applied sciences

MDPI

Article

A Method for Managing Software Assets in the Automotive Industry (Focusing on the Case of Hyundai Motor Company and Parts Makers)

Changhan Ryu [1] and Sungryong Do [2,*]

[1] Technology Planning Team, HYUNDAI AUTOEVER Co., Ltd., Seoul 06176, Republic of Korea; rchworld@gmail.com

[2] Department of Computer Software, Kyungmin University, Uijeongbu 11644, Republic of Korea

* Correspondence: imdosungryong@gmail.com

Abstract: We propose a method for managing software assets in the automotive industry to enhance software competitiveness and to reduce development costs. The ownership of software assets in the automotive industry is held by automotive parts companies, making it challenging to exchange these technologies. Moreover, the criteria for determining software assets are often unclear, resulting in difficulties in integrating automotive software and implementing over-the-air updates. To address these issues, we suggest breaking down black-boxed software assets into tradable components, valuating them, and introducing the concept of exchanging software technology assets. Additionally, we provide a structured approach for recycling used software assets and establish a software asset management system for registration and tracking. Our proposed approach can help traditional automotive OEMs narrow the technology gap with automakers such as Tesla and improve their software competitiveness in the automotive industry. This paper contributes to the advancement of software asset management practices in the automotive industry, and provides insights into the integration of automotive software and over-the-air updates.

Keywords: automotive industry; software asset; technology valuation; R&amp; D management

check for updates

Citation: Ryu, C.; Do, S. A Method for Managing Software Assets in the Automotive Industry (Focusing on the Case of Hyundai Motor Company and Parts Makers). *Appl. Sci.* **2023**, *13*, 4174. https://doi.org/10.3390/app13074174

Academic Editor: Vito Conforti

Received: 4 February 2023
Revised: 19 March 2023
Accepted: 21 March 2023
Published: 24 March 2023

1. Introduction

In the automotive industry, where hardware competitiveness has always been emphasized, software is now breaking down the barriers between industries and emerging as a crucial factor in determining the future success of companies [1]. Software is no longer an additional component but is recognized as a core competitive factor, on par with the existing powertrain [2]. The hardware competitiveness gap among automakers is gradually closing, and most of the innovation in the automotive sector is expected to be triggered by software [3]. In particular, the transition to the MECA (Mobility, Electrification, Connectivity, Autonomous) era is an important background factor that motivates automakers to focus more on securing software competitiveness [4]. Accordingly, major automakers such as Volkswagen and Toyota are pursuing the transition to software companies through various methods such as startup investments, M&A (Mergers and Acquisitions), and joint ventures to strengthen their own competitiveness in software [5].

The automotive industry tends to favor incremental, process-oriented innovation over radical innovation, as the latter involves complex operating methods, low profits, and high risks [6,7]. Additionally, due to its capital-intensive and highly competitive nature, companies in the industry strive to concentrate their core competencies and prevent the emergence of new competitors, alternative business models, or technologies [8]. However, since there are limitations to internal technology and R&D capabilities, the proportion of ICT companies and telecommunications companies in the automotive sector is expected to increase [9–11].

Customers are increasingly demanding advanced software functions such as advanced driving assistance and autonomous driving capabilities. However, the current vehicle architecture, which is independently designed for each part, has limitations in realizing these functions [12]. A new electrical and electronic architecture is needed to maximize software commonality, integration, and connectivity [13]. Major global automakers are strategically transitioning towards solving these problems through step-by-step approaches [14,15]. The recent trend of separating hardware-software development and procurement, along with the creation of dedicated software organizations to strengthen independent development capabilities, is seen as an effort not only to secure proprietary software technology but also to reduce the cost of transitioning to a new electrical and electronic architecture [16].

In fact, to shorten the development period of high-quality software, it is necessary to develop a standardized platform and ensure that legacy code, which has been applied to mass production with proven reliability, can be effectively utilized [17]. Currently, the problem of legacy code ownership and the difficulty of evaluating the value of software technology is hindering software technology transactions, and the reality is that it is difficult to justify its value [18]. Additionally, investment in advanced features is mainly made by automakers, and when most parts companies develop software centered on mass-production development, it is difficult to distinguish ownership solely based on the conventionally used contracts [19]. To solve this problem, it is necessary to understand the importance of software and prepare a new, detailed contract between automakers and parts makers based on a mutually beneficial attitude [20].

In other words, a win–win situation is possible between automakers and parts makers who belong to group companies that recognize the core competitiveness of the MECA era as software [21], and a strategy for strengthening software competitiveness is promoted at the group level [22]. To expand their global business and strengthen the group's software competitiveness, this study proposes the establishment and operation of software value estimation standards that can be jointly used by group companies [23]. To achieve this, benchmarking information obtained through domestic and foreign channels is used to establish software value calculation standards and conduct software technology value evaluations for group companies' software (Basic SW, Application SW). It is judged to enable reasonable technology value evaluations.

2. Conceptual Background

2.1. Characteristics of the Automotive Sector

In the past, automobiles were merely a basic means of transportation with machinery as the primary component. With recent advancements in ICT technology, sophisticated technologies such as electronic devices, electronic control devices, and ITS have emerged, providing car users with safer and more convenient features [24]. As a result, automobile manufacturers are gradually transitioning from individual module-based control to an integrated control approach at the component level, and new product development is also increasingly focused on developing electronic components for safety and convenience purposes, rather than solely on design-oriented development [25].

As the use of electronic devices continues to expand, the software complexity and integrated control modules for overall vehicle control are also increasing [26]. Furthermore, as the number of lines of code (LOC) in software increases with module coupling and cohesion, complexity also increases, leading to more vehicle safety accidents and recalls due to software defects [27]. Vehicle software can be categorized based on function and has been updated and mass-produced individually to meet the needs of each field without taking into account the vehicle architecture [28,29].

To summarize, automakers have been developing electronic systems and application services based on firmware for each function without thoroughly considering the overall vehicle architecture (Table 1). This structure, which closely and intricately combines hardware and software, has been developed and updated only for specific functions and specific hardware, making it difficult to reuse software. This slows down the development cycle of automotive electronic parts and acts as a factor that lowers the level of integrated control.

Table 1. Classification of vehicle SW.

Category	Key Technology
Infotainment software	Infotainment SW platform, tools, etc., applied to convenience devices, etc.
OS for integrated control of electronic parts	SW for integrated vehicle control such as OSEK and RTOS; SW platform for safety and electronic components such as electronic devices applied to vehicle safety devices, tools, etc.
SW for external connection and communication service	Connected SW area used to connect inside and outside the vehicle, such as communication and big data, artificial intelligence and IoT interworking, V2X interworking, etc.
SW for development and verification	Vehicle SW development tool, verification tool, verification device driving, etc.

Recently, the need for automotive companies to develop a comprehensive architecture has been further increased by the interlocking of vehicle convenience devices with mobile devices and IoT devices, as well as the expansion of user convenience through connections with vehicle peripheral facilities, transportation infrastructure, OTA, and so on. To improve the efficiency of automotive electronic embedded software development, automakers first define and apply a standardized structure based on a protocol called AUTOSAR (AUTomotive Open System ARchitecture) [30]. This standardized software enables collaboration and division of labor among different companies for each module, shortening the development schedule, and modularizing frequently used functions for horizontal deployment to other vehicle models [31]. However, at this time, legal issues regarding liability arise in case of quality problems if additional contracts are not made with parts companies according to the horizontal development of each vehicle type.

2.2. Characteristics of Software Supply in the Automotive Sector

The traditional structure of the automotive supply chain is a pyramid structure with OEMs at the top, where Tier 1 supplies vehicle electronics subsystems to the OEMs. In this subsystem, Tier 1 provides its own controller, operating systems, and application in the form of a black box, and the OEM receives these subsystems from multiple tiers, integrates them into a single electronic system, and applies them in mass production for the finished vehicle [32].

In summary, Tier 1 provides the (sub) system called ECU (Electronic Control Unit) to OEMs in the form of a black box, and here its own IP is included in the cost. At this time, the value of software is not recognized alone, but acknowledging the subordination of intellectual property rights to the hardware system, their significance is duly recognized. However, with the recent emergence of specialized software companies, the software Tier 1 is now communicating directly with the OEM and supplying only the software to the OEM based on its own software specifications. Currently, the OEM interacts with the traditional tiers in a different way. The software Tier 1 requests the OEM to implement its own software or software development specifications in hardware. In this case, Tier 1 may be reduced to a supplier of hardware only, and as ECUs that were previously provided as black boxes become white boxes, intellectual property rights (IP) may not be recognized, resulting in the deterioration of Tier 1 profitability. This trend is expected to intensify in the future.

Until now, Tier 1 has supplied (sub) systems equipped with proprietary hardware, operating systems, and domain software. The OEM had to accept the ECUs required for electronic systems even though they used various operating systems and semiconductors (MCU, CPU, GPU, etc.) from different manufacturers, as without Tier 1's cooperation, it was difficult to optimize the required performance and function in a timely manner or to release the product required by the market in time. However, as the importance of software is being emphasized, OEMs are working to improve this business structure by standardizing their own software platform or, furthermore, the architecture for electrical and electronic equipment. As a result, they will move on to receiving only (sub) systems with low added value from their suppliers. The following data explain that the Electrical/Electronic (E/E) architecture of traditional vehicles has a distributed structure with dozens of ECUs connected in a network [33], while Tesla's architecture is integrated in a centralized manner (Figure 1). The technology gap between traditional automotive OEMs and Tesla's software control architecture is estimated to be around 10 years, and steps are being taken to narrow this gap.

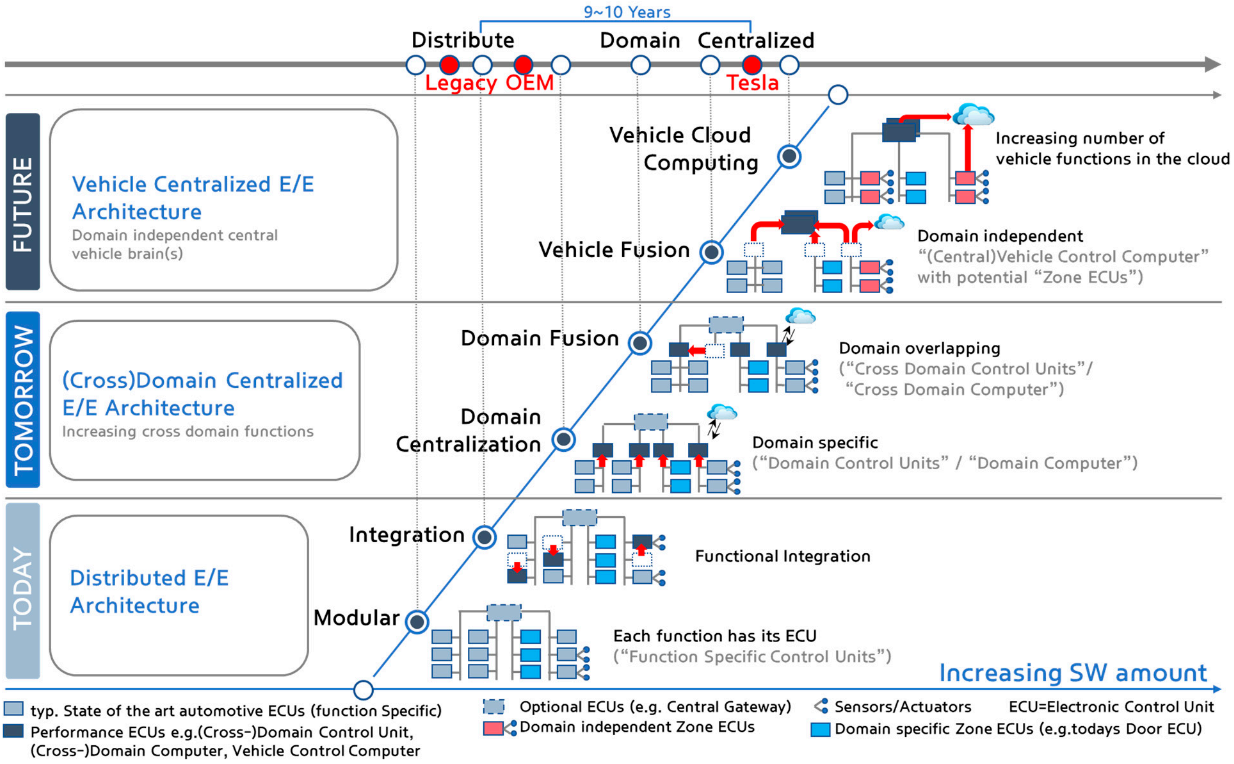

9~10 Years
Distribute
Domain
Centralized
Legacy OEM
Tesla
FUTURE
Vehicle Centralized E/E Architecture
Domain independent central vehicle brain(s)
Vehicle Cloud Computing
Increasing number of vehicle functions in the cloud
Vehicle Fusion
Domain independent
"(Central)Vehicle Control Computer" with potential "Zone ECUs")
TOMORROW
(Cross)Domain Centralized E/E Architecture
Increasing cross domain functions
Domain Fusion
Domain overlapping
("Cross Domain Control Units"/ "Cross Domain Computer")
Domain Centralization
Domain specific
("Domain Control Units" / "Domain Computer")
TODAY
Distributed E/E Architecture
Integration
Functional Integration
Modular
Each function has its ECU
("Function Specific Control Units")
Increasing SW amount
typ. State of the art automotive ECUs (function Specific)
Performance ECUs e.g.(Cross-)Domain Control Unit, (Cross-)Domain Computer, Vehicle Control Computer
Optional ECUs (e.g. Central Gateway)
Domain independent Zone ECUs
Sensors/Actuators
ECU=Electronic Control Unit
Domain specific Zone ECUs (e.g.todays Door ECU)

Figure 1. Possible evolution of vehicular E/E architectures (Revised).

Major global OEMs, such as Volkswagen, Toyota, Mercedes-Benz, and Volvo, are already implementing this phenomenon [34]. By standardizing their own software platform or electrical and electronic equipment architecture, OEMs can achieve cross-layer optimization, shorten the development period, improve software quality, and diversify their suppliers away from Tier 1-dependent business structures [35]. This is a desirable outcome for OEMs if they can meet the conditions of technological prowess and timely recruitment of excellent software talents, but it will be a significant threat for Tier 1.

In other words, the cooperative relationship between OEMs and Tier 1 is shifting towards competition as both parties strive to internalize the underlying software platform, such as the BSW (Basic Software). In order to maintain control over ECU software, Tier 1 is expected to compete with OEMs by providing black-box based products in a larger format. For instance, Volkswagen is reducing software costs by utilizing legacy codes, minimizing source code changes through a common software platform, and promoting software development through an update method that reuses previously applied source codes for mass production. Volkswagen is currently developing VW. Operating System, a single software platform for their vehicles, and has established a company with 2000 employees dedicated to developing Volkswagen vehicle software [36]. Toyota has created a company named TRI-AD, which can develop software for their vehicles. TRI-AD was established by bringing together developers from Toyota, Denso, and Aisin [37]. Tesla shares software developers with SpaceX, and software developers from NASA are collaborating to develop vehicle software [38].

2.3. Software Valuation Methodology

There are various methodologies for technology valuation, which can be classified into three categories: the income approach, market approach, and cost approach. A mixed methodology that combines these categories has also been developed.

The income approach is based on discounted cash flow methods (DCF). This approach calculates the expected future free cash flow that a company can generate from its operating activities in the future, and the level of risk associated with this cash flow. By dividing the net present value, which is discounted using a rate that reflects the total number of issued shares, this method calculates the future rate of return that a company can achieve as a future profit per share.

The market approach is a valuation method that assesses the value of technology based on the market price that is typically established between parties in a transaction who possess sufficient transaction information and willingness to trade. To utilize the market approach, there must be transactions between independent parties in an active market where comparable technological assets are traded, and there must be ample technological assets that have been traded in the past, with accessible information on their transaction prices [39]. In many instances, it can be challenging to find a comparable target [40].

The cost approach is a valuation method that estimates the various costs of software development, based on the principle that a rational person will not pay more than the cost to obtain a software asset with equal need or utility [41]. However, there are limitations to this approach in that input costs do not necessarily relate to future value or revenue generation, and not everyone can create the same technology asset value with the same cost [42,43]. To overcome these limitations, research is being conducted to determine optimal prices through software development cost analysis, markup level analysis, and software standard pricing based on the cost-oriented pricing method.

This study employs a combination of the income approach, market approach (specifically, the royalty approach), and cost approach to establish a reasonable standard for valuing software in the automotive industry. Furthermore, the study includes an in-depth analysis of the concept proposal for technology transactions and the utilization of existing legacy code to expand IP-based business.

3. Analysis

3.1. Software Asset Identification and Securing

The automotive industry is experiencing an increase in the added value of software due to the decoupling of software from specific hardware, which is being driven by OEMs' efforts to capitalize on software [44]. Tesla has emerged as a game-changer in this regard, leading the way in car changes centered on software using Over the Air (OTA) updates [45]. With the entry of ICT companies such as Google (San Bruno, CA, USA), Samsung (Seoul, Republic of Korea), and QNX (Ottawa, ON, Canada) into the automotive industry, software technology in the automotive sector is rapidly advancing, and software is becoming more modular and common in the MECA areas [46]. As this trend continues to spread to other OEMs, it is highly likely to extend to all areas of vehicle control.

Despite OEMs' efforts to take ownership of software and decouple it from specific hardware dependencies, OEM-owned software remains a black box, with a high likelihood of shared ownership with other companies. Therefore, if the software code is transferred to a third party for outsourced development, issues regarding ownership are likely to arise.

To facilitate IP-based business expansion and collaboration with global companies, development results should be conceptualized, managed, and operated as BIP, FIP, SIP, PIP, and OIP, and the IP concept should be clearly specified in the contract [47].

If the IP concept is not addressed in the contract, it becomes challenging to distinguish between existing and newly developed source code, making it difficult to claim ownership of existing technological assets when providing output (Table 2). OIP was added to reflect the increasing use of open-source software in the automotive industry and the trend of global companies developing source codes using open-source software. Automotive manufacturers are required to provide notice of their use of open-source software, with each company operating its own notification website. A review of notification sites belonging to Volkswagen Group, Ford Motor, General Motors, Stellantis, and Hyundai Motor revealed that the most recent models exhibited an increased rate of open-source utilization. Given the rising percentage of software incorporated into automobiles, it is reasonable to expect a corresponding increase in the employment of open source solutions.

Table 2. Intellectual Property (IP) concept (Revised).

Category	Definition
BIP (Background Intellectual Property)	Existing IP developed by each partner before the start of the collaboration contract
FIP (Foreground Intellectual Property)	IP created by the cooperative partner within the scope of the collaboration agreement
SIP (Sideground Intellectual Property)	IP related to collaboration and developed during the collaboration, but not within the scope of the collaboration
PIP (Postground Intellectual Property)	IP related to the collaboration and developed by each of the collaboration partners after the end of the collaboration
OIP (Opensource Intellectual Property)	IP related to freely used, modified, and created by individuals or groups

The time has come for collaboration with future competitors to shorten technology development periods and apply necessary technology in a timely manner through open innovation [48]. However, in the case of joint development, it is necessary to clarify the BIP that the two companies developed and secure ownership of it (Figure 2), and to actively address the IP division issue that occurs at the beginning and end of the collaboration [49]. This problem may become more complicated if multiple companies develop multiple controllers together [50]. If the BIP is not systematically arranged in advance, it is inevitably impossible to secure one’s technical assets [51,52].

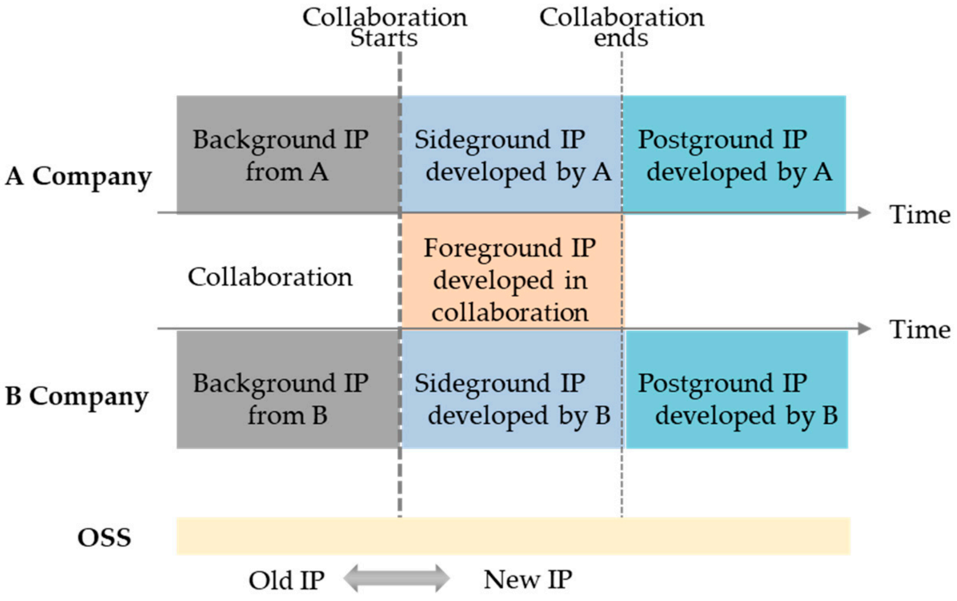

Collaboration Starts
Collaboration ends
A Company
Background IP from A
Sideground IP developed by A
Postground IP developed by A
Time
Collaboration
Foreground IP developed in collaboration
Time
B Company
Background IP from B
Sideground IP developed by B
Postground IP developed by B
OSS
Old IP
New IP

Figure 2. IP definition for joint development (Revised).

When developing jointly with a partner company, the following must be specified in the contract [51]. First, when developing the ‘1st controller’, the BIP of the partner company and the company shall be specified in the contract, and it shall note that the FIP jointly developed within the project period is jointly owned, while the SIP developed separately by each company is owned by each company. After the project is over, the PIP developed by each company is owned by that company. When developing the ‘2nd controller’, the IP obtained from the ‘1st controller’ development goes through the process of capitalization and is registered in BIP, and then will be developed jointly with another partner. After the joint development is completed, the BIP is secured by going through the IP capitalization process in the same way as in the previous process and will then again be used for the development of the '3rd controller'. OIP used for controller development is well organized and sent to OEMs for notification. Currently, ownership follows the open-source license regulations (Figure 3).

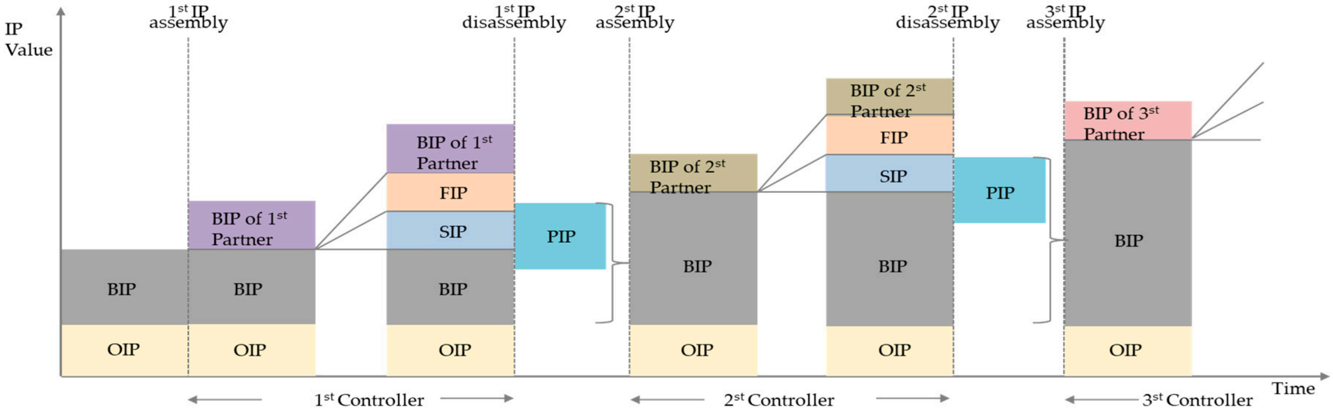


Figure 3. Systematic securing of BIP (Revised).

3.2. *Software Value Calculation Method*

Software valuation can be performed by selecting an appropriate methodology depending on the intended purpose. The methodology for calculating value is determined based on the technological maturity of the software, which can be evaluated using the income approach, market approach, or cost approach. The specifics of each approach are detailed in the table below (Table 3). In summary, the appropriate value estimation methodology can vary depending on whether the software is in the early stages of development or has reached the commercialization stage [53].

Table 3. Calculation of value based on technological perfection.

Value Degree	Stage	Value Calculation Method	Accuracy
High ↑ Low	Commercialization Development completed Developing Initial	Market approach/Income approach Income approach/Market approach Alternative cost approach/Royalty deduction approach Regeneration cost approach	High ↑ Low

The target of technology valuation can be categorized into technology assets, technology, intellectual property, human assets, and market assets. Depending on the purpose and nature of the valuation, it is common to exclude human assets and market assets during the evaluation process [54]. At present, the evaluation of technology encompasses a range of factors, including the economic life cycle of the technology, sales projections, financial analysis, economic profit flow, discount rate, business value, technical contribution, and technology value. When conducting a software valuation, the value assessment is typically based on technical maturity; however, in certain cases, all three methodologies are utilized to appraise the value, followed by a consultation process.

The income approach uses the DCF (Discounted Cash Flow) method to calculate the VT (Value of Technology). It is determined by multiplying the degree of technology contribution, which is the extent to which technology contributes to business value, as shown in Equation (1) below. When determining the technological contribution, it is important to avoid underestimating the value of software assets by calculating the contribution of technology as the proportion of software in certain modularized parts of the vehicle's value. In Equation (1), t represents the period during which cash flow estimation is made; n represents the economic life cycle of the technology; CFt represents the cash flow in period t; r represents the discount rate; and technology contribution represents the contribution of technology to business value [53].

$$\mathrm{VT} = \sum_{i=0}^{n} \frac{CFt}{(1+r)t} \times \text{technology contribution} \tag{1}$$

The royalty deduction method assumes that a company licenses technology from a third party because it does not possess the target technology. Generally, a license fee must be paid during the economic life of the licensed technology, and the present value of the total amount of royalties paid is estimated as the technology value. To use the royalty deduction method, there must be many comparable technology transaction royalty data in the technology transaction market. In the case of automobile and trailer manufacturing, the median royalty value of 3% can be applied to the calculation if statistics of the technical evaluation guide are applied. When evaluating the value of vehicle software, the accuracy of the value evaluation can be increased by calculating the proportion of R&D personnel involved in the parts if the software contribution to the price is limited by modularization. The royalty rate estimation formula is as follows [55]. The standard rate is 3%, which is the median car royalty; the utilization rate ranges between 0 and 100%, depending on the software's contribution to the price; the increase/decrease rate considers special factors such as the license situation, and the default value is 100%; and the pioneer rate considers the case where a large cost is required for commercialization, and the default value is 100%.

$$\text{Royalty Rate} = \text{standardrate} \times \text{utilizationrate} \times \text{increase/decreaserate} \times \text{pioneer rate} \tag{2}$$

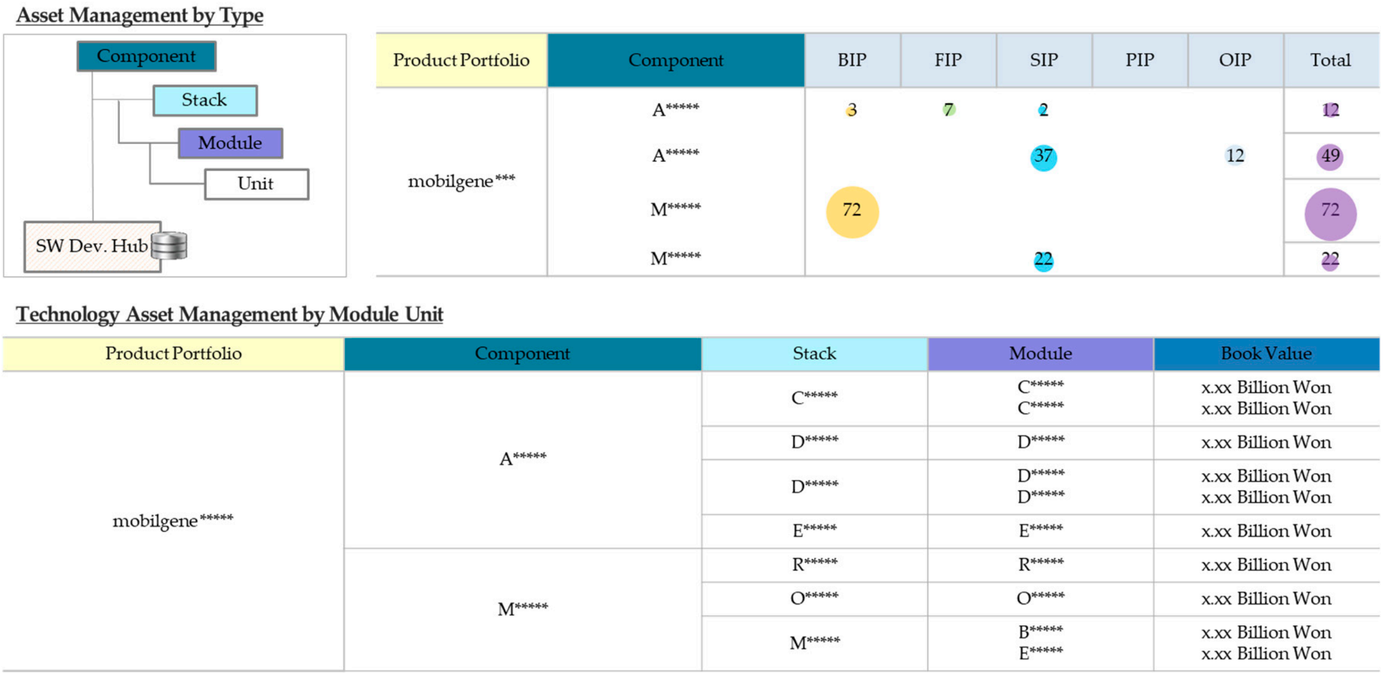
The estimation of software price is carried out through a cost approach, which involves analyzing the software development cost, markup level, and software standard price for the software that a company intends to develop or has already developed. The formula for software standard pricing is given as in [56].

$$\begin{gathered}\text{SW standard price} = \text{SW development cost} \times (1 + \text{markup}) \\ \text{SW development cost} = \text{SW manufacturing cost} + \text{SW nonmanufacturing cost} \\ \text{SW manufacturing cost} = \text{SW development price} - \text{profit(or technology fee)}\end{gathered} \tag{3}$$

To evaluate the markup index for vehicle software, it is necessary to consider additional factors such as the disclosure of source code in the basic module, provision of libraries, verification of reliability for mass production, expected quantity, maintenance period after discontinuation, and factors contributing to the improvement of marketability of the finished product.

3.3. *Software Technology Trading Method*

Currently, Application Software (ASW) and Basic Software (BSW) are developed using a standard process linked to the software integrated development environment for vehicles (referred to as SW Dev Hub), which is a virtual development solution based on digital twin technology to facilitate efficient controller development (Figure 4). To prevent duplicated development, functions that can be shared are classified and structured. The developed technology identifies software assets as BIP, FIP, SIP, PIP, and OIP based on the contribution rate of each company to clearly classify the assets of each company. Software is classified into Component-Stack-Module-Unit (based on the AUTOSAR Guide), technical assets are classified, input cost is calculated, and software assets with tradable technology are listed, enabling the trading of software parts when necessary [57].


Asset Management by Type
Component
Stack
Module
Unit
SW Dev. Hub

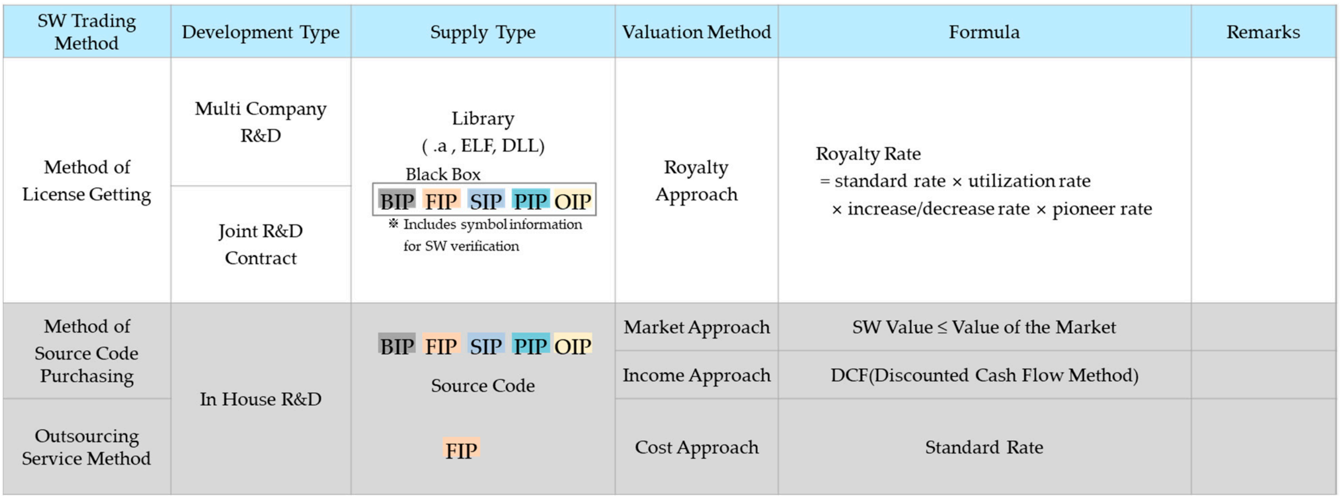
Product Portfolio	Component	BIP	FIP	SIP	PIP	OIP	Total
mobilgene***	A*****	3	7	2			12
	A*****			37		12	49
	M*****	72					72
	M*****			22			22

Technology Asset Management by Module Unit

Product Portfolio	Component	Stack	Module	Book Value
mobilgene*****	A*****	C*****	C***** C*****	x.xx Billion Won x.xx Billion Won
		D*****	D*****	x.xx Billion Won
		D*****	D***** D*****	x.xx Billion Won x.xx Billion Won
		E*****	E*****	x.xx Billion Won
	M*****	R*****	R*****	x.xx Billion Won
		O*****	O*****	x.xx Billion Won
		M*****	B***** E*****	x.xx Billion Won x.xx Billion Won

Figure 4. SW Assets Identification.

A Component refers to the highest level of configuration in the software platform architecture, Stack is a combination of functions that represent the highest structural/functional level, a Module is a logical/structural group/set of units that compose functions for the same purpose, and a Unit is defined as the function level of the minimum functional unit. The income approach, market approach (royalty deduction method), and cost approach are all used in software value calculation through a previously agreed upon valuation model, and reasonable cost estimation is carried out through mutual consultation. All valuation methodologies, including Equations (1)–(3), are performed to ensure the reliability of software technology valuation (Figure 5).



SW Trading Method	Development Type	Supply Type	Valuation Method	Formula	Remarks
Method of License Getting	Multi Company R&D	Library (.a , ELF, DLL) Black Box BIP FIP SIP PIP OIP ※ Includes symbol information for SW verification	Royalty Approach	Royalty Rate = standard rate × utilization rate × increase/decrease rate × pioneer rate	
	Joint R&D Contract				
Method of Source Code Purchasing	In House R&D	BIP FIP SIP PIP OIP Source Code	Market Approach	SW Value ≤ Value of the Market	
			Income Approach	DCF(Discounted Cash Flow Method)	
Outsourcing Service Method		FIP	Cost Approach	Standard Rate	

Figure 5. Type of Providing SW Output.

In case of profit from a technology transaction, compensation is distributed based on the stake. The BIP of each company is provided in a black-box format, and when the source code is disclosed, a weighting factor is added and reflected in the valuation amount for technology evaluation. To facilitate verification, the black-boxed source code is marked with information that allows for reading a specific signal value, which reduces the user's efforts to white-box the software. This approach also helps to protect the software vendor's technical assets.

The HW Dev Hub aims to align the Hyundai Motor Group's development process with the goal of creating a joint software development environment. This involves extracting common technologies from the Group Company SW Structured and identifying the software assets owned by each company through SW Assets Identification. The SW Valuation Methodology is used to evaluate the value of the software owned by each company, and Technology Transfer facilitates the transfer of technology assets between companies (Figure 6).

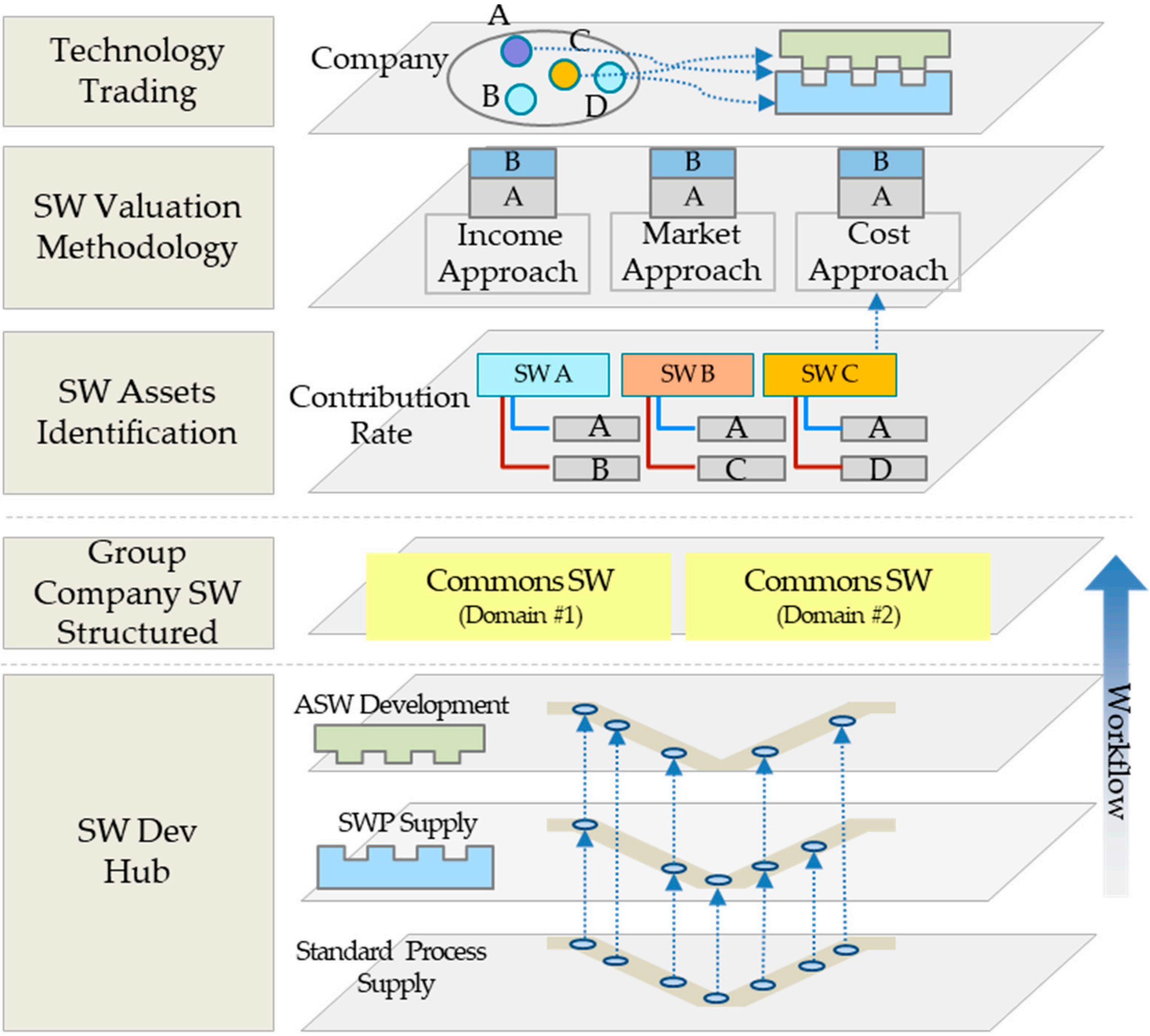


Figure 6. Software co-development concept.

3.4. *Software Supply and Pricing Structure*

If the software is divided into Component, Stack, Module, and Unit and supplied with necessary technologies, product prices can be intuitively compared with competitors, and companies can also gain an advantage in negotiation with their customers by focusing their request on necessary technologies. The high initial capital cost of securing software reduces competitiveness when bidding for contracts. It also raises the risk of return on investment due to the difference between the time of capital investment and the time of recovery, as well as the problem of software price adequacy, which are factors that lead to lower competitiveness.

It is important to "white-box" software that has been "black-boxed" prior to delivery. This involves identifying the technical assets of the software to determine the extent of open-source usage and the company that owns the software assets. It is only possible to trade necessary technologies when the software is identified in module units and the total amount is calculated (Figure 7).

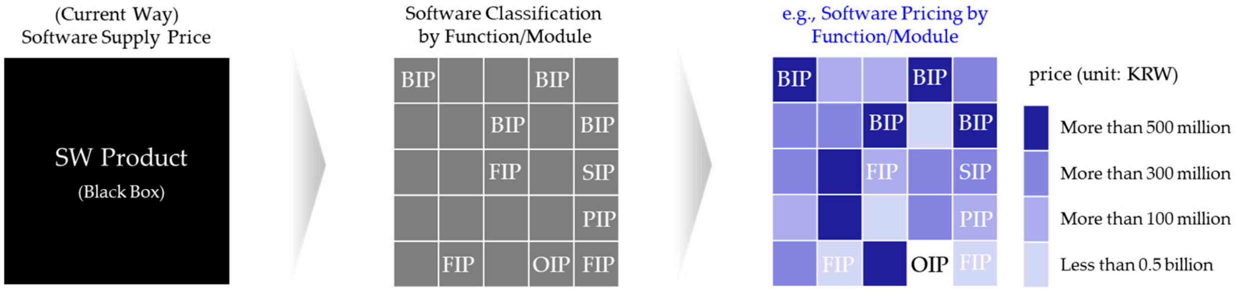


Figure 7. Software co-development concept.

In addition, the preparation of various data for fair price calculation, schedule delays due to price negotiation, and excessive reduction requests without customer standards are important factors that can decrease the competitiveness of software quality. The investment cost for a software module with a high frequency of use is billed as a subscription fee for the mass production period (BIP), and for a software module whose return on investment is unclear due to low frequency of use, a fee is charged at the development stage (FIP) and developed as needed to secure performance. If one module is charged (SIP, PIP) as a subscription fee to the next project through the capitalization process, a win–win relationship can be formed between the software developer and the user.

If a software developer converts a portion of their cash inflow to the time of mass production and proceeds with billing based on vehicle sales performance, software users can be relieved of excessive cost burden (Figure 8). However, this concept will not work unless it is operated in a mutually beneficial attitude based on trust and a culture that recognizes software technical assets between the two companies.

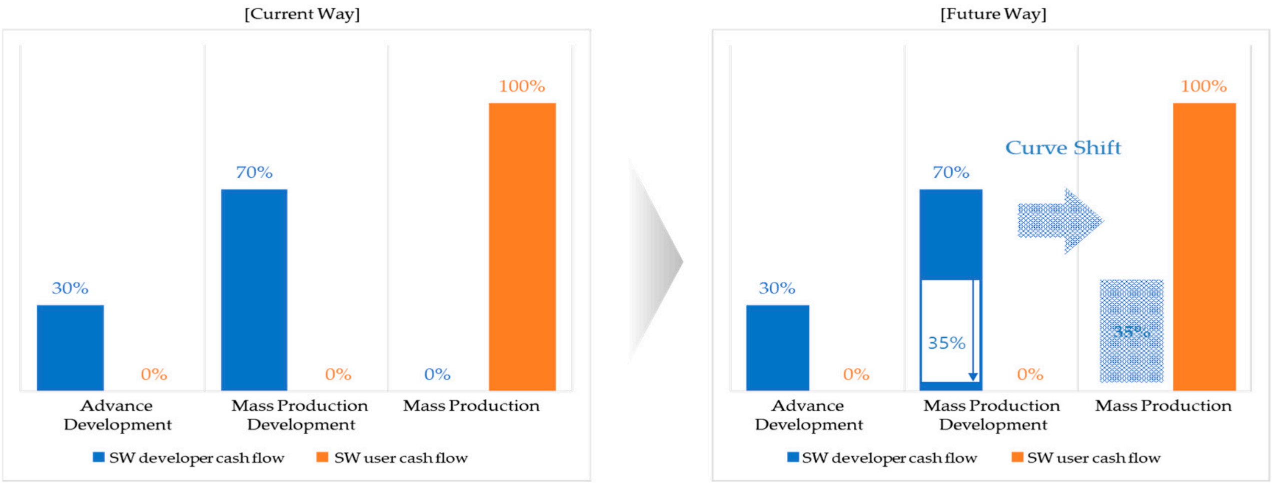


Figure 8. Software billing concept.

At present, BIP, FIP, SIP, PIP, and OIP are developed and supplied without differentiation, but a license policy that considers the separation and development time of each IP should be established and implemented in the future. It is believed that competitiveness in the electronic software field can be secured by establishing a strategic pricing policy that takes into account various supply scenarios, such as supplying each component of the software product according to the customer's request and whether source code is included (Figure 9).

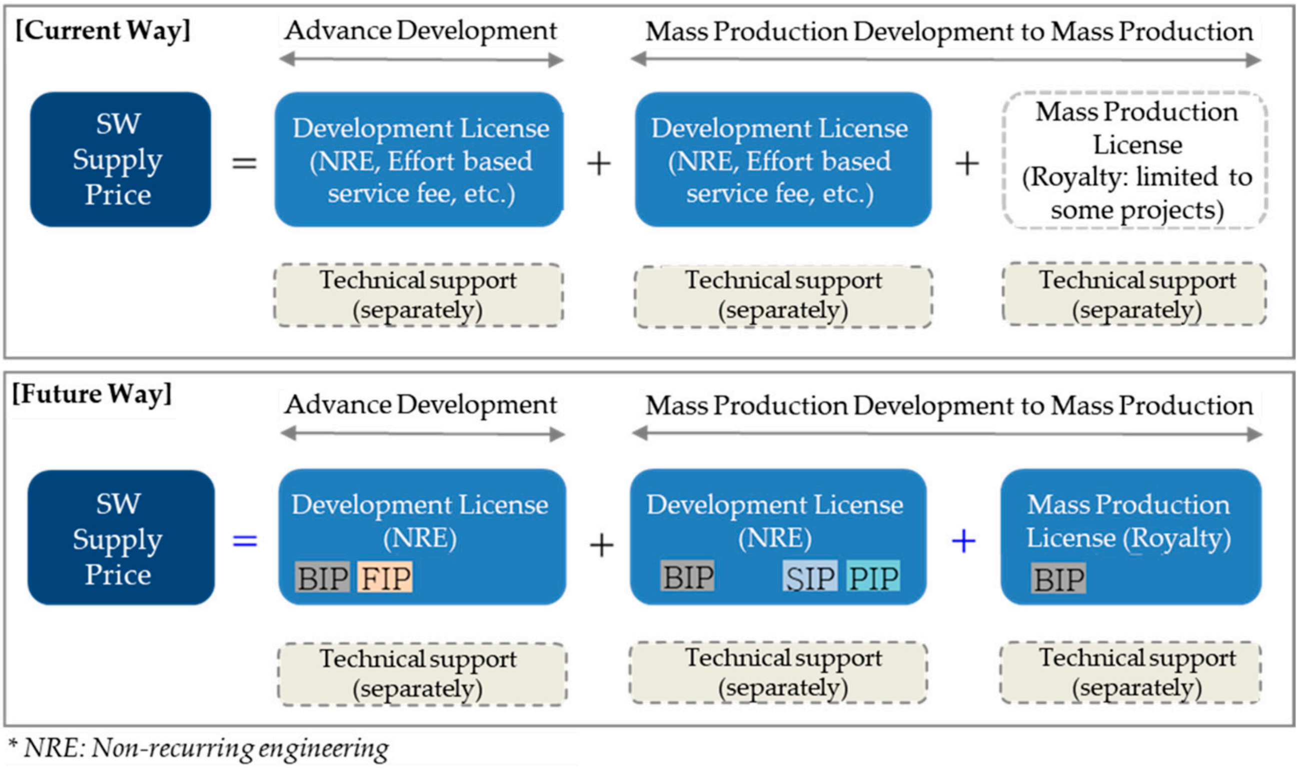
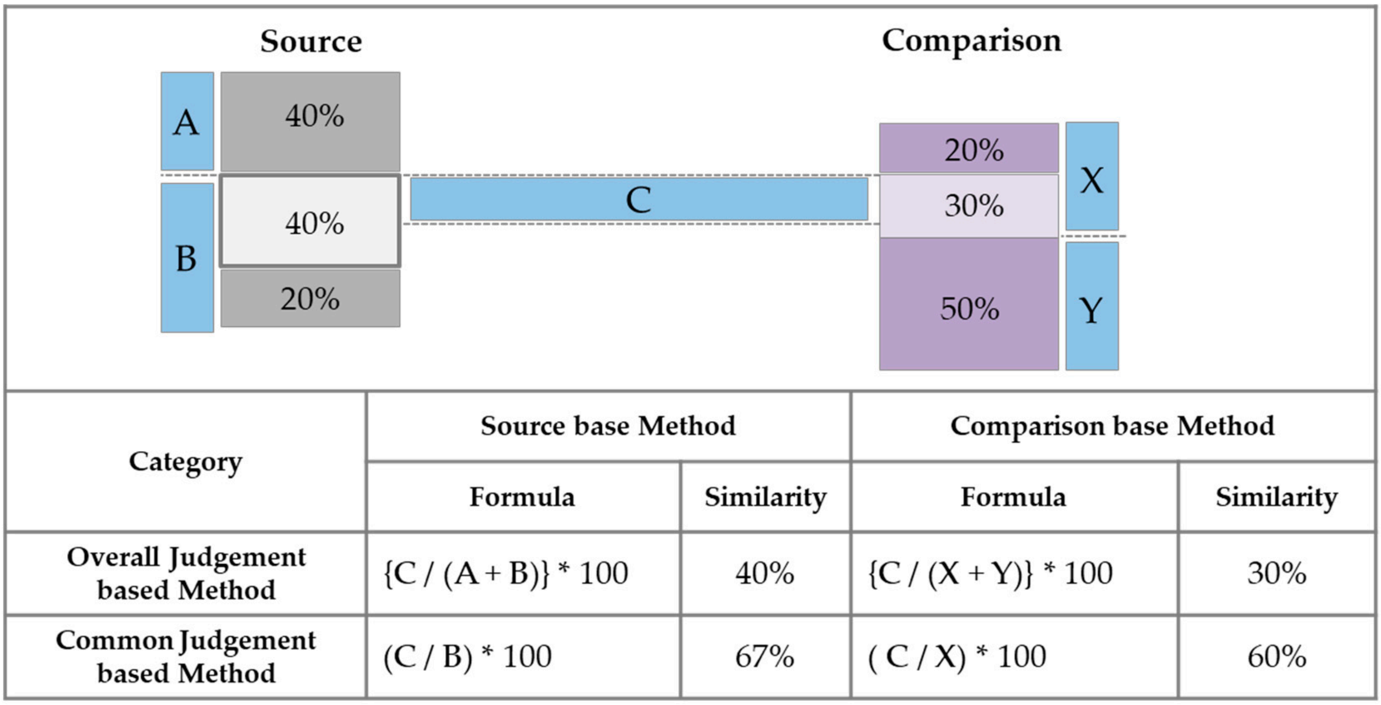

[Current Way]
Advance Development
Mass Production Development to Mass Production
SW Supply Price
=
Development License (NRE, Effort based service fee, etc.)
+
Development License (NRE, Effort based service fee, etc.)
+
Mass Production License (Royalty: limited to some projects)
Technical support (separately)
Technical support (separately)
Technical support (separately)
[Future Way]
Advance Development
Mass Production Development to Mass Production
SW Supply Price
=
Development License (NRE)
BIP
FIP
+
Development License (NRE)
BIP
SIP
PIP
+
Mass Production License (Royalty)
BIP
Technical support (separately)
Technical support (separately)
Technical support (separately)
* NRE: Non-recurring engineering

Figure 9. Software module unit supply concept.

3.5. Method of Determining the Material Similarity of Software Works

As mentioned above, even if the BIP is systematized and specified in the contract, issues of software ownership and copyright infringement by third parties may arise. When determining whether there is substantial similarity between two works to determine copyright infringement, only those corresponding to the creative expression form should be compared. Unauthorized copying of someone else's work infringes on the right to copy, and, in this case, if the original work is not reproduced as it is, new creativity cannot be recognized even if it is modified, increased, decreased, or changed; it is seen as a simple copy.

When judging the similarity of a program work, it is based on practical similarity (objective similarity, structural similarity), and quantitative similarity (Figure 10). When judging the similarity of the source code, similarity can be judged by verifying the similarity between the original and the comparative version. The criteria for determining the level of similarity are as follows: code similarity of 76.06–100% indicates a copyright infringement issue, and 0–49.76% indicates no copyright infringement issue [58].



Category	Source base Method		Comparison base Method	
	Formula	Similarity	Formula	Similarity
Overall Judgement based Method	{C / (A + B)} * 100	40%	{C / (X + Y)} * 100	30%
Common Judgement based Method	(C / B) * 100	67%	(C / X) * 100	60%

Figure 10. Source code similarity review method.

4. Examples of Practical Application Concepts

4.1. Example of Joint Development System

In the automotive industry, incremental and process-oriented innovation is often preferred over radical innovation due to the latter's complexity, low profitability, and high risk [59,60]. This is especially true for a mature and capital-intensive industry with intense competition, where companies strive to focus on their core competencies and prevent the emergence of new competitors, alternative business models, or technologies [61]. However, relying solely on internal technology and R&D capabilities has limitations, making joint development systems crucial for survival in the industry [59]. Recently, new paradigms such as autonomous driving, electric power, and mobility services have emerged, leading to both destructive innovation and structural change. As environmental regulations are strengthened, there is also a push towards electrification of power sources [62].

With the development of electric power for automobiles and autonomous driving technology, new business models and mobility service platforms that utilize them are emerging, and the nature of the automotive industry is changing as the supply chain system is being transformed based on software and electric and electronic technology. An example of this is the software integration concept where several companies collaborate to develop integrated power control devices through open innovation. This necessitates the disclosure of software technology assets owned by each company and the distribution of software revenue.

The feasibility of implementing these software models is limited to the unique corporate culture of the Hyundai Motor Group. The sharing of source code poses significant obstacles and potential hazards. While using similar software may result in cost savings and improved performance, it may also result in the loss of promising technological domains due to compulsory acquisition (Figure 11).

In CASE 1, there was a failure to deliver a requested product for a global automotive OEM. This failure was due to the lack of a concept for establishing SW technology assets.

In contrast, CASE 2 is a success story involving HYUNDAI AUTOEVER's development of Classic Autosar and Adaptive Autosar. These models have been provided to multiple companies and are being used in the development of Advanced Driver Assistance System (ADAS) components.

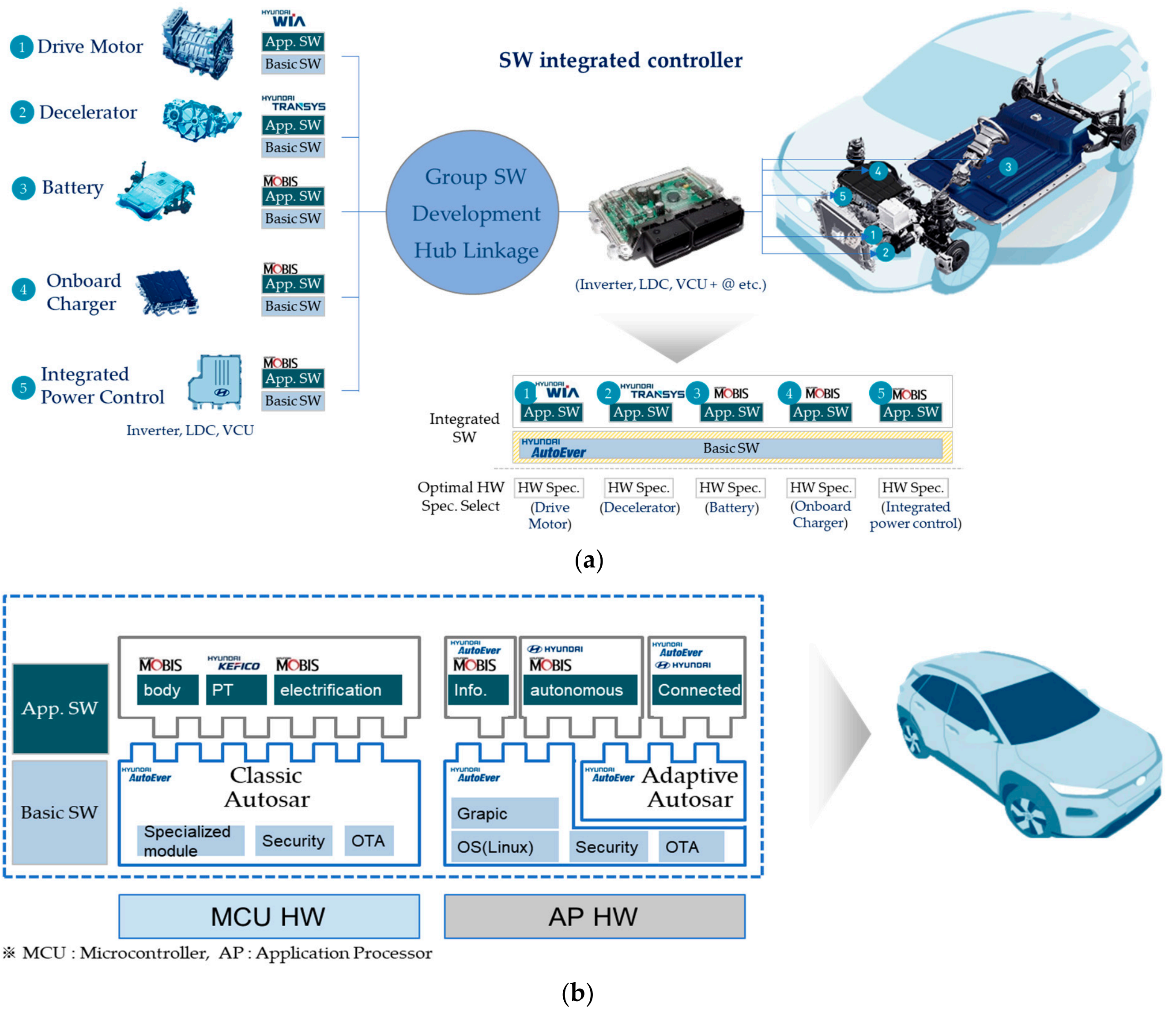


Figure 11. Example of joint development concepts for open innovation (Revised) [63,64]. (**a**) CASE 1. SW integrated controller; (**b**) CASE 2. ADAS +(Advanced Driver Assistance System).

4.2. Example of SW Joint Development

Figure 12 shows that ① Each company identifies the software assets (background IP) they own and ② evaluates its SW value using an income approach, a market approach, and a cost approach based on the development stage. ③ Copyright information for each company is added to the source code, and ④ the code is registered in the SW Dev HUB in the form of inner source.

⑤ Components are downloaded and combined like Lego blocks, a development license is signed, and ⑥ the software is used for development after paying a minimum cost. ⑦ Additional technology is developed as needed, and ⑧ a certain level of compensation is provided for each product after signing the mass production license. ⑨ During the development process, SW verification reports are submitted at the beginning and end of the development, and ⑩ open-source verification tools such as BlackDuck, FossID, and White Source are used to detect copyright information in the source code and confirm ownership. ⑪ The final mass-produced SW undergoes the SW capitalization process, and the assets are identified again and stored in the joint development hub.

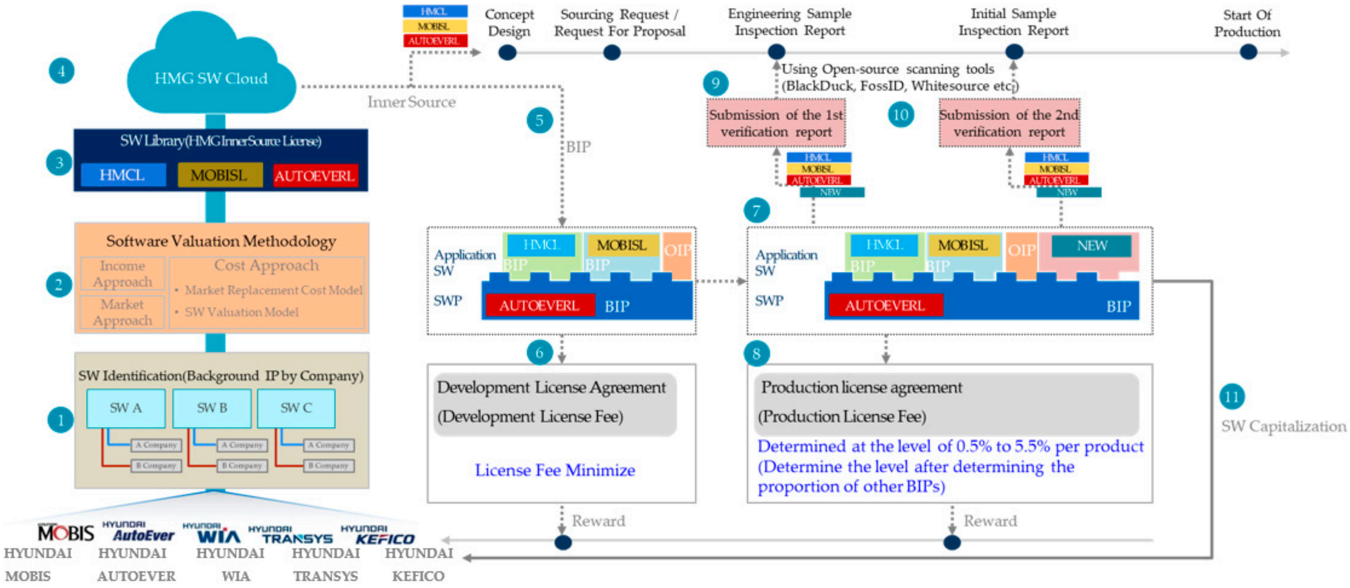

HMG SW Cloud
InnerSource
SW Library(HMG InnerSource License)
HMCL
MOBISL
AUTOEVERL
Software Valuation Methodology
Income Approach
Cost Approach
Market Approach
Market Replacement Cost Model
SW Valuation Model
SW Identification(Background IP by Company)
SW A
SW B
SW C
A Company
B Company
Concept Design
Sourcing Request / Request For Proposal
Engineering Sample Inspection Report
Initial Sample Inspection Report
Start Of Production
Using Open-source scanning tools (BlackDuck, FossID, Whitesource etc)
Submission of the 1st verification report
Submission of the 2nd verification report
BIP
OIP
NEW
Application SW
SWP
Development License Agreement (Development License Fee)
License Fee Minimize
Production license agreement (Production License Fee)
Determined at the level of 0.5% to 5.5% per product (Determine the level after determining the proportion of other BIPs)
Reward
SW Capitalization
HYUNDAI MOBIS
HYUNDAI AUTOEVER
HYUNDAI WIA
HYUNDAI TRANSYS
HYUNDAI KEFICO

Figure 12. Example of SW Joint Development.

4.3. Example of Revenue Sharing Concept

SW Dev HUB operates in the form of open innovation, Figure 13 shows that ① allowing several companies to participate jointly. ② After assigning a weighting factor based on the provision of source code, the percentage of SW for each company is identified to calculate the share ratio and value.

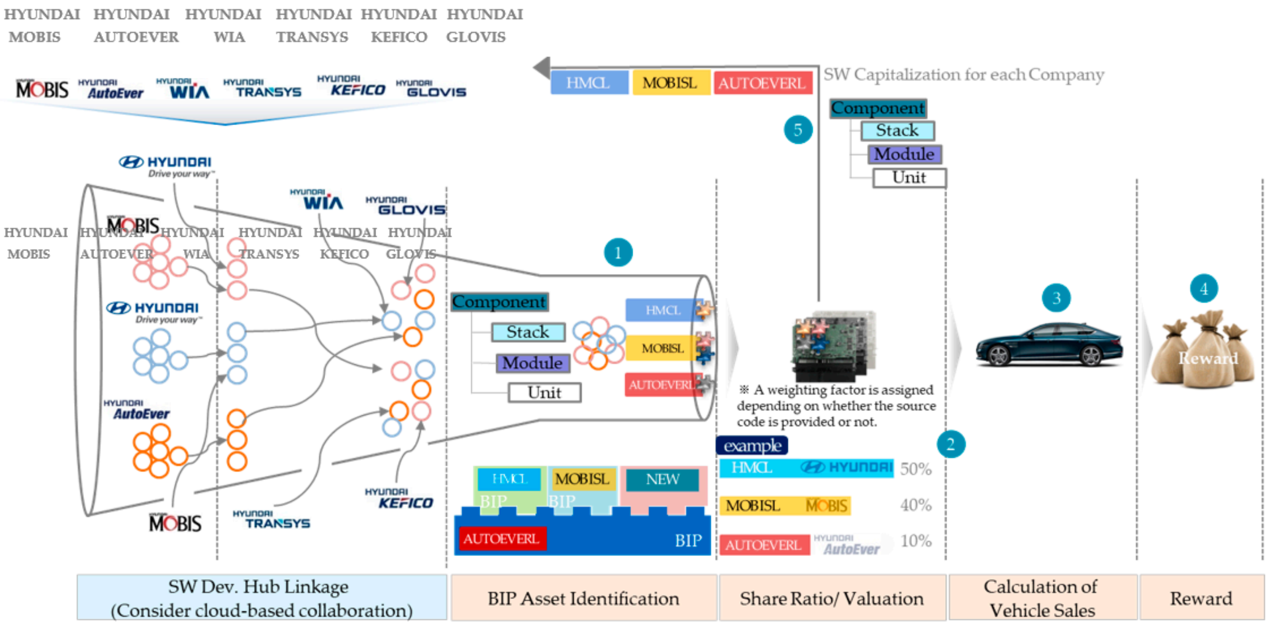

HYUNDAI MOBIS
HYUNDAI AUTOEVER
HYUNDAI WIA
HYUNDAI TRANSYS
HYUNDAI KEFICO
HYUNDAI GLOVIS
HMCL
MOBISL
AUTOEVERL
SW Capitalization for each Company
Component
Stack
Module
Unit
NEW
BIP
※ A weighting factor is assigned depending on whether the source code is provided or not.
example
50%
40%
10%
Reward
SW Dev. Hub Linkage (Consider cloud-based collaboration)
BIP Asset Identification
Share Ratio/ Valuation
Calculation of Vehicle Sales
Reward

Figure 13. Example of revenue sharing concept.

③ The purpose is to maximize the utilization of legacy codes and reduce costs by recycling high-quality SW applied to mass production. ④ Compensation is differentially paid according to the number of vehicles sold later. ⑤ SW assets for each company are tiered into Component, Stack, Module, and Unit, and the SW assets are organized and mass-produced, followed by capitalization. To link the number of vehicles sold and compensation, a reasonable profit distribution concept is presented, considering the positive correlation between the number of vehicles sold and the proportion of SW assets used.

4.4. Analyze the Strengths and Weaknesses of the Proposed Method

Utilizing group-owned software (BSW, ASW) to develop software and share validated software in the automotive development process is an essential software development strategy for achieving global competitiveness. However, because software assets can be mixed, it is necessary to clarify the R&R of each business area and part development for each group company before jointly owning the software. Accurate understanding of the joint ownership software license and the process of continuously updating jointly owned software are also crucial. When multiple companies jointly develop critical intangible assets, ownership of these assets transitions from "sole ownership" to "joint ownership," and royalty policies that were previously operated independently by each company are now applied differently based on the type of joint participation. Therefore, establishing appropriate royalty calculation and ownership distribution standards for royalty income is necessary. Additionally, tax treatment resulting from joint development may also be an issue. Research and development (R&D) is an activity actively promoted at the national level to enhance national competitiveness. As such, the government provides various R&D subsidies and indirectly offers significant tax benefits. One such example is the tax deduction for research and development expenses and personnel development. Therefore, when receiving tax benefits and establishing a new company to conduct business using jointly developed SW technology, it is necessary to systematically identify each company's SW assets and provide sufficient compensation to the existing company.

5. Conclusions

The automotive industry faces significant challenges in terms of sustainability management, including MECA, climate change mitigation, carbon emissions reduction, circular economy in the supply chain, and digital responsibility such as information protection and security. Therefore, it is becoming increasingly important for the automotive industry to establish and implement ESG strategies. MECA is a movement that aims to transform traditional automotive manufacturing companies into Smart Mobility Solution enterprises, by focusing on mobility, electrification, connectivity, and autonomous driving. The foundation of the technology that provides MECA to consumers is software, and in the future automotive mobility industry, where software and hardware are separated and provided, software differentiation will become an important factor in determining the competitiveness of device operations. Moreover, it is imperative to endeavor towards building an API ecosystem in the automotive industry [65].

Securing software technology is critical for enhancing the core competitiveness of the MECA era. To bolster corporate competitiveness, a software value calculation method has been established and implemented, and a technology value calculation method has been developed to determine the suitability of software supply prices and to separate orders for vehicle SW-HW. Three factors are essential for automobile companies to increase software reusability and gain a competitive edge in software development: (1) How to identify software assets? (2) How to calculate software value? (3) How to conduct software technology transactions? In other words, the fundamental concept is to identify tradeable technologies and combine them to quickly develop proven technologies with minimal time investment using existing legacy codes and incentives based on stakes. Failure to change will make it impossible for automobile companies to prevent startups and IT firms from taking on challenges in the automotive sector.

To expand IP-based business and collaborate with global companies, it is necessary to manage and operate IP under classification into BIP, FIP, SIP, PIP, and OIP. The IP concept must also be specified in the contract to prevent legal disputes due to technology asset division in advance. A model for systematically building up BIP through the process of capitalization was presented, and a methodology for value calculation based on technological perfection was discussed.

The software is divided into Component, Stack, Module, and Unit, both structurally and functionally, and a concept for the transaction of legacy code is proposed through a clear classification of IP assets. The concept of billing for cost and the concept of supplying module units were also proposed.

This study presented an open innovation case that utilized an open-source verification tool during the SW verification process to identify each company's copyright information, verify ownership of the source code, and update the SW hub by converting the mass-produced SW into an asset. When multiple companies participate in the joint development of important intangible assets, ownership of the intangible assets is converted from individual ownership to joint ownership, and the royalty policy that was previously managed individually by each company is applied differently, depending on the type of joint participation. It is necessary to calculate an appropriate royalty and establish a standard for the distribution of royalty income among owners in such cases.

The automotive industry is quite broad, and academic alliances are also important. In fact, companies are developing core technologies through investments in universities and research institutes, but problems arise between efforts to assign the technological assets to the company and efforts to prevent them from being taken away.

In future research, it is necessary to think deeply about the tax/ownership problems when switching from single ownership to joint ownership.

Author Contributions: Conceptualization, C.R. and S.D.; methodology, C.R. and S.D.; formal analysis, C.R.; investigation, C.R; writing—original draft preparation, C.R.; writing—review and editing, S.D.; supervision, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Please contact the corresponding author or first author of this paper.

Acknowledgments: The authors wish to thank Ki-Chun Lee, Senior Vice President, Planning, Hyundai Autron, for his inspiration and ideas for the thesis.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Perkins, G.; Murmann, J. What does the success of Tesla mean for the future dynamics in the global automobile sector? *J. Manag. Organ.* **2018**, *14*, 471–480. [CrossRef]
2. Ebert, C.; John, F. Automotive software. *IEEE Softw.* **2017**, *34*, 33–39. [CrossRef]
3. Cisneros, J.R.A.; Fernández-Y-Fernández, C.A.; Vázquez, J.J. Innovation Strategy to Automotive Sector through a Software Development Perspective. In Proceedings of the 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT), San Luis Potosi, Mexico, 24–26 October 2018.
4. Song, Y. Trends of Autonomous Driving and Its Mobility as a Service. *Auto J.* **2018**, *40*, 18–20.
5. Dzienis, A.M.; McCaleb, A. Motives behind Sino-Japanese strategic alliances in the new energy vehicles sector in the age of the Belt and Road Initiative. *Asia Pac. Bus. Rev.* **2022**, 1–26. [CrossRef]
6. Lim, H. Technological Cooperation Network Analysis through Patent Analysis of Autonomous Driving Technology. *J. Korea Acad. -Ind. Coop. Soc.* **2020**, *21*, 688–701. [CrossRef]
7. Monteiro, L.; Mol, M.; Birkinshaw, J. Ready to be open? Explaining the firm level barriers to benefiting from openness to external knowledge. *Long Range Plan* **2017**, *50*, 282–295. [CrossRef]
8. Hoed, R. Sources of radical technological innovation: The emergence of fuel cell technology in the automotive industry. *J. Clean. Prod.* **2007**, *15*, 1014–1021. [CrossRef]
9. Seo, D.; Choi, Y. Strategy for Future Manufacturing in Response to Industrial Paradigm Shift-Focusing on Emerging Sectors of Smart Cars, Fusion & Compound Metal, Healthcare, and Internet of Things. *KIET Res. Rep.* **2020**, *21*, 688–701.
10. Katz, M. An analysis of cooperative research and development. *Rand J. Econ.* **1986**, *17*, 527–543. [CrossRef]
11. Jacquemin, A. Competition and Cooperation between European Companies in Terms of Research and Development. Prix Publics au Luxembourg 1986, pp. 1–16. Available online: http://aei.pitt.edu/41241/ (accessed on 22 March 2023).
12. Rosenzweig, J.; Bartl, M. A Review and Analysis of Literature on Autonomous Driving. *E-J. Mak.-Innov.* **2015**, 1–57.
13. Fletcher, R.; Mahindroo, A. The Case for An End-to-End Automotive-Software Platform. *McKinsey & Company*. 2020. Available online: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform#/ (accessed on 19 March 2023).

14. Yun, H.; Kim, S.; Lee, J.; Yang, J. Analysis of Case of Disengagement Based of U.S. Califonia DMV Autonomous Driving Disengagement Report. *KSAE* **2018**, *26*, 464–475. [CrossRef]
15. Park, M.; Son, J. Reference Test Scenarios for Assessing the Safety of Take-over in a Conditional Autonomous Vehicle. *KSAE* **2019**, *27*, 309–317. [CrossRef]
16. Saidi, S.; Steinhorst, S.; Hamann, A.; Ziegenbein, D.; Wolf, M. Future automotive systems design: Research challenges and opportunities: Special session. In Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis, Turin, Italy, 30 September–5 October 2018.
17. Nyberg, M.; Gurov, D.; Lidström, C.; Rasmusson, A.; Westman, J. Formal verification in automotive industry: Enablers and obstacles. In Proceedings of the International Symposium on Leveraging Applications of Formal Methods, Limassol, Cyprus, 5–9 November 2018.
18. Sandu, I.A.; Salceanu, A. Improved Technique for Measuring the Number of Defects in Automotive Agile SW Development: Defect debt trend. In Proceedings of the 2018 International Conference and Exposition on Electrical and Power Engineering (EPE), Iasi, Romania, 18–19 October 2018.
19. Pawlik, E.; Ijomah, W.; Corney, J.; Powell, D. Exploring the application of lean best practices in remanufacturing: Empirical insights into the benefits and barriers. *Sustainability* **2021**, *14*, 149. [CrossRef]
20. Han, W.; Huang, Y.; Hughes, M.; Zhang, M. The trade-off between trust and distrust in supply chain collaboration. *Ind* **2021**, *98*, 93–104. [CrossRef]
21. Rothstein, J. Reframing the "Domestic" Auto Industry: The Decline of the Detroit Automakers and the Rise of a Diversified and Geographically Dispersed Industry. *Contemp. Sociol.* **2020**, *49*, 402–406. [CrossRef]
22. Bogers, M.; Chesbrough, H.; Heaton, S.; Teece, D.J. Strategic management of open innovation: A dynamic capabilities perspective. *Calif. Manag. Rev.* **2019**, *62*, 77–94. [CrossRef]
23. Gawer, A.; Cusumano, M.A. Industry platforms and ecosystem innovation. *J. Prod. Innov. Manag.* **2014**, *31*, 417–433. [CrossRef]
24. Vdovic, H.; Babic, J.; Podobnik, V. Automotive software in connected and autonomous electric vehicles: A review. *IEEE Access* **2019**, *7*, 166365–166379. [CrossRef]
25. Steinbach, T. Ethernet-based network architectures for future real-time systems in the car. *ATZ Worldw.* **2019**, *121*, 72–77. [CrossRef]
26. Chen, S.; Chen, Y.; Zhang, S. A novel integrated simulation and testing platform for self-driving cars with hardware in the loop. *IEEE Trans. Intell. Veh.* **2019**, *4*, 425–436. [CrossRef]
27. Chi, C.F.; Sigmund, D.; Astardi, M.O. Classification scheme for root cause and failure modes and effects analysis (FMEA) of passenger vehicle recalls. *Reliab. Eng. Syst. Saf.* **2020**, *200*, 106929. [CrossRef]
28. Wu, W.; Li, R.; Xie, G.; An, J.; Bai, Y.; Zhou, J.; Li, K. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 919–933. [CrossRef]
29. Bae, H. Current Status and Development Direction of SW for Vehicle. *KESSIA Issue Report*. 2016. Available online: https://www.kessia.kr/bbs/board.php?tbl=memberwork&mode=VIEW&num=124&chr=&category=&findType=&findWord=&sort1=&sort2=&page=4 (accessed on 8 January 2020).
30. Park, J.; Choi, B.W. Design and implementation procedure for an advanced driver assistance system based on an open source AUTOSAR. *Electronics* **2019**, *8*, 1025. [CrossRef]
31. Jose, A.; Susan, R.J.; Roshin, M. Design, development and testing of AUTOSAR TCP/IP stack: UDP protocol. In Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2020 (ICCMSE-2020), Crete, Greece, 29 April–3 May 2020.
32. Rahman, K.M.; Alam, C.M. Japanese Automobile Industry and Its Supply Chain Management Condition during Corona Crisis Period. *J. Manag. Res.* **2022**, *37*, 111–131. [CrossRef]
33. Navale, V.; Williams, K.; Lagospiris, A.; Schaffert, M. (R)evolution of E/E architectures. *SAE Int. J. Passeng. Cars–Electron. Electr. Syst.* **2015**, *8*, 282–288. [CrossRef]
34. Ruhe, G.; Saliu, M. The art and science of software release planning. *IEEE Softw.* **2005**, *22*, 47–53. [CrossRef]
35. Zhang, Y. Innovation Dynamics between Original Equipment Manufacturers (OEMs) and Tier-1 Suppliers in the Automotive Industry. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, May 2022.
36. Cariad. Available online: https://cariad.technology (accessed on 4 March 2023).
37. Toyota Research Institute-Advanced Development, Inc. (TRI-AD). Announces It Will Expand and Improve Its Operations by Forming Woven Planet Holdings and Two New Operating Companies, Woven Core and Woven Alpha. Available online: https://global.toyota/en/newsroom/corporate/33786527.html (accessed on 4 March 2023).
38. Back from Software to Staffers, Tesla and SpaceX Share More than Elon Musk. Available online: https://www.livemint.com/Companies/UlqhKUHvgeRs9TILIc3ulI/From-software-to-staffers-Tesla-and-SpaceX-share-more-than.html (accessed on 4 March 2023).
39. Lim, S.; Kim, S.; Park, H. A Study on a Conceptual Model for Technology Valuation Based on Market Approach. *J. Korea Technol. Innov. Soc.* **2015**, *18*, 204–231.
40. Masuda, Y.; Whang, S. Dynamic Pricing for Network Service: Equilibrium and Stability. *Manag. Sci* **1999**, *45*, 857–869. [CrossRef]
41. Aboody, D.; Lev, B. The Value Relevance of Intangibles: The Case of Software Capitalization. *J. Account. Res.* **1998**, *36*, 161–191. [CrossRef]

42. Dasgupta, P.; Das, R. Dynamic Pricing with Limited Competitor Information in a Multi-Agent Economy. *Lect. Notes Comput. Sci.* **2000**, *1901*, 299–310. [CrossRef]
43. Baldwin, C.; Clark, K. Between "Knowledge" and "the Economy": Notes on the Scientific Study of Designs. *J. Knowl. Econ.* **2006**, 299–328.
44. TUD, G.F.; Koszescha, J.; Neyer, M.; Cogez, P.; NXP, P.P.; Wambacq, P.; TUD, V.R. European Core Technologies for future connectivity systems and components. *COREnect* **2022**.
45. Daum, T. Agile Methods on the Shop Floor: Towards a" Tesla Production System"? *SSOAR* **2022**. [CrossRef]
46. Jie, Y.; He, H.; Xing, K.; Yue, A.; Tan, W.; Yue, C.; Jiang, C.; Chen, X. MECA-Net: A MultiScale Feature Encoding and Long-Range Context-Aware Network for Road Extraction from Remote Sensing Images. *Remote Sens. Environ.* **2022**, *14*, 5342. [CrossRef]
47. Granstrand, O.; Holgersson, M. The challenge of closing open innovation: The intellectual property disassembly problem. *Res. Technol. Manag.* **2014**, *57*, 19–25. [CrossRef]
48. Bengtsson, M.; Kock, S. Coopetition in Business Networks—To Cooperate and Compete Simultaneously. *Ind. Mark. Manag.* **2000**, *29*, 411–426. [CrossRef]
49. Ritala, P.; Hurmelinna-Laukkanen, P. Incremental and Radical Innovation in Coopetition—The Role of Absorptive Capacity and Appropriability. *J. Prod. Innov. Manag.* **2013**, *30*, 154–169. [CrossRef]
50. Arora, A.; Athreye, S.; Huang, C. The paradox of openness revisited: Collaborative innovation and patenting by UK innovators. *Res. Policy* **2016**, *45*, 1352–1361. [CrossRef]
51. Granstrand, O.; Holgersson, M. Managing the intellectual property disassembly problem. *Calif. Manag. Rev.* **2013**, *55*, 184–210. [CrossRef]
52. Grossman, S.; Hart, O. The costs and benefits of ownership: A theory of vertical and lateral integration. *J. Political Econ.* **1986**, *94*, 691–719. [CrossRef]
53. *Technology Valuation Practice Guide*, 2nd ed.; Korea Institute for Advancement of Technology: Seoul, Republic of Korea, 2021; pp. 7–158.
54. Ozkaya, I.; Kazman, R.; Klein, M. Quality-Attribute Based Economic Valuation of Architectural Patterns. In Proceedings of the 2007 First International Workshop on the Economics of Software and Computation, Minneapolis, MI, USA, 20–26 May 2007.
55. Cho, K.; Lim, J. *Intellectual Property Finance and Valuation Practice*, 1st ed.; Korea Banking Institute: Seoul, Republic of Korea, 2014; pp. 125–135.
56. Seong, T.; Kim, S.; Park, H. Development of a reference model for calculating the appropriate software price for software technology value evaluation and case verification analysis. In Proceedings of the Conference of the Korean Society of Technology Innovation, Jeju, Republic of Korea, 6–7 November 2015.
57. Jayan, J.; Srinivasan, G. AUTOSAR Based Dual Core Partitioning for Power Train Application of BMS. In Proceedings of the 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputhur, India, 11–13 April 2019.
58. Darae Law&IP Group. Available online: https://www.daraelaw.co.kr/kr/news/forum.do?SQ=503&CODE=060301&PG=10&pageSize=10&pageNo=1 (accessed on 4 January 2022).
59. Cammarano, A.; Michelino, F.; Caputo, M. Open innovation practices for knowledge acquisition and their effects on innovation output. *Technol. Anal. Strateg. Manag.* **2019**, *31*, 1297–1313. [CrossRef]
60. Radnejad, A.B.; Vredenburg, H. Disruptive technological process innovation in a process-oriented industry: A case study. *JET-M* **2019**, *53*, 63–79. [CrossRef]
61. Harvie, C. *Micro-, Small-and Medium-Sized Enterprises (MSMEs): Challenges, Opportunities and Sustainability in East Asia*; Palgrave Macmillan: Singapore, 2019; pp. 155–174.
62. Yigitcanlar, T.; Wilson, M.; Kamruzzaman, M. Disruptive impacts of automated driving systems on the built environment and land use: An urban planner's perspective. *J. Open Innov. Technol. Mark. Complex.* **2019**, *5*, 24. [CrossRef]
63. Hyundai Motors Group. Available online: https://www.hyundai.co.kr/story/CONT0000000000000906 (accessed on 19 March 2023).
64. Eugene Investment. Available online: https://www.eugenefn.com/common/files/amail//20220304_307950_leejaeil_873.pdf (accessed on 19 March 2023).
65. Bondel, G.; Nägele, S.; Koch, F.; Matthes, F. Barriers to the Advancement of an API Economy in the German Automotive Industry and Potential Measures to Overcome these Barriers. In Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020), Online Streaming, Prague, Czech Republic, 5–7 May 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.