CHAPTER-4 BUSINESS MODELS IN OPERATING SYSTEM SOFTWARE MARKET

4.1: Generic Business Model

Business model establishes linkage between strategy planning and implementation. It transforms strategies into action. The taxonomies of Business models are enormous, yet the definite framework of a business model fails to exist. Traditionally, business model was a tool to identify the revenue generating activity. However, business model consisted of many dimensions that positively influenced organizations. Burkhart et.al (2011)\(^1\) made a seminal contribution to business model literature. The study identified gaps in the business model literature, viz lack of knowledge of interdependencies between the components of business model, lack of structured and comparable visualization of business models, less empirical studies on appropriate tools and criteria to evaluate business models. Raman Casadesus-Masanell and Joan Ricart (2011)\(^2\) state that alignment of business models with organizations’ goals, objectives and strategies are essential for organizations. Richardson (2008)\(^3\) opines that Business model establishes linkage between strategy formulation and strategy implementation.

The concept of Business model existed as an intuition based exercise. However, business model gained prominence with the advent of internet in 1990’s. The business model is one of the most researched concepts post internet era. Below figure indicates the number of publications related to business model in publication of non-academic journals (PnAJ) and publication in academic journals (PAJ) compiled by Zott et.al (2010)\(^4\).

![Figure 4-1 Articles related to Business Model in Business/Management journals](image)

---


The above Figure indicates that research on business model exploded beyond 1995 though it was conceptualized earlier. The attention to business model is attributed to many reasons by management thinkers. A few of the notable attributes are, digital economy, advent of internet, rapid growth in emerging markets, convergence of value chain etc.

According to Drucker, a good business model answers questions such as, who is the customer and what does the customer value? How do we make money in this business? What is the underlying economic logic that explains how we can deliver value to customers at an appropriate cost? The Business model design and product-market strategy are compliments not substitutes (Zott& Amit, 2008). Magretta (2002) opines that business modeling is the managerial equivalent of the scientific method you start with a hypothesis which you then test in action and receive when necessary. A Business model is widely distinct from strategy. It has a strategic importance due to its impact on business performance. Business model can, however lead to competitive advantage (Afuah and Tucci, 2001). Hence, business model components and its design gain strategic importance. Business model can be specific for organizations or an industry. Though there is no standard definition of a business model.

A few of the components that are core to business model can be identified as:

- A set of activities in the firm
  - Value chain
  - Value creation process
- Revenue generating process

The concept of business model can be examined from activity based view. An organization consists of a set of activities. The ability of a firm to integrate all activities efficiently and effectively might lead to competitive advantage. The ultimate objective of integrating activities in organization is to serve the customers better by making high quality products/services available. Activity based view considers perspectives of all stakeholders in the organization. The first set of components consists of activities that are required in making product/service and second part consists of making services/products available to customers.
The activity based business model varies across industry due to the nature of value chain involved in the industry. A few studies are available which evaluates industry specific business models. Rajala et al. (2003)\(^8\) identifies four elements of business model in software industry, namely product strategy, distribution model, revenue logic and services and implementation. Rajala and Westerlund(2005)\(^9\) identifies four types of software business models based on two dimensions, namely level of involvement of customer relationship and level of standardization. Schief & Buxmann (2012)\(^10\) made a seminal contribution in identifying business models existing in software industry. The study identified five clusters of business model elements, namely, strategy, revenue, upstream, downstream and usage. Software industry is one of the emerging industries. This is due to the fact that information technology is creating value in the organization and Software is one of the backbones of information technology infrastructure. Software industry has mainly two categories of a product namely, system and application software. Each classification represents a sub-category of software industry.

Business models in software industry are unique in many ways. One is uniqueness of value chain of software industry. Secondly, software product users are manufacturers of software for open source based software. The changes in the product are identified & modified by customers themselves under open source software environment.

Operating system software belongs to system software category. System software is created and made available to customers in various forms. The entire process of system software is examined from activity based view which covers the activity of creating system software and making system software available to customers. The analysis uses four pillars of business model proposed by Alexander Osterwalder (2004)\(^11\). The four pillars are:

- **Product**: What business the company is in, the products and the value propositions offered to the market?
- **Customer Interface**: Who the company's target customers are, how it delivers them products and services, and how it builds a strong relationship with them?
- **Infrastructure Management**: How the company efficiently performs infrastructural or logistical issues, with whom, and as what kind of network enterprise?
• Financial Aspects: What is the revenue model, the cost structure and the business model’s sustainability?

The next section elaborates the existing business models of system software. The business models have been analyzed from value creation and revenue generation perspectives. The first part deals with value creation and the revenue generation is analyzed from licensing and patents perspective in the second part.

4.2: Business Models from Value creation perspective

The value creation in the software context is related to writing of source code. The section below elaborates the value creation process of writing source code.

Existing Business Models for OS products

The system software is developed by software developers. The software is developed on technology platforms. Software developers write program/source code using programming languages. Generally, software is sold through licenses. The license indicates validity of product purchase and authorizes the product usage. Operating system (OS) products are generally sold using licenses. The licenses are used to protect intellectual property rights that define the conditions of usage of software purchased. A detailed discussion on licensing is covered in Section 4.7 of this chapter.

Source code is a set of programs that constitute a software product. Access to the source code is main differentiator in the business models adopted by major players in operating system software market. There are two broad categories based on access provided to source code. They are Open Source Code and Closed Source Code.

4.3: Open Source Code

In this category source code is open for access to users. This holds good for both corporate as well as home users. The users can download the source code, make necessary changes or customize according to their requirements and install on hardware. However, a few of the software are governed by General Public License (GPL) for the usage of open source code in the commercial products. The open source code concept was proposed and started by Mr. Richard Stallman who started GNU project. The aim of this project was to develop free and quality software.
4.4: Closed Source Code

Closed source code does not provide access to the source code to the users. Users have to buy software as a package and install on hardware. The users buy the software through licenses. The licenses and products vary depending on the corporate or home users. However, some of the software are developed using open source code, but they are required to acknowledge the usage of open source code in their product.

4.5: Business Model for Open Source code

The source code is written by a group of programmers. These set of programmers develop the source code generally out of their passion. Open source code is considered to be more secured and highly reliable, since the code is written and tested by a number of programmers. The code is tested under various platforms and applications. If any errors/bugs are found, it is resolved by programmers. The issues identified will be discussed on the network and solution to the error/bugs is identified. The programmers are connected with each other through online network platforms such as discussion forums, blogs and other membership based websites. The modified code is made available to customers as soon as the errors/bugs are fixed. Therefore, the code is considered to be more authentic and secured. The source code is made available on the internet and is open for download for all users. If there are any problems with the code, it is brought to the notice of the programmers through online network. Generally, the users of open source code OS have technical background and have basic knowledge of programming and hardware. Broadly, three categories of business models in open source code can be identified as mentioned below.

1. Direct download;
2. Purchasing Packaged open source code operating system; and
3. Sponsored open source code operating system.

4.5.1: Direct download

In this mode, users directly download the source code from internet. The source code is made available on the internet. The users can get customer support from third party vendors at a cost. Linux OS is an example in this type of business model.
4.5.2 Purchasing packaged open source code operating system:

In this category, different versions of the source code are bundled together and are made available to the users by vendors. The users have to pay for the bundled product. Generally, the bundled product includes CD (compilation of source code), users’ guide, installation guide etc; the users also do have an option of purchasing support services from the same vendor. In this model, though the source code is available free of cost, the package and customer support has to be purchased at a cost. Redhat, Suse etc, adopt this category of business model.
4.5.3: Sponsored open source code operating system

A few of the hardware manufacturers’ sponsored open source code operating system by supporting open source application products. These organizations have resources to develop open source code operating system and other applications which will be compatible to their hardware. The hardware manufacturers will be treated as ambassadors of open source code operating system. For instance, IBM established the Linux Technology Center (LTC) as the primary vehicle to participate in the Linux community. IBM and the LTC have established four goals for participation in the Linux community (IBM Systems and Technology Group, 2010):

- make Linux better;
- expand Linux’s reach for new workloads;
- enable IBM products to operate with Linux; and
- increase collaboration with customers to innovate in ways IBM cannot do by itself.

Hardware manufacturers like IBM and HP develop hardware which is based on Linux operating system. In addition, the application software of these organizations is compatible with Linux operating system. The hardware is loaded with open source operating system and is sold either directly or through distribution channels.

4.6 Closed Source Code

In this category, we have broadly two classifications:

1) OS sold to customers as a separate package and
2) OS integrated with hardware.

4.6.1: OS sold to customers as a separate package:

In this category, the users have to purchase the OS license separately in addition to the hardware. This category also includes OEM licenses, where users will receive pre-loaded operating system with the hardware purchased. Microsoft Windows operating system predominantly adopts this mode of selling operating system to customers. The licensing policy varies for home and corporate customers. The software package for corporate customers will have many features that are required and used at the organizational level under various technical projects.
4.6.2: OS integrated with hardware

In this category, the users need not buy OS separately. The OS is embedded with the hardware. This hardware is, generally, compatible with only a few numbers of applications. Mac OS follows this model. The OS supports only hardware of Apple. The OS is not much different for corporate and home customer segment. Since, the OS is integrated with the hardware; it is designed in such a way that it is compatible with most complex hardware specifications. This category will have end-to-end product.
4.7: Revenue generation in System software

The revenue is generated through licensing and customer support. Licensing is the main source of revenue. However, Licensing in software is a complex issue. It has many legal implications. Software patenting is another aspect of licensing which deals with intellectual property rights of creating software. The licensing and patenting have implications on internal and external environment of organization. Internally, it influences the features and functionality of software and externally influences the software distribution and usage. Therefore, implications of software and patenting on internal and external environment are enumerated in the following section. The external environment is examined through international trade and TRIPS agreement. Licensing scenario and software patenting is discussed in detail below:

4.7.1: Licensing of software

License in software context is authorizing an individual or organization to use software. End user license agreement is a legal document that authorizes the use of software. According to International Licensing Industry Merchandisers’ Association, License is an agreement through which a licensee leases the rights to a legally protected piece of intellectual property from a licensor – the entity which owns or represents the property — for use in conjunction with a product or service. The general definition of licensing of software indicates that the users do not buy licenses, but buy the rights/authority to make use of software. Software buyers cannot be owners of software unlike consumer products.

World intellectual property organization (WIPO) identified three categories of technology licenses, namely,

1. Licenses may be for certain intellectual property (IP) rights only (e.g. a license to practice an identified patent or to copy and distribute a certain work of authorship).
2. Licenses may be for all the IP rights of any kind that are necessary to reproduce, make, use, market, and sell products based on a type of technology (e.g. a license to develop a new software product that is protected by patent, copyright, trademark and trade secret law)
3. A license may also be for all the IP rights necessary in order to create and market a product that complies with a technical standard or specification (e.g. a group of
enterprises has agreed on a technical standard to ensure interoperability of devices—the group agrees to pool their IP rights and license to each other all rights each will need to manufacture and sell the product).

Current System software uses all three categories of licensing. Windows and Mac OS use licenses to protect its intellectual properties of operating system source code. Windows engages IP protection with group enterprises. For instance, Windows and Intel agree to produce interoperable hardware and software. They pool their IP rights and license to each other to produce processors that are compatible with Windows. Mac OS also follows the same structure of license agreement. System software license can be viewed from two prospective, one licenses for development and the other is license to use system software.

Licenses for development consist of agreements with hardware manufacturers or software development vendors. Windows or Mac OS will generate revenue through royalty or license fees in the mode of licensing for development. Licensing to use system software identifies the possible license options to customers while purchasing the software. The licensing policies of Microsoft, Apple Mac OS and Linux to users are briefly discussed below:

4.7.2: License options of Microsoft

The license options provided by Microsoft to customers are listed below. The license options have been cited from Microsoft’s Annual report 2012:

- **Open Licensing:** This license is suitable for small-to-medium organizations. Open Programs provides an option for customers to acquire perpetual or subscription licenses and, at the customer’s choice, access to future versions of software products over a specified time period. The process of providing rights to future versions of certain software products over the agreed period is referred as software assurance. Software assurance also provides support, tools, and training to help customers deploy and use software efficiently. Open Program has several variations to fit customers’ diverse way of purchasing. Under the Open License Program, customers can acquire licenses only, or licenses with software assurance. They can also renew software assurance when the existing volume licensing agreements expire.
• **Select Plus Licensing:** This license is found suitable for medium-to-large organizations, the Select Plus Program provides an option for customers to acquire perpetual or subscription licenses and, at the customer’s choice, access to future versions of software products over a specified time period. Similar to Open Programs, the Select Plus Program allows customers to acquire licenses only, acquire licenses with software assurance, or renew software assurance when existing volume licensing agreements expire. Online services are also available for purchase through the Select Plus Program.

• **Services Provider Licensing:** The Microsoft Services Provider License Agreement (SPLA) is a program aimed at service providers and Independent Software Vendors (ISVs) allowing these partners to provide software services and hosted applications to their end customers. Agreements are generally structured with a three-year term, and partners are billed monthly based upon usage.

• **Enterprise Agreement Licensing:** Enterprise agreements are suitable for medium and large organizations. Enterprises can choose perpetual licenses or, chose Enterprise Subscription Program, which provides non-perpetual, subscription agreements for a specified time period.

Microsoft volume licensing is an alternative to boxed software purchase. Volume licensing is provided to customers needing minimum of five licenses. Microsoft volume licensing provides only license of Microsoft software without user guide, CD/DVD. Customers will have many license options to choose from depending on requirement and organizations size. The licensing strategy of Microsoft has two components. One, software license and secondly support. Support is provided through software assurance subscription. Software assurance provides 24X7 customer support, upgrade of software, training, deployment planning etc. The license options have been categorized for corporate, academic, charity and government.

Microsoft Windows operating system on server needs server for license and client access licenses (CAL). CAL is required to operate on server either remotely or directly connected client. Customers have an option of buying multi-user client licenses or one user client licenses. These options are available in the recent window operating system for server. Microsoft Volume Licensing policies provide options of Windows client, Windows Server, and Windows Multipoint Server operating systems.
when the potential for multiuser scenarios exists. The Windows client operating system license terms do not permit multiple users to access or otherwise use one licensed copy of the software simultaneously. However, Windows Server operating systems and Windows Multipoint Server are designed and licensed for multiuser scenarios and should be used for all Windows multiuser scenarios. (Windows Licensing, 2011)\(^\text{13}\)

**4.7.3 License options of Apple Mac OS**

Apple Mac OS does not have elaborate license options unlike Microsoft. Mac OS is hardware integrated operating system. Mac OS works only on hardware of Apple. The operating system license is bundled with hardware for Mac OS. In the context of server licenses, Mac provides many simplified features for system administrators. For instance, The Mac OS server operating system does not require client access licenses. It provides unlimited client access to users. For instance, Mac OS Snow Leopard provides unlimited client access on servers. This will result in low cost, less administrative activities and less system administration resources.

**4.7.4: License options of Linux**

Linux operating system is considered to be open source. This operating system is one of the outcomes of GNU project which aimed at developing free software. In order to understand the license options of Linux, the knowledge of technical details of operating system is a must. Linux operating system consists of Linux kernel and other utilities. Linux kernel is low-level software manages hardware and multitask requests. Utilities include the program that helps users to copy files, create files, edit files etc.

Linux kernel licensed under general public license (GPL). GPL is conceived and designed by GNU project. The GNU General Public License is a free, copy left license for software and other kinds of works. The licenses take away the flexibility of sharing and changing the source code. The GNU general public licenses are aimed at providing flexibility to change and share the source code at no cost. The freedom to access source is not referring free software it is just freedom to access modify and release the updated source code. The pricing is optional. GPL allows the freedom to developers to design and distribute the source code with or without price. (GNU Licenses, 2012).\(^\text{14}\)
However, the packaged Linux operating system from various distributors such as Redhat, Suse etc, have license terms. Generally, the term “subscription” is used in place of license. The subscription is for per instance of Linux operating system on server. The subscription provides installation support, software upgrades, training and deployment of Linux operating system.

There are a few organizations like HP and IBM who sponsor open source operating system. These organizations design and develop Linux based servers. These organizations will also have resources working on open source based applications. Licensing terms in this context are defined based on processor or socket or users or per server. IBM offers licenses based on processor or number of users accessing the server. HP provides licensing options based on socket/processor/server or cluster. Processor based licenses grant rights of unlimited user access on a single processor. Socket license is similar to processor license. User licenses grants rights to access the software per user based. Cluster license provides right to access software from a cluster of nodes/clients.

Sponsored operating systems are integrated with hardware. Purchase of hardware and appropriate licenses will help customers access the operating system features. Generally, the maintenance contract provides customer support, software upgrade features and user training etc.

The licensing options provided by Microsoft, Apple Mac and Linux are directly related with product usage. The product usage is dependent on features and functionality of operating system. The features and functionalities refer to interface and inter-operability of software with other platforms and hardware. Licensing establishes the interface and inter-operability of software. The licensing options provided by operating system developing organization belong to two categories. One end user license and other licensing to third party vendors or complimentary product manufactures. End user license authorizes the usage of features by end users and third party vendor/complimentary product manufacturer’s licenses provide inter-operability features. The license with third party vendor/complimentary product is a critical component of operating system. This license authorizes the third party vendors consists of software application developers such as Adobe PDF, music players, network management products, data monitoring tool to develop their application that would be compatible with operating system. The complimentary product licenses
generally belong to organizations which develop their products in association with operating system developing organization. Microsoft works with Intel and Mac works with both Intel and AMD for processor hardware. Both organization will have mutual agreement and license the operating system with hardware that incorporates complimentary product such as License. The licensing has many implications to customers and society. The license might bind a customer to use a specific hardware and software which will result in customer lock-in. The customer may not have much choice but to use the applications that are bundled with operating system. The features and functionality provided by organizations are based on the business agreement between third party and complimentary product vendors. This forces third party and complimentary product vendors to provide an exclusive contract with operating system developer. The agreement might provide the exclusive compatibility feature with operating system. The other aspect of software licensing is patenting. Patenting protects the intellectual property rights of innovator. In software context, patenting is for an idea or functionality of software but not for a source code. Therefore, operating system developing organizations receive royalties by third party vendors or complimentary product vendors of operating system products for functionality of its inter-operability and other functionalities. However, software patenting has been area of agreement and disagreement by many thinkers and practitioners. Software patenting is a complex process and it is a challenge to identify the innovator of a patented functionality. Since the source code is a result of intellectual contribution of many programmers. The other aspect of software patenting is domestic law of patenting. Each country will have its patenting authority which considers the software patenting according to its domestic law. For instance, there are two countries, country A and country B. Country A is against software patents and country B uses software patents. Software developer in country A develops a software application and the similar application exists and is patented in country B. If organizations in country B use the non patented software application developed by country A, then the software developer in country B can take legal action against the organization which has used software application developed by country A. In this situation, who is at fault is debatable, the organization in country B or the software developer in country A. The absence of global software patenting framework results in such situation. Legal issues of System software is no exception in software patenting. Microsoft and Apple have been facing these issues right from its inception.
The legal issues faced by these companies are discussed below. The legal framework of software and patenting is discussed in the following section.

4.8: Brief overview of legal cases in operating system software market:

There have been many legal cases against the players in system software market. Microsoft faces more number of litigations/legal cases as compared to other players.

United States v. Microsoft is one of the significant cases of reference. During early 1990’s it was reported that Microsoft had anticompetitive strategies to establish its monopoly. The investigation of Department of Justice in the year 1994 ended with consent of decree in 1995. The key provisions of the consent of decree were:

Microsoft agreed to end OEM contracts with manufacturers which were based on per processor contract. However, Microsoft could offer unrestricted quantity discounts. Microsoft was . But this restriction will not stop Microsoft to develop integrated products. (Economides, 2001)\textsuperscript{15}. But this decree was not followed in 1998 when it released Windows 98. United States Department of Justice in 2003 asked Microsoft to split its business into OS business and application business and ordered interim conduct restrictions. The detailed verdict is provided in Annexure-6.

The allegation of monopoly against Microsoft was due to the fact that Microsoft Windows operating system was bundled with browser application called “internet explorer”. It had application program interface that would support few selected third party applications. This move was considered to eject Netscape browser from the market. The other factor that contributed to the monopoly was with its tie up with Intel processors. Windows operating system was compatible with only Intel processors.

Judge Thomas Penfield Jackson of United States District Court closely evaluated the strategies and found that, Microsoft possesses monopoly power in the market for Personal Computer (PC) operating systems. Microsoft engaged in a wide-ranging effort to protect its operating system monopoly, utilizing a full array of exclusionary practices. Microsoft’s actions were harmful to innovation and to consumers. One of the options given by court was to split Microsoft into operating system division and application division. The judgment also imposed restrictions on
conduct of Microsoft’s business. The restrictions were (Economides, 2001)\textsuperscript{16}.

1. Microsoft would create a pricing schedule that would apply to all buyers, so that price would not be conditioned on the sale of other Microsoft products.
2. Microsoft would not be allowed to have exclusive contracts that do not allow the other party to use, display, or feature its opponent’s products.
3. APIs and other technical information of Windows should be shared with outsiders as it is shared within Microsoft.
4. Microsoft is not allowed to take actions against manufacturers who feature competitors’ software.
5. Microsoft will allow OEMs to alter Windows in significant ways.
6. Microsoft is not allowed to design Windows to disable or compromise rivals’ products.

The restrictions revolved around scuttling the entry barriers and compatibility constraints set up by Microsoft. The antitrust cases against Microsoft existed in Europe region as well. European commission found bundling of windows media player with Windows operating system was affecting the European media players. The commission fined Microsoft €497 million in 2004. It had to remove windows media player from Windows for European region. Similar incidence was also reported in South Korea and was fined $32 million in December 2005 and had to sell Windows without instant messaging, Windows Media Player and Windows Media Service.

Microsoft had litigations with organizations as well. In 1988 Apple filed copyright infringement case against Microsoft for using GUI similar to Lisa and Mac Machines. However, the judgment was in favor of Microsoft. Similarly, Microsoft had litigations with AOL, Opera, and Alcatel-Lucent etc.

In Indian context, a case was filed against Microsoft. Singhania& Partners had complained to the Competition Commission of India (CCI) in 2010 that Microsoft was abusing its dominant position in the market, by offering lower-price software to original equipment manufacturers (OEM) than it did in volume licensing to law firms or other businesses, which were forced to buy allegedly overpriced volume licenses. The case was on pricing strategies of Microsoft across its distribution channels. CCI order under section 26(2) of the competition act, 2002, dismissed the allegations of Singhania & Partners. The case fails to illustrate the fact that the licenses are similar on the basis of intended use of the customer or all are equal in terms of efficiency,
royalty and on technical aspects. If the licenses provided similar features across all license types and charged different prices then CCI would have taken decision in favor of Singhania & Partners. Microsoft had legitimately sold differently designed software at different prices, that it had the right to sell a different license to an OEM than to an end-user, and that it could rightfully refuse transferring the software sold with one particular Machine to another old Machine. The current information did not reveal any other significant case against Microsoft in India.

The legal cases against Microsoft were pertaining to the anticompetitive and copyright infringements. The company’s strategic moves indicated that Microsoft wanted to establish monopoly and establish customer lock-in by bundling applications and erecting entry barriers for other competitors.

Apple also faces many legal cases. Most of the cases are on unfair practices and copyright violation across its products such as iPhone, iPad etc. There are very few notable cases on Mac operating system. Xerox had filed copyright infringement case against Apple for making GUI similar to Xerox Machines. However, the court took the decision in favor of Apple Mac. There was lawsuit by independent Apple resellers against direct Apple retail outlets. The allegation was that direct Apple retail outlets provided better pricing and discount offers than independent resellers. However, Apple reached to a settlement with its independent resellers.

Linux operating system does not face any legal cases directly. However, there have been a few cases on pricing and antitrust on sponsored Linux. For instance, IBM had to face a legal case due to its sponsorship of Linux operating system. The case accused IBM of not providing the agreed support and pricing for UNIX based IBM Machines. The petitioner mentioned that IBM is supporting open source applications and not providing agreed support to UNIX based Machines. There were not many notable cases against Linux in the present literature.

4.9: Software License and Patenting in International Trade context

4.9.1: International Trade and Business Model:

Business model essentially deals with value creation and distribution of product or service. Operating system products are truly global products. The product is conceived and designed by technical engineers across various countries. The product is exchanged across borders. The product exchange can be complete software or semi-finished software. Therefore, the value creation and distribution activities of system software products are across borders. The policies of international trade will
have an impact on value creation and distribution of system software. Therefore, it is essential to examine the relationship between international trade and business model. There are many countries involved in the exchange of system software. System software is a technology product. The international trade of technology is mostly governed by World trade organization.

World Trade Organization (WTO) is a body established to manage standards and policies for international trade. In order to cater to the trade policies for technology, Information technology agreement (ITA) was formed. US, Europe and Japan countries were technology oriented countries post 1990’s. They had many agreements related to manufacturing of semiconductor and other computer related technologies. These countries were able to foresee that technology will be one of the drivers of global economy. Hence, they contemplated having technology agreement for easier exchange of technology.

Information technology agreement (ITA) was a significant trade agreement signed by 14 WTO member states. This was the first sectoral agreement to be successfully negotiated between developed and developing countries. It was also the first agreement to fully liberalize trade in a specific sector. After the Uruguay Round, ITA provided participants to completely remove duties on information technology (IT) products covered by the Agreement. There are currently 74 participants – representing 7 per cent of world trade in IT products. (Information technology agreement, 1996)\(^{17}\).

The product categories identified by the ITA are:

- Computers,
- Semiconductors,
- Semiconductor manufacturing equipment,
- Telecommunication apparatus,
- Instruments and apparatus,
- Data-storage media and software, and
- Parts and accessories.
The categories are evolving over a period of time. It is a challenge for ITA to identify the appropriate classification due to the complexity and fast pace of changing technology. Computers and Data-storage media and software are relevant in the present context of study. ITA’s Computers category focuses more on hardware integration activities and data storage-media and software focuses on software in physical support. ITA policies are evolving over a period of time. The policy formulation are challenging for technology products due to complexity of product. WTOs objective is to make technology accessible across all the countries more easily from economic and utility perspectives. One of the challenges with respect to technology products is presence of global production network (GPN). The technology is conceived and developed across different countries. The intermediate technology goods cross many borders for value addition processes and final product is made available in a country other than the countries where value addition processes took place. In order to measure the impact of GPN on trade, ITA identified vertical specialization (VS) to measure the value added activities across borders. VS calculated as the percentage value of imports directly and indirectly embedded in the exports of a country. In addition to GPN, ITA faces a challenge in classification of technology. The classification is difficult due to technology complexity and interrelation between technologies.

The policies of ITA are directly related with system software. ITA might identify standard features of software and all the organizations competing have to consider standard features and develop product. Standardization of product features would affect the product offer in turn will have an impact on business model.

The categories lack clarity with respect to software. The classification of software is more towards the physical exchange of software. However, the software can be made available without physical exchange over internet. This could be one of the major limitations to interpret ITA for a software exchange. The Classification of computers focuses on exchange of hardware technology across borders.

A few studies were conducted to measure the impact of ITA on international trade. Jospeh and Parayil (2006) argued that there is no change in the demand for ITA products post agreement. However, ITA will help and cannot drive technology product demand across borders. There are other factors such as poor infrastructure, institutions, human capital and policies that might influence the demand of ITA.
products. Bora and Liu (2006)\textsuperscript{19} used gravity regression model to measure the impact of ITA. The results of the study showed that the participation in ITA has increased bilateral trade.

The other aspect of ITA which needs to be considered is the dominance of a few countries in the international trade. US, EU and Japan were the pioneering countries of ITA. The basic purpose of trade agreements was to facilitate easy exchange of technology across borders creating value for all trading countries. One of the unique features of technology is “standardization” of technology features. The standardization features establishes the basic architecture of technology. Technology architecture is a framework that describes the interaction and interconnectivity of components in technology. Generally, the country that establishes the technology standardization will have first mover advantage. The standard features set are generally based on the technology standard set in domestic market. The countries involved in ITA would like to establish technology standard. Once the standard is accepted and established, the country which has set standard will have competitive advantage. There are several factors that influence the acceptance of standard. A few of the factors could be customer lock-in, critical mass, demand of technology in domestic market. Many countries compete to establish technology standards. China is one of countries competing to establish technology standards. It is one of the key players in ITA. Suttmeier and Xiangkui (2004) call the attempt of China to establish standards as “neo-techno-nationalism”. It is a technological development in support of national economic and security interests which is pursued through leveraging the opportunities presented by globalization for national advantage. China is giving special attention to domestic software market. The software standard can be set if there is a strong domestic software market. The standards set at the domestic market will be followed by technology organization to manufacture hardware and software. Hardware and software operate as per the standard set by China. These hardware and software are used by US and European. Hence, US and Europe have to accept and incorporate theses standards in their technology products. For instance, WLAN authentication and privacy infrastructure (WAPI) standard set by China for wireless devices. The importers of this technology have to adopt WAPI since the wireless device manufacturers in China would manufacture devices according to WAPI standards. A few thinkers opine that this is a gross violation of WTO agreement,
however, none of the importers of technology which is based standard set by China have reported yet formally.

China is also engaged in developing alternative to Windows operating system. China-Japan-Korea open source software promotion partnership is established to find alternative to Windows. The partnership works on open source platforms such as Linux operating system to establish technology standards. Due to large domestic demand, China is capable of firmly establishing standards for Linux operating system. It is a challenge for the established standards of Windows. There exists open source movement across the globe. China does not want to miss this opportunity to establish its standard and control on software. Kylin is a Linux based operating system developed by China. Recently, Ubuntu had a deal with China to develop Ubuntu-Kylin for China. This is an attempt to replace Windows and establish open source based standards across technology industry.

Due to ITA and WTO, technology is freely exchanged across countries with appropriate tariff measures. Technology exchange has many dimensions that influence the acceptance of technology across the globe. From business model perspective, the technology policies of ITA will influence value creation and distribution process. There are not many restrictions in terms of distribution as such. The distribution is mainly governed through global licenses. These licenses are universally accepted. The other aspect of business model is value creation process. International trade policies are significantly related with value creation process. Technology is exchanged across borders for value adding processes. ITA essentially defines the framework of exchange of technology by GPN. However, GPN has to follow technology standards to add value to technology. The technology standard is strongly influenced by a country which is aggressively promoting its standards. The competing organizations have to follow global technology standards to produce their technology product. Hence, the business model has to consider the global technology standard for the value creation process of system software product in the upstream of value chain.

4.9.2: Legal environment and software products

There are two aspects of legal framework for software. One is software license and the other is software patent. These two aspects have been discussed below.
1. **Legal framework of Licenses**

Legal framework will not have any impact on licensing policies of organization among software products. Licensing policies of Microsoft, Apple and Linux are global licenses. However, there is geographic specific usage license which will be specified in the license agreement. The geographic specific usage restricts usage of license to the agreed upon location of use. The user can also purchase global licenses which can be easily used and transferred across geographical boundaries.

Software enables users to use computers efficiently and effectively. The usage is dependent on many factors. The existing literature indicates that one of the main factors that drive usage is interface across platform. This is related to the compatibility feature of software. The compatibility factor might lead to customer lock-in. The software might force users to use a specific hardware or software and restrict them to use hardware and software based user’s choice. This is one of the technical constraints in software. The situation of compatibility might lead to monopoly. A few of the studies indicate that Microsoft was able to establish monopoly due to technical compatibility constraints erected by its operating system products.

Microsoft in its annual report of 2012 stated that many antitrust and unfair competition class action lawsuits were filed against Microsoft across various state, federal, and Canadian courts by direct and indirect purchasers of PC operating system and other specific software products between 1999 and 2005. All claims in the US have been settled dismissed. It has been estimates that total cost to resolve charges range between $1.9 billion and $2.0 billion. At June 30, 2012, it recorded a liability related to these claims of approximately $500 million. The software product strategy using compatibility feature might be perceived as anticompetitive strategy in legal framework. Microsoft has been facing legal issues within and outside United States. As it is reported in its 2012 Annual report, the European commission was concerned about the inclusion of web browsing software. Based on this the Microsoft displayed an option of browser choice screen to users across all the personal computers in Europe which has Microsoft operating system. Microsoft failed to provide this option for Windows 7 preloaded PCs due to technical error. However, Microsoft did provide the Microsoft fixed this error as soon as it noticed. After fixing the error users got on option on screen to choose the browser application. However, on July 17, 2012,
European Commission announced that it had opened proceedings to investigate whether Microsoft had failed to comply with this commitment. The Commission mentioned that if any company is found to have breached a legally binding commitment, the company may be fined up to 10% of its worldwide annual revenue.

The journey of Microsoft has not been trouble free. The legal issues related to Apple Mac OS or Linux has been very negligible. One of the reasons could be the open access to source code. Apple Mac OS has been developing compatible interface to avoid compatibility constraints. However, Apple does have legal cases pending against their other products like iPhone, iPad etc.

The other aspect of legal framework is country specific. In the context of software, legal framework considers two components. One, software product and second its distribution in the country. Software product is evaluated based on features and functionality of product. A few countries’ legal framework may not accept the bundling of software applications with operating system. It might restrict user’s choice to use software application. Case against Microsoft at European Union cited above is an illustration legal interpretation/framework of product features and functionality. There are not notable cases against Microsoft in India. In fact, one of the major challenges for Microsoft in India is piracy. There have been many cases filed by Microsoft against Indian vendors for software piracy. Business Software Alliance (BSA) reports India has 63% piracy rate in PC software.

The second component of software license legal framework is distribution of software. Software is a global product. It is exchanged across borders in different formats. The law of the land determines legal requirements of software distribution. In Indian context, software attracts import duty if it is purchased outside Indian border. However, the software is purchased in various forms. Generally, the software is distributed through CD/DVD, OEM or download. CD/DVD and OEM will attract custom tax and free download may not come under the purview of the categories identified. Ministry of Finance under Circular No. 15 /2011-Customs dated 18 March 2011 clarifies the custom duty requirements and tax exemption for the sale of imported software in any form in India. The paper licenses, CD/DVD and OEM licenses fall under the categories where import duty has to be paid and service tax exemption/discount may be provided for resale of imported software.
2. **Legal framework of Software patents**

Patenting of software is a complex process. According to Article 10 of TRIPS agreement, software is classified under the category of Arts not under technology category. Lack of clarity on software in TRIPS agreement has created vacuum for interpretation of Law. The domestic law interprets software component of TRIPS according to its convenience. The agreement fails to categorize software appropriately. Therefore, it is difficult to identify the copyrighted or patented components of software. The software producers are unable to patent their source code due to lack of clarity in TRIPS agreement. Reichman (1995) states that software code cannot be patented. However, the software usage behavior of customer can be patented. Software usage behavior is related with the interface and interconnection with multiple platforms. The user pays for the interface not just the software code alone.

Software patenting has been interpreted in several ways by countries. There has been disagreement on software patenting across many nations. A few countries
are advocating open standards of software, where patenting will not have any relevance. A few countries do have patenting framework for software but lacks the clarity of what has to be included in patenting. There are two major agencies granting software patenting. One is confined to European region known as European Patent Office (EPO) which grants software patents and the other is for US region known as United States Patent and Trademark Office (USPTO).

Gert Kolle (1977)\textsuperscript{22} was one of the early advocates of open standards. Kolle argued that software patenting cannot exist. The software does data processing through instructions. The instructions are in the form of source code. The source code cannot be complied by one individual or organization. It involves a group of programmers. The source code is compiled from various authors/programmer. The author cannot be singled out and grant paten for a specific source code. The source code will also be integrated with hardware. Therefore, the patenting process gets more complicated.

According to en.swpat.org, In USA, the patent office is the authority which grants software patents and they have been upheld many times in lower courts., However, the Supreme Court never gave a verdict on whether a software is patentable or no. The European Patent Office is an authority that grants software patents in Europe. Most of the Courts in Germany have rejected them, but a few courts in the UK have upheld them. There is always uncertainty of the decisions. The patent holders are afraid of losing their patents and therefore they avoid going to court. However this may lead to more problems. There is always possibility of Software patent holders misusing the patent. They can threaten software developers, and they can demand sums of money. If the software developer doesn't have enough financial strength to defend themselves in court, resulting in the patent holder winning and will get money or market control though their patent is probably invalid. The other side of the issue is that the software developers are afraid of adding some compatible features due to the threat by patent holders. They are afraid of the cost involved in resolving legal issues. Therefore, they might exclude some of the applications or compatibility features.

There have been numerous studies and discussion on software patenting. Neither academia nor industry has resolved to a basic framework of software patenting. This will directly impact consumer. The usage of software is dependent on
software features. However, the software features such as interface and interoperability are linked with licensing and patenting. If the software developing organization is unable to have clarity on patenting, then the organization may not develop software as a bundle of many applications and features due to the fear of legal issues arising out of patenting. Some of the applications and features bundled in software may be patented and may not be disclosed. Mark Shuttleworth states that Microsoft is involved in an activity of racketeering. He says that Microsoft is asking to pay for patents but do not specify which features are patented.

Generally Microsoft is blamed for Patent trolls. Patent trolls are mechanism where organizations acquire patents to extract money from product developers. In the context of software, Microsoft is engaging patent trolls. It is apparent from a few of the cases filed in the court. For instance, Microsoft sued Melco group which deals with network attached storage devices. Microsoft said that Melco uses Linux operating system and a few of the functionality and features used in the system are patented by Microsoft. However, Microsoft has not declared the details of patents infringed by Melco. Patenting of software might result in customer lock-in. The features and functionality will be controlled through the patents. The software will be made available with product developers who abide by the patents and pay the required usage fees to add patented feature in their product.

It is evident from the facts that software patenting is a complex activity and will have direct impact on consumers. The software patenting revolves around the functionality and features of software.

The functionality and features can be managed through licensing. Therefore, licensing could be a substitute for copyright or patenting. Most of the licensing policies are universal. Therefore, the licensing terms might also cover the internationally traded software. WTO can incorporate software under appropriate category and provide the licensing framework for internationally traded software or technology. The clarity must be established in order to establish common ground to interpret copyright and patenting of software products.

4.10 Licensing and Business Model – An Evaluation:

Licensing is highly relevant in the software context. The software is made available to customers through licenses. It has direct implication on a few important
parameters of business model. Generally, licensing in software gains prominence in the downstream value chain of software products. License authorizes usage of software. Therefore, the license options play a significant role in product usage. Value chain analysis is used to demonstrate the importance of licensing in software industry.

Figure 4-6  Spiral value chain of open source software – A Macro View

Source: Author

The above figure shows the macro view of spiral value chain of open source software. The value chain has three activities viz, software development, software usage and software distribution. Software development involves the activities of source code development. The source code development can be developing new source code or editing the existing open source code. Software distribution includes the activities of making source code available to users. The source code is made available through various channels such as packaged CD/DVD, online or integrated with hardware. The software usage is the actual installation of software. The value chain is spiral because of the reason that open source software is used to develop one more software application which could be open source or proprietary. For instance, Mac OS is based on UNIX and UNIX is based on Linux. Hence, cyclical value chain exists in open source software.
The software distribution provides access to the source code or software application and software usage integrates the application with other application software. The usages of application software lead to new application development based on the existing application.

Business model essentially captures value creation and revenue generation. In the above value chain, the value creation is captured in two phases. One, product value creation and secondly usage value creation. Product value creation is through the development of new software application based on open source application. The new software application will add more functionalities and features to previous software application. Usage value creation is through licensing.

Licensing provides access to the source code of application and rights to use the software. In the context of open source software, usage value is created through the open access provided to source code and make use of it to further develop new software. The pattern of usage of open source application is dependent on licensing terms. The broad categories of licensing terms of open source applications are mentioned below:

- Open to open source application license scenarios;
- Open to close source application license scenarios and
- Hybrid source application license scenarios

Open to open source application license scenario exists when an organization uses open source application to develop another open source application. The source code of derived software application will be freely accessible. Open to close source application license scenario is a result of an organization developing closed source application based on open source application. The derived software will be proprietary software. Hybrid source application license scenario exists when an organization develops software based on open source and makes the derived software partially open. A few functionalities’ source code of derived software is proprietary.

The above scenarios are linked with value creation and revenue generating components of business model. Value creation is illustrated through product and usage value creation. Usage creation is dependent on licensing scenarios and licensing scenarios lead to source of revenue generation. Source of revenue generation is dependent on the license scenario. Revenue can be generated either through product
licensing or customer support. The adoption of licensing framework will identify the source of revenue generation.

Open source initiative (OSI) is a voluntary organization that establishes and classifies the licenses for open source applications. It advocates the fact that open source does not restrict to only access to source code, but also to distribution of open source. There are a few criteria for distribution of open source application. The criteria identified by OSI are mentioned below:

1. Free Redistribution

The license will not restrict any party from distributing derived or original software. This mode of license does not require any royalty fee. There is no price attached to this mode of operation.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form, where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications to derived works, and must allow to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code

The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination against Persons or Groups

The license must not discriminate against any person or any group.
6. No Discrimination against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of utility. The software can be used across many domains IT, pharma etc.

7. Distribution of License:

The license terms attached to the program must apply to all those to whom the program is redistributed without any modifications to the original license terms.

8. License Must Not Be Specific to a Product

The license must not be tied to a specific application or program. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not restrict the usage of other software which is distributed along with the licensed software.

10. License Must Be Technology-Neutral

The license should not be predicted or designed which is applicable to a specific application or software. This holds good for all the open source applications. The criteria cover the source code distribution of open source application.

    Licensing terms of OSI or general public license (GPL) includes the above mentioned licensing scenarios. The above scenarios directly relate to revenue generation component of business model. Licensing influences business model and the business model influences licensing. (Onetti & Verma, 2008) Onetti and Verma have identified three business models to be adopted by open source company as follows.
The business model of Onetti & Verma is based on the revenue generation. Reciprocal business model generates revenue from support and installation services. The organization that adopt reciprocal business model develop applications using open source software and will make the derived work available under the same licensing terms of the base open source software on which derived software is developed. Derived work is the application which is developed using open source software application. Dual business model provides an option for customer to either adopt open source or proprietary licensing terms for the derived work. Academic business model uses free source code. The users can use open source application at no cost and no licensing terms. The redistribution of derived work can also be made available free of cost. The above business model classification provides a restricted view of open source business models since the classification is majorly based on revenue generation perspective.

Therefore, it is imperative that licensing is linked with value creation and revenue generation components of business model. Licensing has a significant influence on business model and its performance. The analysis leads to similar conclusions drawn by Onetti and Verma (2008).
4.11 Analysis of Business Models

The business models are expected to enhance the performance of organization. The business model’s performance is linked with the performance of organization. There are many tools and framework available to evaluate business models. In the present study business model of Microsoft, Apple and Linux are analyzed based on the pillars and building blocks identified by Osterwalder (2004) shown below:

<table>
<thead>
<tr>
<th>Pillar</th>
<th>Building Block of Business Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Value Proposition</td>
<td>A Value Proposition is an overall view of a company's bundle of products and services that are of value to the customer.</td>
</tr>
<tr>
<td>Customer Interface</td>
<td>Target Customer</td>
<td>The Target Customer is a segment of customers a company wants to offer value to</td>
</tr>
<tr>
<td></td>
<td>Distribution Channel</td>
<td>A Distribution Channel is a means of getting in touch with the customer.</td>
</tr>
<tr>
<td></td>
<td>Relationship</td>
<td>The Relationship describes the kind of link a company establishes between itself and the customer</td>
</tr>
<tr>
<td>Infrastructure Management</td>
<td>Value Configuration</td>
<td>The Value Configuration describes the arrangement of activities and resources that are necessary to create value for the customer.</td>
</tr>
<tr>
<td></td>
<td>Capability</td>
<td>A capability is the ability to execute a repeatable pattern of actions that is necessary in order to create value for the customer.</td>
</tr>
<tr>
<td></td>
<td>Partnership</td>
<td>A Partnership is a voluntarily initiated cooperative agreement between two or more companies in order to create value for the customer.</td>
</tr>
<tr>
<td>Financial Aspects</td>
<td>Cost Structure</td>
<td>The Cost Structure is the representation in money of all the means employed in the business model.</td>
</tr>
<tr>
<td></td>
<td>Revenue Model</td>
<td>The Revenue Model describes the way a company makes money through a variety of revenue flows.</td>
</tr>
</tbody>
</table>

Table 4-1 Business Model building blocks

<table>
<thead>
<tr>
<th>Pillar</th>
<th>Building Block of Business Model</th>
<th>Microsoft</th>
<th>Mac OS</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
<td>Value Proposition</td>
<td>Stand alone Operating system with bundle of application software that meets minimum end user requirements</td>
<td>Hardware Integrated operating system</td>
<td>Hardware integrated and stand alone operating system</td>
</tr>
<tr>
<td><strong>Customer Interface</strong></td>
<td>Target Customer</td>
<td>Home/Enterprise Segment</td>
<td>Home/Enterprise Segment</td>
<td>Home/Enterprise Segment</td>
</tr>
<tr>
<td></td>
<td>Distribution Channel</td>
<td>OEM/ Distributors and Resellers/online</td>
<td>Online/direct/retailing</td>
<td>Network of users/partners/corporate sponsors</td>
</tr>
<tr>
<td></td>
<td>Relationship</td>
<td>Through partners and direct through online channel</td>
<td>Direct and Through partners</td>
<td>Direct and Through partners</td>
</tr>
<tr>
<td><strong>Infrastructure Management</strong></td>
<td>Value Configuration</td>
<td>Technical Engineers at Microsoft develop operating system, Microsoft has fairly good network with computer peripheral manufacturers which will design and produce Microsoft operating system compatible system</td>
<td>Technical Engineers at Apple develop Mac OS operating system bundled with hardware and software which are compatible on multiple platforms</td>
<td>The developers are users. The operating system is developed and errors are fixed in a network of developers</td>
</tr>
<tr>
<td></td>
<td>Partnership</td>
<td>Partnership exists with processor manufacturers, computer hardware manufacturers to create value for customers</td>
<td>Partnership does exist with limited application software developers</td>
<td>Linux is developed with partnership between users and corporate sponsored developers</td>
</tr>
<tr>
<td><strong>Financial Aspects</strong></td>
<td>Cost Structure</td>
<td>Cost of software development</td>
<td>Cost of software development and hardware development</td>
<td>No cost</td>
</tr>
<tr>
<td></td>
<td>Revenue Model</td>
<td>Licensing and user support</td>
<td>Hardware based operating system licenses and user support</td>
<td>User support</td>
</tr>
</tbody>
</table>

Table 4-2 Building Blocks of Business Model of Microsoft, Mac OS and Linux

Source: Compiled by Author based on annual reports of Microsoft &Mac OS and www.Linux.com
4.11.1: Building blocks of business model of Microsoft Windows, Apple Mac OS and Linux

The building blocks of business model proposed by Alexander Osterwalder have been used to enumerate the business ontology of operating system software market. The building blocks are:

1. **Value Proposition:** A Value Proposition is an overall view of a company's bundle of products and services that are of value to the customer.

   Microsoft Windows provide operating system as a bundle of applications. Operating system has to be loaded on hardware which has compatibility with Microsoft Windows. A few selected applications will be bundled with operating system. The applications such as media player, internet web browser, word processing applications etc. are bundled with desktop operating system and features such as server management and automation, virtualization, networking etc. are bundled with Microsoft Windows server operating system. Mac OS provides hardware as well as bundle of applications to users. The users buy bundle of applications along with hardware. Users need not buy operating system license and hardware separately. The hardware will have pre-loaded application software similar to Microsoft Windows. Linux operating system requires technical skill set. Linux operating system is available in three forms, viz, free download, packaged distributors and sponsored operating system. Free download, Linux kernel the low level software can be downloaded free of cost and install on hardware. There are many vendors who provide Linux based application software which have to be installed and configured by users. The second form of Linux operating system is through packaged distributors who provide Linux operating system and required applications to manage Linux operating system. The distributors provide technical support, installation support and training to the users. The third form of Linux operating system is available through sponsored organization such as IBM and HP. Sponsored organization bundle Linux operating system and supporting application software with their hardware. Sponsored organization will have resources that provide customer support, training and installation services. The users will be able to access the best of the hardware and software in this mode.

2. **Target Customer:** The Target Customer is a segment of customers a company wants to offer value to.
Microsoft Windows has been targeting both enterprise segments as well as home segment. Microsoft has two business divisions that focus on both segments. One, Windows and Windows live division focuses on desktop PC’s, secondly, server and tools division focuses on enterprise servers. Apple Mac OS has operating system for both home segment as well as enterprise segment. Mac operating system was predominantly used in home segment, but recent versions of Mac operating system provides many options for enterprise customers. Generally, the designers, developers and architects have been using Mac operating system. iMac® or Mac Pro® provide the features that are needed by enterprise customers. Mac Book Pro® and Mac Book Air® are portable system that will help home as well enterprise customers. Linux operating system is predominantly focuses on enterprise segment due to the technical nature of Linux operating system. In the recent days, the packaged distributors of Linux are focusing on home segment as well. A few of the hardware manufacturers are shipping laptops with pre-loaded Linux operating system.

3. **Distribution Channel:** A Distribution Channel is a means of getting in touch with the customer.

Microsoft Windows has three distribution channels. One is original equipment manufacturer (OEM), secondly through distributors and retailers and thirdly online. The major OEM licenses are for the Windows pre-installed PCs. The OS is also sold through servers and embedded operating systems. The other two channels viz. distributors & retailers and online sell Microsoft application software. The proportion of operating system license sale is less through these channels. Predominantly, operating system licenses are sold through OEM. Mac operating system makes product available through Online or direct retailing or third party resellers. Uniqueness of Apple distribution is direct retailing. Apple’s retail stores have been strategically located which has high traffic and quality shopping malls. It operates its own stores and locating them in desirable high-traffic locations the company is benefited and has helped to ensure a high quality customer buying experience and attract new customers. The format of the stores has simplified design which provides a better experience to customers. The retail stores are of various sizes to accommodate market-specific demands. Apple strongly believes in direct contact with its customers which it deals with an effective way to demonstrate the advantages of its products over those of its competitors. The stores employ experienced and knowledgeable personnel who provide product advice. The stores offer a wide selection of third-party hardware,
software, and other accessories and peripherals that complement the Company’s products. (Apple Annual report, 2012)\textsuperscript{26}.

Linux makes operating system available through online or package distributors such as Redhat, Suse etc and through sponsored organization such as IBM and HP.

4. **Relationship:** The Relationship describes the kind of link a company establishes between itself and the customer.

    Microsoft reaches customers typically through its partners. It does not establish direct formal relationship with customers. However, through online channel it establishes direct relationship with customers. Apple’s strategy is to directly establish relationship with customers. The direct retailing strategy of Apple provides direct access to customers. Linux operating system is developed by users. Therefore, there is direct relationship with customers. Sponsored Linux operating system establishes relationship through package distributors. All the major players establish relationship with customers either directly or through their partners.

5. **Value Configuration:** The Value Configuration describes the arrangement of activities and resources that are necessary to create value for the customer.

    The discussion of value configuration is based on source code development. Value is created for the customer in each activity of organization. The discussion focuses on the core activity of organization. The core activity of software is source code development.

    Microsoft Windows operating system is developed by the technical engineers hired by Microsoft. The innovation of product features is highly dependent on the caliber and creativity of technical engineers. The similar situation exists with Apple Mac operating system. Mac operating system is also developed by hired technical engineers. Linux operating system’s value configuration is unique as compared to Windows or Mac operating system. The users are only the developers of source code. The value creation process is initiated and contributed by users. Network of users contribute to the value creation. The errors, bugs and required modifications are identified and rectified by users only. Engineers at Microsoft and Apple edit and fix the bugs based on the user feedback.

    The source code is modified and customized by the value adding distributors. The distributors such as Redhat or Suse will have resources to modify the source code and customize applications.
based on the customer requirements. This will be additional value configuring activity in the value chain of operating system product.

6. **Capability:** A capability is the ability to execute a repeatable pattern of actions that is necessary in order to create value for the customer.

The component of capability is not included in the discussion due to the nature of operating system. The capability component of operating system is pertinent to source code development process. The source code development process is discussed in value configuration component. Therefore, the capability component is not included in the present discussion.

7. **Partnership:** A Partnership is a voluntarily initiated cooperative agreement between two or more companies in order to create value for the customer.

Partnership is a significant factor in system software market. The partnership will enable hardware and software organizations to arrive at a common standard and develop the product for customers. Microsoft has partnership with hardware and software application development organizations. For instance, Microsoft partners with software vendors through distribution policies for independent software vendors (ISV). The Microsoft ISV Royalty Licensing Program is a worldwide software licensing program that offers ISVs a convenient way to license Microsoft products and help vendors to integrate software business application. This is a relevant option for:

- Organizations that exclusively develop software or software solutions instead of being part of a computer system or being a hardware manufacturer;
- Organizations that obtain at least 30 percent of their overall revenue from the sales of licenses of that solution. (Microsoft annual report, 2012)\(^{27}\).

Microsoft has partnered with hardware manufacturers. For instance, Microsoft and Intel. Intel develops processors that have compatibility with Microsoft operating system. They will develop hardware and software which will be compatible to each other’s offerings. Similar to Microsoft, Apple also works with a few software vendors. The software vendors will develop applications that will run on Mac operating system. Apple and Intel work together to design and develop processors that support Mac operating system. Linux operating system is designed and
developed through partnership between users. The partnership also exists with sponsored organizations. Linux operating system is product of partnered networks.

8. **Cost Structure:** The Cost Structure is the representation in money of all the means employed in the business model.

   The cost structure in the context of operating system is based on cost involved source code development and hardware manufacturing. Microsoft operating system incurs cost in developing source code. The major cost in source code development is human resources. Apple Mac operating system has two costs, one hardware design and development and secondly the source code development. The source code of Mac operating system is based on UNIX. However, it engages technical engineers to customize Mac operating system. Linux operating system does not involve any cost of development. The operating system is conceived and developed by users. The users will be technically sound and are motivated to contribute to the development of source code. Recognition of source code contributors is one of the significant sources of motivation.

9. **Revenue Model:** The Revenue Model describes the way a company makes money through a variety of revenue flows.

   In operating system software market revenue is generated through licensing and customer support. Microsoft offers many licensing options to purchase licenses based on user requirements, user base and technical requirements. Microsoft has software assurance program through which it provides customer support, user training and configuration support to customers at a cost. Apple generates revenue through sales of Mac desktops and portable devices pre-loaded with Mac operating system. The user maintenance contract will provide necessary support, training and deployment facilities to customers.

   Linux operating system does not generate revenue per se. but sponsored and packaged Linux operating systems generate revenue through licensing and customer support. The detailed discussion on licensing is covered in section 4.7.

**4.11.2 Designing of Business Model**

According to Amit R. and C.Zott, the objective of business model is to create value to customers as well as the organizations involved. The business model should create customer
surplus while generating profit for the focal firm. That objective can also be referred to the value creating insight on which the firm turns and it is reflected in the customer value proposition. The value creation process is analyzed on the basis of activity system. Design of business model needs to consider two factors namely, design elements and design themes. Design elements include, content, structure and governance. Design themes include novelty, lock-in, complementarities and efficiency. The same framework is used to evaluate the business model of operating system products.

Below table captures design elements:

<table>
<thead>
<tr>
<th></th>
<th>Microsoft operating system</th>
<th>Mac OS</th>
<th>Linux operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content</strong></td>
<td>Source Code: Proprietary, Developed by Microsoft engineers</td>
<td>Hardware and software developed by Apple engineers</td>
<td>Open Source, developed by Technocrats, package distributors and sponsored organization’s technical engineers</td>
</tr>
<tr>
<td><strong>Structures</strong></td>
<td>Programme Development and core product</td>
<td>Hardware and Programme development</td>
<td>Programme Development</td>
</tr>
<tr>
<td><strong>Governance</strong></td>
<td>OEM and distribution partners</td>
<td>Direct retailing and distribution partners</td>
<td>Online and packaged distributors</td>
</tr>
</tbody>
</table>

**Table 4-3 Business Model design elements**

**Source:** Author

1. **Content:** The content factor identifies the activities to be performed. The core activity in operating system is programme development. This activity is varied across three brands. Microsoft and Linux have the similar activities except Mac OS which integrates hardware with software development. The programme is developed by hired technical engineers at Microsoft and Apple, whereas, Linux is developed by independent technocrats.

2. **Structures:** The structure refers to the linkage of activities. The linkage is the sequence of activities. The activity is classified as core, supporting or peripheral (Zott & Amit, 2007)\(^{28}\). In case of Microsoft, operating system is one of the core products and it is limited to only programme development. In case of Linux, program development is the core activity. However, for Mac OS, the core activity is hardware development and programme development.

3. **Governance:** Governance is referred to the component that performs the activity. It could be
franchising which could be a part of activity system governance. (Zott & Amit, 2007)²⁹.

The operating system is sold through licenses. Microsoft OS licensees are sold through OEM & distribution partners. OEM license is sold through hardware pre-loaded with Microsoft operating system. Microsoft will have an agreement with hardware manufacturers to ship the hardware with pre-loaded operating system. The hardware specifications to the customers will include the operating system as well. The licenses can also be purchased in commercial off-the-shelf. A few of the electronics and computer retail outlets will sell the licenses. It is mostly purchased by home customers. Microsoft OS licenses are also sold through distribution partners. Typically, the distribution partners cater to the requirements of enterprise customers. Linux operating system is made available through two channels, namely, online and packaged distributors. Linux online is available at no cost, whereas packaged distributors’ operating system licenses are available at a cost. The licenses are typically available through distributors for enterprise clients. However, in recent days, a few of the hardware manufacturers are selling hardware pre-loaded with Linux operating system. Apple Mac OS is hardware integrated operating system. These are sold typically through direct retailing and online. Mac OS is also made available through distribution partners.

**Below table captures design themes**

<table>
<thead>
<tr>
<th></th>
<th>Microsoft</th>
<th>Mac OS</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novelty</td>
<td>Nil</td>
<td>Integration of Apple Hardware.</td>
<td>Nil</td>
</tr>
<tr>
<td>Complementarities</td>
<td>A few of the supporting software with high switching cost.</td>
<td>Hardware and software with high switching cost.</td>
<td>Partially exists.</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Exists through compatibility features.</td>
<td>Exists through compatibility features.</td>
<td>Exists through compatibility features.</td>
</tr>
</tbody>
</table>

**Table 4-4 Business Model design themes**

*Source: Author*
1. Novelty: Novelty centered activity is related to the adoption of new set of activities or structure or governance. (Zott& Amit, 2007)\textsuperscript{30}. Microsoft and Linux have only operating system as a product. Mac OS has activity system which gets extended to all the products that are on Mac OS platform. The system is integrated seamlessly with other hardware which have on Mac OS. For instance, the data on iphone, ipad and iMac can be seamlessly integrated to access from one hardware. Hence, Mac has more novelty activity system as compared to Microsoft or Linux.

2. Lock-In: Activity system can result in customer lock in aimed at keeping third party/vendors as business model participants. Lock-in is established through switching costs, or network externalities derived from the structure, content and/or governance of the activity system (Zott& Amit, 2007)\textsuperscript{31}. In case of Microsoft, there exists high switching cost both tangible and intangible. Tangible costs include the feature of compatibility and intangible cost includes unlearning and learning new operating system. Unlike earlier days, the switching cost has reduced to a great extent due to the features of interfaces and compatibility. In case of Mac OS, there also exists switching cost; however, Mac OS is a hardware integrated operating system which reduces the compatibility and interface issues to a great extent. It will have switching cost only when a few of the hardware or software are not compatible with Mac OS. Linux operating system is yet to achieve its critical mass. Generally, switching cost is higher when customer base reaches critical mass. Customers will get used to the operating system and the vendors would develop the applications that are compatible to the operating systems. This result in lock-in. Linux may not achieve forcible lock-in unlike Microsoft or Mac, the lock-in for Linux might be achieved through product superiority.

3. Complementarities: Complementarities are result of bundling of activities in a system. It will create more value than running those activities independently (Zott& Amit, 2007)\textsuperscript{32}. In case of operating system, complementarities are delivered through the features of compatibility and interface. Microsoft does not manufacture any hardware, it provides only software complementarities. Similar to Linux, however, Mac OS provides hardware and software complementarities.

4. Efficiency: Efficiency refers to the way in which activities are arranged resulting in efficiency. An efficient activity system might result in reducing transaction costs. (Zott& Amit, 2007)\textsuperscript{33}. The factor of efficiency exists by providing compatibility features across multiple platforms on hardware and software.
4.11.3 Key Features of Microsoft, Mac and Linux operating system

Business Model analysis is based on an annual report of Apple and Microsoft. In addition, the table below is also used to analyze the business models of Microsoft Windows, Mac OS and Linux operating system.

<table>
<thead>
<tr>
<th>Microsoft windows server(2012)</th>
<th>Mac OS server</th>
<th>Linux operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Server Virtualization</td>
<td>• File Sharing</td>
<td>• Multitasking</td>
</tr>
<tr>
<td>• Storage</td>
<td>• Profile manager</td>
<td>• Demand loads executables:</td>
</tr>
<tr>
<td>• Networking</td>
<td>• Catching server</td>
<td>• Virtual memory using paging (not swapping whole processes) to disk</td>
</tr>
<tr>
<td>• Server management &amp; automation</td>
<td>• Xcode server</td>
<td>• A unified memory pool for user programs and disk cache</td>
</tr>
<tr>
<td>• Web &amp; application platform</td>
<td>• Time Machine</td>
<td>• Multiple virtual consoles:</td>
</tr>
<tr>
<td>• Access &amp; information protection</td>
<td>• Wiki server</td>
<td>• Supports several common file systems,</td>
</tr>
<tr>
<td>• Virtual desktop infrastructure</td>
<td>• Calendar, contacts and mail servers</td>
<td>• Many networking protocols:</td>
</tr>
<tr>
<td></td>
<td>• Xsan 3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-5 Key Features of Microsoft, Mac OS and Linux

The table 4.5 captures key features of latest variants of Microsoft Windows and Apple Mac operating system. The key features of Microsoft Windows Server (2012) revolve around optimum utility of server storage and accessibility. Apple Mac OS server key features focus on multiple features of user requirement such as networking, storage utility, application development etc. Linux operating system focuses on multiprocessor, multitasking activities for better utility of servers.

4.12 Proposed Conceptual Business Model

Business model of system software market has been analyzed from internal and external environment. The internal environment was evaluated based on value creation process, taxonomy on business model evaluation. The external environment was evaluated based on distribution channels, revenue generation, licensing, legal perspectives and international trade.
The analysis on these parameters indicates that business model of system software market comprises of value creation and distribution components which significantly affect players of market. The value creation belongs to internal environment and distribution corresponds to external environment.

1. **Internal environment**

   The core value creation is through writing code. The code is generated through either users or technical engineers recruited by organizations. The details of the core value creation are discussed in the model below:

2. **External environment**

   The external environment influences users to a great extent. The external environment deals with licensing and distribution of software. These two factors affect the usage of software. There is lack of clarity on software licensing and patenting in the international trade context. The software is truly a global product. The international community must address this issue and establish clarity on software patenting and licensing. This is directly linked with distribution and features of system software. The clarity on these issues can be established by creating techno-art policies. The policy must identify the standard features for software. These standard features are confined to the inter-operability functionality. If there are standardized features, then the software developing organization can develop software based on the standardized features. This will also solve the issues related to antitrust legal hurdles and anti-competitive strategies.

   The clarity of policy has to be established by WTO in its ITA and same should also be reflected in TRIPS agreement.

The proposed conceptual model through this study is given below:
Business model is shown below to depict the components of system software business model.

![Proposed conceptual business model](image)

**Figure 4-8 Proposed conceptual business model**

**Source:** Author

The above figure is proposed conceptual business model. The model has two value chains, one upstream value chain and secondly downstream value chain. The upstream value chain consists of value creation through writing of source code and downstream value chain pertains to the distribution of software. The components of proposed conceptual business model are explained below:

1. **Source code generation**

   Source code generation is the basic activity in system software. Source code consists of instructions to computer to perform a specific task. The source code can be open source or closed source. Open source provides access to source code whereas closed source does not provide access to source code.
2. Distribution

System software is made available to customers through various channels. It adopts traditional distribution channel strategies of direct retailing or independent retailing. For instance, Apple has direct retailing as well as independent retailing. Open source system software can be distributed through online or through independent retailing such as Redhat, Suse etc. The system software is also integrated with hardware and sold to customers.

3. Licensing Distribution

Licensing distribution provides the legal framework of usage and distribution of system software. The license can be distributed as a free download yet governed by GPL or non GPL. GPL and non-GPL have been discussed in section 6.3. OEM licenses are computers with pre-loaded operating system. Retail licenses cover paper licenses or licenses which are sold independent of hardware as a commercial of the shelf software. Licensing distribution will have direct relationship with revenue generation.

4. Support

Support includes installation, training, troubleshooting and documentation. System software is a technical product which demands for technical support to install and use the software. Hence, this is an important component in system support. This is one more source of revenue to organization.

4.13 Summary

The current business models of Microsoft, Apple Mac OS and Linux have been enumerated in this chapter. Business model building blocks and design elements have been discussed above. The role of WTO and ITA and its implications on business model has been explained. Based on the internal and external environment analysis, the key components of upstream and downstream value creation of business model for system software market have been identified and explained.
References


16 Ibid.
24 Ibid.


29 Ibid.

30 Ibid.

31 Ibid.

32 Ibid.

33 Ibid.