

Perpetual versus saas sales licensing models essay sample



**ASSIGN
BUSTER**

Increased competitiveness in markets is putting pressure on organisations to improve the productivity of their information technology by either reducing IT expenditure or getting more utility out of it (Robinson 2006: 34). To this end organisations have been exploring and taking advantage of the alternatives available to procure their information technology (IT) software. Software procurement consists of transactions where the software author grants a software user permission to use the software under predetermined terms and conditions. The terms and conditions are presented in the form of a software license. (Gangadharan & D'Andrea, 2011: 48).

Ferrant (2006: 24) define software licensing as “ any procedure that lets an enterprise or user purchase install, and use software on a machine or network in accordance with the software vendor’s licensing agreement.”

Software licenses can either be purchased or rented as a service from the owner or provider of the software (Gangadharan & D'Andrea, 2011: 48). Purchased licenses confer property rights of a copy of the software to the purchaser, thus allowing unlimited usage of the software (perpetual license). Perpetual license holders are responsible for installing software onto their hardware and the subsequent update and maintenance of the software (Choudhary 2007: 142). Service based licenses (also known as Software as a service, SaaS), on the other hand, provide the purchaser rights of use of the software for a limited period in time. The users of service license access the software via the internet from a vendor or service provider who runs and maintains the software (Choudhary, 2007: 142).

The SaaS licensing model is fast becoming the IT procurement route of choice and as such the importance of gaining an understanding of this model cannot be understated. The reason for preparing this report is to enlighten readers on what SaaS is, how it differs from perpetual licensing, the benefits and the pitfalls one should be aware of when considering using it. The first part of the report will provide a description of SaaS, in order for the reader grasp this concept by providing an overview of SaaS, its origins, trends, characteristics, lifecycle process and software development process. The next section will primarily focus on some of the software licenses available under the perpetual and SaaS licensing models. As an introduction to this second part, reasons for license use will be given in order to draw the readers' attention to importance of compliance and the implications of non-compliance with the terms and conditions of a software license. The third part of the report addresses the benefits and pitfalls of using SaaS in comparison to perpetual licensing models followed by recommendations and the conclusion.

Software as a service

1 Overview

Cloud technology has enabled the formation of business models that exist on the internet offering software applications as a service (SaaS) with the independent service vendors owning the software licenses (Baltzan & Phillips, 2010: 459). SaaS is a software delivery model (Laplante, et al., 2008: 46) which delivers software to a client in a flexible and cost effective way (Matthew & Nair, 2010: 53). It delivers measured usage of software to users,

from one or more owners or providers who are in a remote location, on a pay as you go or subscription basis (Gartner, n. d), through a service provider (Kaplan, 2007: 48). The service provider acts as a mediator between the software developer or author and the end user (Matthew & Nair, 2010: 53). The service provider's responsibility is to deliver, manage and secure software applications, data and the underlying infrastructure (Kaplan, 2007: 48). Therefore the collaborating actors in SaaS are:

- Independent software vendor (ISV) – This is the software manufacturer, or developer or owner of the software
- Business service provider (BSP) – They add value to the software application offerings by packaging and hosting the software of the ISV on their servers thus facilitating users to access the software applications.
- Client – The client pays for services based on payment arrangements made with the BSP, to use the applications that are hosted on the BSP's servers on a subscription or pay as you go basis. (Matthew & Nair, 2010: 53).

In order to commercialise SaaS the BSP requires three technologies which are:

- Platform as a service (PaaS) – It provides the environment to host the service and support to the service provider by monitoring and auditing users access and utility.
- Licensing and the rights to manage the software applications
- Billing. (Konary, 2012: 2).

2 Background

SaaS originates from the now defunct application service provider (ASP) model which in the 1990's provided customised applications services to their clients via the internet. Providing customised services, under the ASP model,

made achieving a scalable and reliable service prohibitively expensive. ASP also failed to attract sufficient customers. The reason was ASP's used perpetual licensing model to provide software services which were already being sold as products by established software providers. Customers preferred to purchase product applications from established providers rather than purchase the same application services from an unknown ASP vendor, whose longevity could not be established in the dot. com era. (Kaplan, 2007: 48).

3 Trend

The SaaS model appears to be a successful alternative to the perpetual licensing model as SaaS market revenue reached \$16. 6 billion in 2010, which equates to an annual growth rate of 31. 4%. It is anticipated that SaaS software delivery will outpace the traditional software product five times. The expectation is 43% of revenue from all software sales in 2013 will come from SaaS transactions due to new adopters and existing users expanding their SaaS software requirements. (Konary, 2012: 1-2).

4 Characteristics of SaaS

i. Multiple tenants are supported by SaaS applications, which are deemed a key requirement of SaaS. Thus all SaaS applications must be multi-tenant aware, which means SaaS providers are able to allow more than one customer (tenant) to use a single application concurrently without affecting how the other customer uses it. (Espadas, et al., 2013: 274). ii. Each SaaS vendor offers standardised software applications. Multi tenancy ensures that any changes a vendor makes to software features or function, based on a <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

single customer's feedback, is transferred to all the other customers of that vendor (Gangadharan & D' Andrea, 2011: 50). iii. SaaS applications allows network-accessible operations which are accessible anywhere at any time (Gangadharan & D' Andrea, 2011: 50). iv. Software programmes are executed or run on infrastructure that is hosted by the software service provider; all the customer sees are the results of the execution (Gangadharan & D'Andrea, 2011: 50). v. Software for services are designed to enable repeated use (Gangadharan & D' Andrea, 2011: 50). vi. SaaS encourages service orientation (Gangadharan & D'Andrea, 2011: 50). Service orientation architecture (SOA) model uses individual and reusable software applications, with specific functionality, as building blocks in the creation of a software system. The software system resulting from SOA creation serves to fulfil a business function. Therefore numerous SaaS vendors can be the providers of application services of a single software application. (Laplante, et al., 2008: 47).

5 SaaS product life cycle

Fig 1. SaaS Lifecycle taken from Jiang, et al., 2009: 679

Software product under the SaaS model goes through the stages shown in fig 1. The stages of requirements analysis and development and testing are based on existing processes. SaaS lifecycles in reality may differ from the general model depicted in fig 1 at the deployment stage. Certain types of services applications that are developed on the web do not need to explicitly go through a service deployment stage whereas in contrast other software services that are very popular and have high levels of customer interest may need additional deployments in order to respond in a manner that will satisfy <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

customers. The software developer defines the business terms of the service application for example is the service going to be hosted in a cloud or is a business partner going to resell the product (see section 2. 5 Software development). The billing policies for the software are also packaged as part of the business terms. Customers after this stage can subscribe for the software service. The vendor accepts the subscription and authorises the customer to access the service on a metered usage basis. The periodic bills for payment are generated from the collected metered information. In order to improve customer service provision the customer feedback stage is critical as it is from this stage the software vendor is able to provide automatic software upgrades and improved features. (Jiang, et al., 2009: 679).

6 SaaS software development

Traditionally software development occurred under some form of employment contract, where the author or authors of software were employed by a company using an employment contract. This contract transfers certain rights to the employer resulting in the employer holding the intellectual rights of the software. The employer then uses these rights to license the use of the software through selling copies of the software. (Lindeman, et al., 201: 31; Gangadharan & D'Andrea, 2011: 48). More current methods use a combination of software applications to achieve the required IT functionality the businesses requires. Therefore a business creates a shopping list of functionalities that it requires for its business system and shops around on the SaaS market for service providers who provides these software services. Once located the business selects <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

applications that will serve the required business functions from the vendor's offerings. The result will be a single business functional system which consists of a combination of small pieces of self contained software functionalities from different vendors which are accessed via the internet through the service provider. (Gangadharan & D'Andrea, 2011: 48).

The open source model is one of the modern ways of combining the various fragments of software application services. It involves collaboration between a vendor's internal employees with the software vendor's competitors, clients, consultants and subcontractor's. The vendor company seeking to undertake software development incentivises the community (that is, those not directly employed the company seeking software development) to participate in software development by using agreements, sales contracts and more predominantly software licenses. When a software license is used in software development, the SaaS vendor (company seeking to develop the software), those who participated in the open source collaboration (excluding the vendor's employees as they are covered by employment contracts) and the users are all parties to the license. Therefore as a software user it is imperative to understand the involvement of others in the software development and the rights inferred to them under your software license in order to protect one from breaching the terms of the license. (Lindman, et al., 2011: 32).

Although other methods exist to incentivise software development by protecting of intellectual property of the authors, this report will be focusing on software licensing which will be described in the next section of this report.

<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

Software licensing

1 Why software licensing?

Since the 1970's, software has been considered to be literary work legally protected by copyright law (Gangadharan & D'Andrea, 2011: 49). Software licensing's function is to facilitate technology transfer from vendors to users in exchange for financial benefits (Gangadharan & D'Andrea, 2011: 48), whilst legal ownership of the software is retained by the vendor. Lindman, et al., (2011: 31) add that a license defines user's limitation of use. Limitations of use are defined in the licensing agreement which are enforced through mechanisms that are built in the software and by tracking and managing the licenses whilst they are in use (Ferrante 2006: 25; Zsigri 2008: 6). In the case of the open source software development model the license serves as a means of allocating rights of the various fragments of applications that make up a software application (Lindman, et al., 2011: 31).

The technical function of licensing seeks to reduce exposure, of both the vendor and the user, to the risks and consequences of piracy of both the vendor and user of the software. The vendor or software author needs to be protected against "hard piracy" whereas the user seeks protection from auditing fines from "soft piracy". Hard piracy is purposeful violation of the software licensing terms and conditions in order to bypass the licensing protection system in order to illegally distribute the software. The result of hard piracy is lost revenue for the vendor. Soft piracy is when users unintentionally uses the software in a manner which violates the terms and

conditions of the license. If this violation is identified the user faces a fine. (Ferrante, 2006: 24-25).

2 Perpetual and SaaS licensing models

Perpetual licenses allow upfront and permanent software purchasing (traditionally purchased as out- of-the-box software) which allows unlimited usage which does not expire (Rohitratana & Altmann 2012: 1328). SaaS based licenses; on the other hand, provide the purchaser rights of use of the software for a limited time (Stephan 2009: 278). The users of SaaS licenses access the software via the internet from a service provider who runs and maintains the software (Choudhary 2007: 142).

The two categories of software licenses are;

1. Static licenses

Licenses under this banner are tied to systems or users and do not allow short notice redistribution or sharing with unlicensed users or systems.

Reassignment is possible after a stipulated time period on some static licenses however those licenses that are firmly assigned to a user or client can only be transferred as stipulated by the manufacturer.

2. Dynamic licenses

Reallocation of dynamic licenses is flexible and can occur at any time the need arises. (Gull & Wehrmann, 2009: 280).

3 Perpetual Licensing models

In perpetual licensing models the software is installed on an organisations server and the licenses allow distribution within the organisation (Dalheimer <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

& Pfreundt, 2009: 133). The typical perpetual license agreement forces customers to purchase technical support, patches and software updates for stated periods of time (Ferrante, 2006: 26). Perpetual licensing models utilise the following mechanisms to deter license breach.

1 Node-locked mechanisms

The node-locking mechanisms are built into software applications to ensure that licensing agreement terms and conditions are not breached by allowing the use of the software applications on a single machine. There are different node-locking methods and the type a vendor uses is dependent on the level of protection that is required. These types of software licensing mechanism (particularly the weaker methods) do not offer users ease of use which results in inadvertent misuse of the software application. The different mode locking methods are as follows: i) Machine host name or named user – When software is purchased it is assigned to a specific end user or machine and provides unlimited usage (Bontis & Chung, 2000: 247). It is considered to offer the weakest software protection as it permits users to change the machine hosts name at any time, thus inviting misuse by users (Ferrante, 2006: 25). ii) Ethernet MAC (media access control) address – This licensing method assigns the software license to the media access control (MAC) address (Juniper 2012), also known as the physical address of the hardware device of the user.

Physical or MAC addresses are used by the ethernet to locate most devices worldwide (Autocad, 2012). A MAC address is unique to network devices worldwide and consists of a 12-digit hexadecimal number written in the

following format MM: MM: MM: SS: SS: SS or MM-MM-MM-SS-SS-SS. The M represents the manufacturer identification code and the S identifies a particular device. The weakness in this method is it is possible to change Mac address thus making it possible to reassign a license to another machine (Rusu 2011). iii) Product key or serial number (Ferrante, 2006: 25) – This is a “ unique, alphanumeric code of any length required by many software programs during installation” (Fisher 2012). This piracy deterrent method is not completely safe as cracking a serial number is considered difficult but possible through the creation of keygens and also serial numbers can be leaked. (Wong 2012). iv) Use hardware devices – Hardware against Software Piracy (HASP) dongle is a piece of hardware (a dongle is an usb device) used to activate the software application by inserting it into a machine’s parallel or USB port (Grimmick, n. d). This method is considered to be the most effective (Ferrante, 2006: 25).

This is because the hardware carries some calculations that participate in the running of the application and those seeking to breach the license have to identify what role the hardware key plays in the software programme. A programme that initiates self destruction can be incorporated on the device if tampering is detected. (Zhang 2010: 6). Although it is very effective in licensing protection it causes an inconvenience for the user who has to purchase a separate hardware device after purchasing and installing an application online (Wong 2012: 4).

2 Network license

Network licenses are typically used in large enterprises where the use of the software application is required by many users in the business. The key benefits of using a network license for big organisations are • Users of the licenses are not named thus allowing the company to reassign use of the license to new employees when others leave. • Monitoring network licenses compliance occurs through server applications thus negating the need for the difficult task of monitoring and maintaining software on each individual machine within the organisation. • Automated retrieval of licenses can be achieved thus facilitating less invasive internal software audits. (Ferrante 2006: 26). The two main network licenses are concurrent (or floating) and network named. i) Concurrent licenses- This license allows software to be run on the server and permits a specified number of users to simultaneously use the software. For example if the license is a five user concurrent license only five users can access the software, if a sixth person attempts to access the program it will not allow it until one of the five users exits the application (PC mag n. d).

As a result businesses using concurrent licenses purchase licenses on the basis of peak use as opposed to average user numbers. The result is businesses fail to achieve maximum value for the investment (Bontis & Chung 2000: 248). The key benefit of concurrent licensing for enterprises is the flexibility that it provides (any user can access software from a pool of software applications) (Ferrante 2006: 27). ii) Network named- In this case the license is allocated to specific users in the organisation and as such it does not permit sharing. Although it is more restrictive than the concurrent license it is less expensive to purchase. Running this license may become

costly because constant communication between the client server and the vendor server is required for license management. If a client server has a large number of users the flow of communication traffic between client server and users may cause large traffic overheads. Some alleviation is provided by allowing users to go offline from the client server (have periods where license monitoring does not occur) for a time period.(Ferrante 2006).

3 Processor licenses

This licensing is based on the number of processors the machine on which the software is going to run has. A machine with four processors will require a four processor license (Epping 2009). Processor licenses have been criticised for penalising customers for having high powered machines, furthermore the license does not take into consideration the user value derived from the license (Bontis & Chung 2000: 247).

4 SaaS based licensing models

The two ways in which licensing occurs on the cloud is by:

1. Bring Your Own license (BYOL)

The users of the cloud are allowed to bring their own software license for deployment in the cloud. 2. License Charges Included (LCI) – The cloud has software installed and the user pays for both the software license and the machines on which it is run. This license is more expensive than the BYOL. (Konary, 2012: 2)

Although traditional licensing models, used in perpetual sales model, are deemed to be inflexible in facilitating delivery of SaaS sales (Zsigiri, 2008: 6),

modified and unmodified versions are still used on the cloud (Mohan Murthy, et al., 2012: 647). There is no universally accepted SaaS licensing framework (Gangadharan & D'Andrea 2011: 50) however the most commonly used are subscription and utility or pay per use licensing models (Ferante, 2006: 27, Mohan Murthy, et al., 2012: 650). These two models are reliant on license use reporting which generates information which the vendor uses for billing the customer or user (Ferante 2006: 28). In practice numerous SaaS license models exist (Mohan Murthy, et al., 2012: 645) however academia and the software industry have failed to develop a SaaS licensing framework (Gangadharan & D'Andrea 2011: 50). The reason, according to Gangadharan and D'Andrea (2011: 50), is that the software service community has not provided sufficient details on information that describes the business models and legal relationships involved in SaaS licensing. They add that, the non-standard technology that service orientated computing uses is responsible for the lack of detail.

Vendors pricing SaaS licensing must ensure that their price structure is aligned with the customer realisation (Bontis and Chung 2000: 247) in order to get continued business.

5 SaaS sales licensing models

SaaS sales licensing models increase cost efficiencies for businesses as they do not require upfront investment for IT expenditure or ongoing maintenance and support costs but allow IT use, maintenance and upgrades on a subscription basis or pay as you use basis (Creese 2010). Gonslaves (2009) contends that cost efficiency benefits are not always achieved, a position

which is supported by Yang, et al., (2010: 94). The results of their study reflected that some SaaS vendors include hidden costs and charges that end up costing more than the perpetual licensing model. Gonslaves (2009) states that businesses are expected to commit to a predetermined contract that is not based on actual utility. Yang, et al., (2010: 94) identified the additional charges that are levied include:

- o to configure the software

- o training

- o implement databases
- o test and run software (\$18 to \$25 per month per user (Hoffman 2006 cited in Yang, et al., 2010: 94).
- o provide support for handheld devices (\$45 per month per user(Hoffman 2006 cited in Yang, et al., 2010: 94)).
- o costs to integrate SaaS application with existing software (costs can be substantial).

1 Subscription

The customer purchases a license that gives rights of use of the software application for a fixed period, for example a year or more (Ferrante, 2006: 27). During the subscription period the software user usually has unlimited access to the software (Rohitratana & Altmann, 2012: 1328) whilst enjoys the benefits of any software upgrades or feature improvements (Ferrante 2006: 27).

The function of subscription licenses is to make the application service offerings concrete by defining service level provision and customer specific billing policy limits. The vendor authorises the user to enjoy use of a software application (service instance) which is based on service elements that are included in the subscription. The billing policy is based on measures of the <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

number of service elements (fragments of software application offerings) used to form the software application and the constraints (for example the maximum number of users of the application) described in the subscription. (Jiang, et al., 2009: 679).

License use monitoring is made transparent to both vendor and user by using a system that allows the user to view license use and payment information through the businesses web site (Ferrante, 2006: 28).

Subscription licensing models are most suitable for long term use and are used in conjunction with other licensing models (Mohan Murthy, et al., 2012: 648).

2 Pay per use licenses

Businesses determine the number of pay per use licenses they require for a particular period and make the purchase. Estimation of demand for new software will be based on the unit of “ use” defined in the license for example number of transactions, process cycles or documents generated (Gull & Wehrmann2009: 280). Billing will be based on the same unit over a specific period of time (Ferrante 2006: 28). This model best suites short term use and a small number of users as the user pays only for what they use. Again this model is also used with other models (Mohan Murthy, et al., 2012: 648).

3 Based on the Subscription to the Functionalities

This licensing option offers the user the flexibility to select individual modules or functionalities that the user requires from the software system

on offer. The service provider will only charge for those selected modules. The service provider either breaks down the application into functionalities from which the user can choose from or may offer different versions of the same software with a different pricing plan. Where a different version of software is offered (for example basic, standard or enterprise) the variation between versions could be, for example, functionalities or number of users supported (Mohan Murthy, et al., 2012: 648).

4 Free Software, Pay for Support licenses

Software vendors such as Red Hat Enterprise Linux provide the software free and charge users for support (Mohan Murthy, et al., 2012: 648).

5 Processor license – SaaS

In the SaaS model different vendors provide this license on the cloud. Microsoft offers this licensing to provide unlimited number of users access to Microsoft software on a subscription basis. The pricing is based on the number of virtual machines a user has regardless of the number of cores the VM has. This type of license is more suited for a business with many users as that is when it becomes more cost effective. (Zettagrid, n. d). In contrast IBM offers their processor license on the basis of a processor value unit. A PVU license bases its measure on the processor core on a chip (socket). Therefore this unit of measure attempts to align usage with price. For example if a processor has 4 sockets per server, and each socket has 6 cores, IBM will measure the number of cores used by the software application as opposed to charging for the whole processor. (IBM 2012).

Advantages and disadvantages of SaaS licensing model (as compared to a perpetual sales) licensing models

1 Advantages

- Multi tenancy enables the benefit of cheaper software consumption due to the economies of scale and automatic software upgrades (Gonsalves, 2009).
- SaaS do not need large upfront investment as is required on perpetual licenses thus impacting positively on cash flow (Choudhary 2007: 142). SaaS expenditure is an ongoing operating expense on the income statement whereas perpetual licensing generates assets. Assets consume cash resources that can be used elsewhere in the business to create value (Creese 2010).
- Software applications offerings from the SaaS model can be tested and adopted more quickly than perpetual models for both small and large organisations. The United States Government Accountability Office and the National Institute of Standards and Technology (NIST) findings suggest that perpetual licensing models are slow and more likely to exceed budgeted costs (52. 7%). These shortcomings result in a third of projects being cancelled (31. 1%). (Kaplan 2007: 53).
- SaaS licensing models provide a steady stream of cash flow for the independent software vendor as opposed to large periodic payments that perpetual licensing provides (Kaplan 2007: 53).
- SaaS providers automatically update software without requiring additional investment from the service user (Kaplan 2007: 53).
- The users of SaaS have increased bargaining power as they do not pay up front for the service.

If the users are not satisfied with the service they have the freedom to move
<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

to another service provider. In contrast the perpetual license users do not have such flexibility since payments are made upfront (Ju, et al., 2010: 387, Kaplan 2007: 53). • Service providers are continually under pressure to provide high quality, reliable services due to the increased bargaining power of the service users therefore high quality on demand applications can be achieved (Kaplan 2007: 53) • Existing licenses can be moved to the cloud using the concept of bring your own license. Amazon and Microsoft support license mobility (Mohan Murthy, et al., 2012: 647) • Savings on ongoing overheads used to employ and train IT staff to manage the software applications are achieved when the SaaS solution is used as the service provider provide these services and deploy the software application over the internet ready for use (Ju, et al., 2010: 387).

According to AMR research cited by Kaplan (2007: 53) deployed software under perpetual licensing maintenance and management can cost up to ten times the original license fee. • Upfront investment on expensive infrastructure on which to run the applications and subsequent unforeseen expenses due to upgrade requirements due to technological advancement no longer concern users who switch from perpetual licensing models to SaaS (Ju, et al., 2010: 387). • Businesses pay only for what they use under services applications (Ju, et al., 2010: 387). • Unplanned increases or decreases in demand of a service can be accommodated through the provision of a measurable, on demand and scalable service which is made available in a matter of hours. (Ju, et al., 2010: 387). Organisations using perpetual licensing models tend to over provide software licenses which are not utilised fully, as they lie idle during non peak periods. This results in

organisations failing to achieve maximum returns on investment due to inefficiencies and waste (Kaplan 2007: 53).

- Economies of scale can be achieved as more users sign up for the service which translates as costs savings to the SaaS user (Ju, et al., 2010: 387).
- Security is top priority for SaaS vendors and as such they employ skilled resources, network redundancies, stand-by power and up to date security and intrusion detection technologies in order to provide a superior service compared to its competitors. Such extensive investment in security measures are beyond the reach of perpetual license holders (Ju, et al., 2010: 387).
- Platform independence is achieved since the applications can be used by Windows, Linux and Mac users (Ju, et al., 2010: 387).
- Information is saved on the internet thus allowing SaaS users to access their information from anywhere using any device. Information on the internet accessible to the whole enterprise worldwide allowing better collaboration between geographically dispersed teams (Ju, et al., 2010: 387).
- Collaboration features (similar to those used on networking sites) are now part of the SaaS offerings which encourages collaboration within the business (Kaplan 2007: 49).

- Easier end user support is offered by SaaS in comparison to perpetual sales models because all user requirements are the responsibility of the service provider. Perpetual licensing on the other hand is prone to finger pointing between the network and application development teams (Kaplan 2007: 52).

- It is easier to resolve virus invasions on cloud applications because of its multi tenant architecture. Resolution of malicious attacks or patching

software is carried out at the service provider's server and cascaded out to <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

all the clients. In contrast managing repairs of perpetual sold software applications presents a challenge as different versions of the software exists on various machines, in some cases the user would have adapted the software to meet their needs which presents a further challenge for repair and support. (Kaplan 2007: 52). • It is more profitable for vendors to adopt SaaS business model as opposed to the perpetual business model (Yang, et al., 2010: 89). • SaaS vendors have a higher propensity to invest in software product development than do perpetual vendors therefore are in a position to offer software of higher quality than perpetual vendors (Choudhary, 2007: 158).

2 Disadvantages

- The risk of downtime due to loss in internet connection is a disadvantage of applications hosted on the internet. In-house hosted applications do not face these problems as their software is on a local server (Kaplan 2007: 52; Yang, et al., 2010: 94).
- Companies using SaaS have no control over the running of the applications and are at the mercy of the service provider. Therefore resolution of disruptions is based on the service provider's schedule (Kaplan 2007: 52).
- In comparison to perpetual software, SaaS is more exposed to risk from hackers and virus attacks (Kaplan 2007: 52).
- SaaS may offer a cheap upfront solution however in the long run (approximately three to four years after implementing SaaS) the total costs tend to exceed the cost of a comparable perpetual software license. However the user has to weigh the total costs of using SaaS against the total cost of ownership of the perpetual license (that is initial software costs, salaries for the IT staff, infrastructure (servers and buildings to house them, etc). (Creese 2010).

<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

- The steady pricing that SaaS solutions offers may encourage enterprises to remain with an outmoded IT business solution for longer than they should. It is advisable for companies to review their SaaS solutions at regular intervals (of maybe three to four years (Creese 2010)).
- Matthew and Nair (2010: 67) found in their study that as software functionalities become more complex the delivery time of the software increases and the ISV has to spend more resources on maintenance and support which costs more money. This translates as increased costs to the BSP and ultimately to the user.
- Security, trust and switching service provider costs are a concern when using SaaS (Choudhary 2007: 159) as the software application data is stored on the service providers site whereas these concerns do not exist in perpetual licensing as the software and data is stored on-site (Kaplan 2007: 53).
- Perpetual model based software applications are more susceptible to adaptation to allow customisation to suite the businesses requirements. SaaS based applications are not as flexible in achieving business specific customisation despite the offerings of many functionality options and combinations (Yang, et al., 2010: 94).

Recommendations and conclusion

1 Recommendations

- Software users must protect themselves from breaching the terms of the license by taking the time to understand the contents the SaaS license.
- Establish all the parties included in the license (including those involved in software development) and the rights inferred to them under your SaaS license in order to avoid inadvertently breaching the terms of the license and

facing a possible fine. • Use pay per use licensing model when using software application for a short period and a small number of people. • Use subscription licensing model when software application is required for longer periods of time. • Carefully scrutinise license for any additional payments that the service provider includes in the pay per use service offering and query additional payments if not satisfied.

- If not satisfied with service from one service provider, terminate the license and move to another service provider if the cost of moving are not prohibitively high.
- Avoid lock in by a service provider by not moving all software applications carte blanche but do it gradually and test software the service of service providers first by moving non-critical functionalities and if not satisfied move you custom elsewhere until you find the right service provider.
- Take advantage of the licensing flexibility offered by the SaaS market by selecting licenses that suits your needs and provides best value for the business. For example if a software application has functions that the business does not use a license that is based on functionalities can be used.
- Not all businesses or functionalities are suited for SaaS. Complex and highly customised software applications may not obtain the benefits of using SaaS. Therefore make an assessment of your business needs against all the software applications offerings available in order to formulate an IT solution that will add most value to the business.

2 Conclusion

Providing software services as an alternative to the traditional perpetual model has evolved (from the initial applications service provider (ASP)

offerings) to Software as a Service and appears to be making headways to achieving success. The future trends of software application sales reflect that perpetual licensing will no longer be the first choice for procuring software applications. This can be attributed to the availability of flexible licensing in SaaS which links customer value to pricing, as well as the many benefits that clearly outweigh the pitfalls. Despite this, perpetual licensing, it appears, will always have a niche to cater for as not all businesses and functionalities are suited for SaaS. Furthermore it is imperative that users considering a move from perpetual to SaaS IT purchasing to take heed of the pitfalls and consider the above recommendations to ensure a successful move from ownership to renting.

Taking into account the arguments for and against SaaS and the way it differs from the perpetual licensing, SaaS software licensing is a better business solution for both users and software developers.

References

Autocad. 2012. AutoCAD Services & Support. <http://usa.autodesk.com/adsk/servlet/ps/dl/item?siteID=123112&id=16485462&linkID=9240617>

Baltzan, P. & Phillips, A. 2010. Business driven technology. New York. McGraw-Hill International Edition.

Bontis, N. & Chung, H. 2000. The evolution of software pricing: from box licenses to application service provider models. Internet Research: Electronic Networking Applications and Policy. 10(3): 246-255

<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

Choudhary, V. 2007. Comparison of Software Quality Under Perpetual Licensing and Software as a Service. *Journal of Management Information Systems*. 24(2): 141-165.

Creece, G. 2010. SaaS vs. Software: The Pros and Cons of SaaS Pricing <http://blogs.gartner.com/guy-creese/2010/05/24/saas-vs-software-the-pros-and-cons-of-saas-pricing/> Date of access 04 December 2012.

Dalheimer, M. and Pfreundt, F. 2009. GenLM: License Management for Grid and Cloud. *Computing Environments*. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid: 132-139.

Epping, D. 2009. Per processor licenses for your application. <http://www.yellow-bricks.com/2009/06/04/per-processor-licenses-for-your-application/> Date of access 05 December 2012

Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R. and Concha, D. 2013. A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures. *Future Generation Computer Systems*. 29 (2013): 273-286.

Ferrante, D. 2006. Software licensing models: what's out there? *IT Pro*. November/December: 23-29.

Fisher, T. nd. Product Key (CD Key). <http://pcsupport.about.com/od/termsns/g/productkey.htm> Date of access 21 November 2012.

Gangadharan, G. R. & D'Andrea, V. 2011. Managing copyrights and moral rights of service-based software. *IEEE Software*. March/ April: 48-55

<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

Gartner. n. d. Software As A Service (SaaS) <http://www.gartner.com/it-glossary/software-as-a-service-saas/> Date of access 21 November 2012

Grimmick, R. n. d. What Is a HASP Dongle? <http://www.wisegeek.com/what-is-a-hasp-dongle.htm> Date accessed 30 November 2012

Gonsalves, A. 2009. Gartner Finds Most-Common SaaS Assumptions Are False. <http://www.informationweek.com/software/business-intelligence/gartner-finds-most-common-saas-assumptio/214600145?queryText=SaaS%20pricing%20models> Date of access 30 November 2012

Gull, D & Wehrmann, A. 2009. Optimized Software Licensing – Combining License Types in a License Portfolio. Business & Information Systems Engineering. 4: 277-288.

IBM. 2012. Processor Value Unit [PVU] licensing for Distributed Software. http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html#fullsubcap

Jiang, Z., Sun, W., Tang, K., Snowdon, J. L. & Zhang, X. 2009. A pattern-based design approach for subscription management of software as a Service Paper presented at Services – I, 2009 World Conference on, Los Angeles, CA, 6-10 July.

Ju, J., Wang Y. , Fu, J., Wu, J. & Lin, Z. 2010. Research on key technology in SaaS. Paper presented at the 2010 International Conference on Intelligent Computing and Cognitive Informatic, Kuala Lumpa, 22-23 June. <https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

Juniper. 2012. Introduction to the Media Access Control (MAC) Layer 2 Sublayer. http://www.juniper.net/techpubs/en_US/junos/topics/concept/mac-qfx-series-understanding.html Date of access 21 November 2012.

Kaplan, J. M. 2007. SaaS: Friend Or Foe? Business communication review. June: 48-53.

Konary. A. 2012. Licensing for SaaS Applications: ISVs Take to the Cloud. <http://www.tmcnet.com/tmc/whitepapers/white-paper.aspx?id=5970&Title=licensing-saas-applications-isvs-take-the-cloud> Date of access 04 December 2012.

Laplante, P. A., Zhang, J. & Voas, J. 2008. What's in a Name? Distinguishing between SaaS and SOA. IT Pro. May/June: 46-50.

Lindman, J., Rossi, M. & Puustell, A. 2011. Matching open source software licenses with corresponding business models. IEEE Software. July/August 2011.

Mohan Murthy M. K., Mohd N. A, Sanjay H. A. & Patel, M. Y. 2013. Software Licensing Models and Benefits in Cloud Environment: A Survey. Proceedings of ICAdC, AISC 174: 645–650 http://link.springer.com/chapter/10.1007%2F978-81-322-0740-5_76?LI=true#page-1 Date accessed 4th December 2012.

Matthew, M. & Nair, S. 2010. Pricing SaaS models: Perception of business service providers and clients. Journal of Services Research. 10(1): 51-68.

PC mag. n. d. Concurrent use license http://www.pcmag.com/encyclopedia_term/0,1237,t=concurrent+licensing&i=40223,00.asp

Robinson, B. 2006. The next wave of software licensing. InfoWorld. com. 25 September. pp 33-42.

Rohitratana, J and Altmann, J. 2012. Impact of pricing schemes on a market for Software-as-a-Service and perpetual software. Future Generation Computer Systems. 28: 328–1339

Rusu, S. 2011. MAC (Media Access Control) <http://forums.soft32.com/topic/50-mac-media-access-control/> Date of access 30 November 2012

Wong, P. 2012. Online Piracy an Keygens http://cosmos.ucdavis.edu/archives/2012/cluster4/Wong_Preston.pdf . Date accessed 30th November 2012

Yang, S., Yoo, B. & Janhg, J. 2010. Does the SaaS model really increase customer benefits? Asia Pacific Journal of information systems. 20(2): 87-100. <http://apjis.or.kr/pdf/MIS020-002-5.pdf> Accessed online 01 December 2012

Zettagrid. n. d. What is a processor license? <https://www.zettagrid.com/faqs/what-is-a-processor-license/> Date accessed 01 December 2012

Zhang, J. R. 2010. Improved Activation Using Multithreading. San José State University. (Thesis-Masters of science) http://scholarworks.sjsu.edu/etd_projects/157/ Date accessed 01 December 2012

<https://assignbuster.com/perpetual-versus-saas-sales-licensing-models-essay-sample/>

Zsigri, C. 2008. D2. 1 Software Licensing, Panorama Today. Version 1.

Seventh Framework Programme of the European Community. [http://www.smartlm.eu/fileadmin/SmartLM/user-data/D2.](http://www.smartlm.eu/fileadmin/SmartLM/user-data/D2.1_Software_Licensing_Panorama_Today.pdf)

1_Software_Licensing_Panorama_Today. pdf Date accessed 01 December 2012.