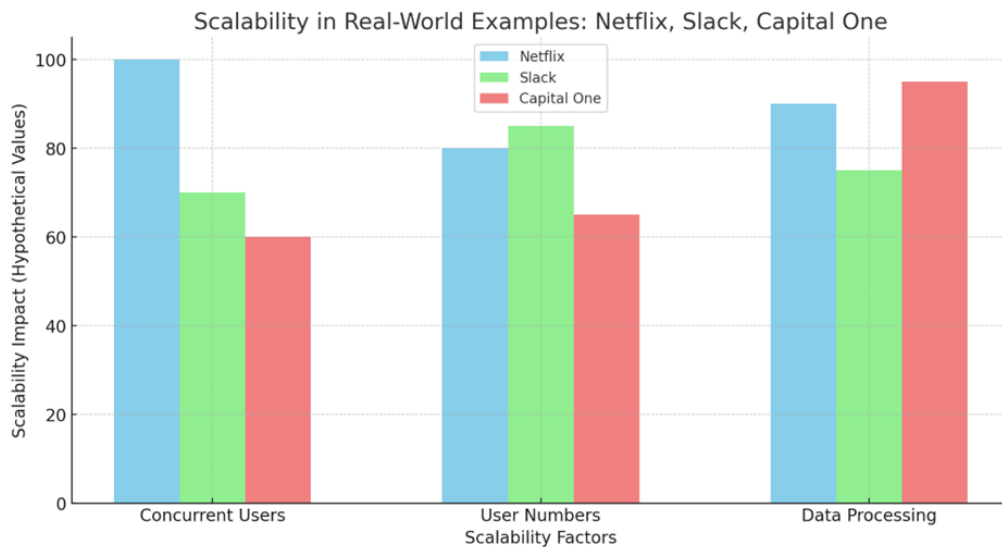
Source: Salesforce customer services [7]

SCALABILITY CONCERNS OF TRADITIONAL SOFTWARE AND SAAS

Scalability is a huge concern in traditional software when it comes to scalability. One of the main issues is hardware limitations because this requires substantial investments in additional servers, storage, and infrastructure components. Setting up traditional software can be time-consuming, especially when physical expansion is necessary. The resource is another barrier, as efficiently managing and allocating resources in traditional environments requires skilled IT personnel.

SaaS overcomes all these problems by giving numerous advantages in terms of scalability. One of the main benefits is elastic scalability. SaaS can leverage its technology and quickly scale up or down according to business requirements, driven by cloud infrastructure that allows for dynamic resource allocation. This results in optimal performance without the need for physical hardware upgrades. The SaaS models are mostly subscription models that can effectively manage costs, and they are predictable, too, with businesses only paying for the resources they use.

Many companies like AWS and Netflix have adopted this SAAS to leverage the scalability of this software model. This underscores the significant benefit of the SaaS model. Here is a graph showing the same



Source: Slack Engineering Blog [8]

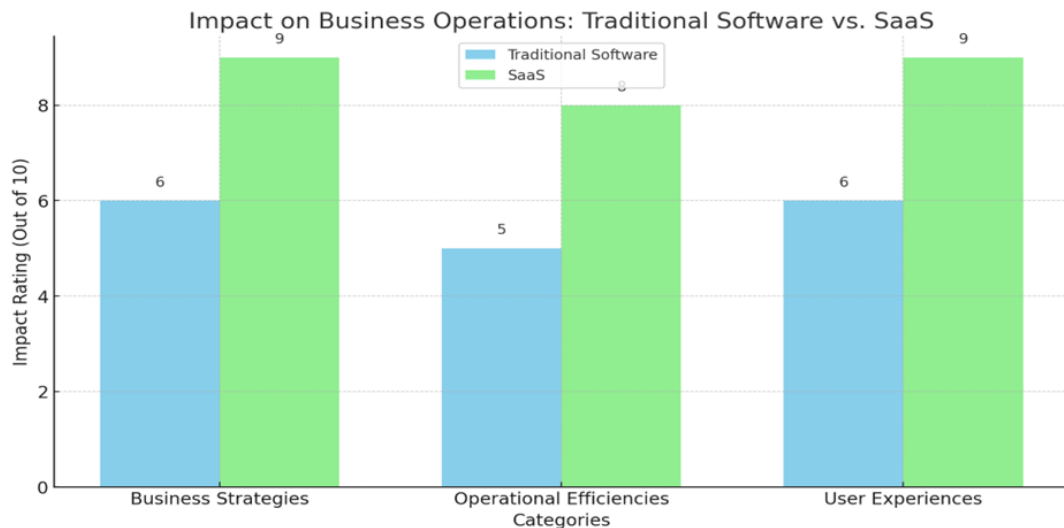
IMPACT ON BUSINESS OPERATIONS

Companies that adopted SaaS have significantly impacted their business by changing fundamental approaches to their operations and growth. SaaS enables us to be more agile and responsive to market changes. By using cloud-based applications, companies can key deploy new features according to the market ahead of the competition's scalability, enabling businesses to expand their operations without significant investment quickly. SaaS's scalability feature helps companies enter new markets and scale up operations during peak stages or promptly change according to customers' demand [9].

SaaS develops operational efficiencies by streamlining processes and decreasing the burden of the IT department. Since the SaaS includes maintenance, updates, and security, internal IT teams can focus more on strategic tasks than routine marinate. This ideally improves efficiency and rescues the operational costs associated with traditional software infrastructure maintenance costs.

SaaS, in terms of user experience, is an essential factor providers prioritize when delivering intuitive, user-friendly interfaces that require minimal training. SaaS focus on usability improves user satisfaction and productivity, as users can quickly implement this SaaS technology. The user of SaaS is constantly updated with new updates due to the cloud-based nature of SaaS. Feedback platforms are part of SaaS applications that users can directly access without the intervention of any platform.

Here's a graph comparing business, operational efficiencies, user experience



Source: Operational efficiencies SaaS [1]

BENEFITS OF SAAS AND TRADITIONAL SOFTWARE

Table 1: Benefits of Saas and Traditional Software

Benefits	SaaS	Traditional Software
Cost Efficiency	Subscription-based pricing eliminates high upfront costs	Large upfront capital investment
Scalability	Easily scalable based on demand	Scaling requires significant infrastructure investment
Accessibility	Accessible from anywhere with an internet connection	Limited to local network access
Automatic Updates	Automatic updates and maintenance by the provider	Manual updates and maintenance required
Quick Deployment	Faster deployment compared to traditional software	Longer deployment time
Integration	Easier integration with existing systems and applications	Complex integration with existing systems
Customization	Limited customization options	Highly customizable to specific business needs
Control	Less control over data and security measures	Complete control over software and data
Performance	Dependent on internet connectivity	Better performance for resource-intensive applications
One-Time Cost	Ongoing subscription fees	One-time purchase cost
Data Security	It relies on the provider’s security measures	Custom security protocols can be implemented

Source: advantages of SaaS [8]

BOTTOM LINE

Traditional software and SaaS each provide their advantages and have barriers. SaaS excelled in cost efficiency, scalability, and accessibility, making it an attractive option for modern businesses seeking agility and decreased IT burdens. SaaS also comes with a subscription model and cloud-based nature to provide flexibility and ease of use, though it depends heavily on internet connectivity and includes security measures [10].

In contrast, traditional software provides customization, control, and potentially better performance for resource-intensive applications. However, traditional software requires in-house IT teams to make massive investments in the first place and maintenance.

This comparative analysis highlights the evolving software landscape, encouraging businesses to navigate their software choices effectively to maximize operational efficiency and competitive advantage.

REFERENCES

- [1]. Amazon Web Services. (2017). Netflix Case Study. Retrieved from <https://aws.amazon.com/solutions/case-studies/netflix/>
- [2]. Amazon Web Services. (2018). Capital One Case Study. Retrieved from <https://aws.amazon.com/solutions/case-studies/capital-one/>
- [3]. Gartner. (2015). Gartner Research. Retrieved from <https://www.gartner.com/>
- [4]. IBM Archives. (2006). Retrieved from <https://www.ibm.com/ibm/history/>
- [5]. Salesforce. (2015). Salesforce Customer Stories. Retrieved from <https://www.salesforce.com/case-studies/>
- [6]. Slack. (2015). Slack Engineering Blog. Retrieved from <https://slack.engineering/>
- [7]. Forrester Research. (2007). Forrester Research. Retrieved from <https://www.forrester.com/>
- [8]. IDC. (2015). IDC Research. Retrieved from <https://www.idc.com/>
- [9]. Microsoft. (2014). Microsoft Customer Stories. Retrieved from <https://customers.microsoft.com/>
- [10]. Google Cloud. (2017). Google Cloud Case Studies. Retrieved from <https://cloud.google.com/customers>