



CONCEPT OF IDEAL SOFTWARE AND ITS REALIZATION SCENARIOS

Dr. P. S. Aithal* & T. Vaikunth Pai**

Srinivas Institute of Management Studies, Pandeshwar, Mangalore,
Karnataka

Abstract:

Ideal software, by definition, is a general purpose software model which can be used for any platform, any type of system, and application automation, without making modifications in the form of structure, coding by an external person/agency. Ideal software is a hypothetical system/instruction set, supported by artificial intelligence, optical neural networks & optical solutions, to modify the algorithms, instructions, and coding based on data processing requirements. In this paper, the characteristics of such hypothetical software called ideal software, and possibility of using such software for any kind of system automation and networking called Universal Automation are discussed. The paper also contains the possibility of development of such software using presently existing and future available technologies and consequences of such software development on IT industry.

Index Terms: Ideal Software, Artificial Intelligence, Optical Neural Networks & Universal Automation

1. Introduction:

Automation of business processes improve an organization's overall workflow in terms of simplifying and improving the process, achieving greater efficiency, adapting to changing business needs, reducing human error, speedup the activities of processes, reduce the cost associated with completing particular process, and clarifying job roles and responsibilities. Automation of business process need software designed for that specific purpose. Developing software for various business processes is greater challenge and massive work and many organizations involved in software planning, design and development. Presently, the two broader classifications of software are general purpose and specific purpose application software. In this paper, we have considered the software as a model of a system. A model is a simplified representation of an operation, or is a process, or a system in which only the basic aspects or the most important features of a typical problem under investigation are considered. The objective of a model is to identify significant factors and interrelationships. The reliability of the solution obtained from a model, depends on the validity of the model representing the real system. A good model called ideal model must have the following characteristics:

- ✓ An ideal model should be capable of taking new formulations in to account without having any changes in its frame.
- ✓ Assumptions made in the model should be as small as possible.
- ✓ Variables used in the model must be less in number ensuring that it is simple and coherent.
- ✓ It should be open to parametric type of treatment.
- ✓ It should not take much time in its construction for any problem.

The significant advantages of using a model for a system are:

- ✓ Problems under consideration become controllable through a model.
- ✓ It provides a logical and systematic approach to the problem.
- ✓ It provides the limitations and scope of an activity of the system.
- ✓ It helps in finding useful tools that eliminate duplication of methods applied to

solve the problems.

- ✓ It helps in finding solutions for research and improvements in a system.
- ✓ It provides an economic description and explanation of either the operation or the systems they represent.

Predictive Models predict facts and relationships among the various activities of the problem. These models do not have an objective function as a part of the model to evaluate decision alternatives. In this model, it is possible to get information as to how one or more factors change as a result of changes in other factors. Example of predictive model is the model of an Ideal system.

Descriptive Models describe facts and relationships among the various activities of the problem. These models also do not have an objective function as a part of the model to evaluate decision alternatives. In this model also, it is possible to get information as to how one or more factors change as a result of changes in other factors. Example of descriptive model is the model of a real system.

2. About Ideal Systems:

It is well known that we can improve the performance of any system by comparing it with a hypothetical, predicted system of that kind called "Ideal system" [1]. Ideal properties of a device or a system can be used to upgrade or improve its properties towards reaching 100% efficiency. By comparing the properties/characteristics of a practical device/system with its ideal counterpart, one can find out the possible modifications in that device /system towards reaching the objective of achieving such an ideal system [2]. Many systems like ideal gas, ideal fuel, ideal solution, ideal fluid, ideal engine, ideal switch, ideal voltage source, ideal current source, ideal diode, ideal transistor, and ideal amplifier are familiar to everybody since school days. Recently, ideal business system [2-3], ideal education system [4-6], ideal technology system [1], ideal strategy [7], ideal energy source [8], ideal banking system [9], and ideal library system [10] are studied and their input, system, output and environmental characteristics are discussed.

Ideal software, by definition, is a general purpose software model which can be used for any platform, any type of system, and application automation, without making modifications in the form of structure, coding by an external person/agency. Ideal software is a hypothetical system/instruction set, supported by artificial intelligence, optical neural networks & optical solutions, to modify the algorithms, instructions, and coding based on data processing requirements. Ideal Software is a predictive model and is hypothetical software product having ideal characteristics and is the ultimate goal of a software developer. Identifying ideal software characteristics give an idea on how present software can be improved further by knowing the gap between present software and ideal software. The concept of ideal software motivates software designers to continuously improve the software towards ideal. Software is a system which consists of set of instructions used to do certain pre-specified operations on input data and gives output in the form of result/decision. Thus, generally, software is a product with set of inputs, various set of instructions and execution of these instructions as processes and defined format of output which are results of the processed instructions on input data. A general software system is given in the figure 1. While developing any kind of software product, the first and foremost objective is that it must meet all the requirements of the customer or end-user. Secondly, the cost of developing and maintaining the software should be low and the development of software should be completed in the specified time-frame. Presently, the software design and development is a major business in worldwide and there are enough

opportunities and challenges in starting software companies to improve the quality of the software [11].



Figure 1: Block diagram of a general software system

3. Model and Characteristics of Ideal Software:

3.1 Ideal Software Model:

Ideal Software System consists of various characteristics under 4 categories such as Input conditions, Systems requirements, Output conditions and Environmental & social constraints, and analyzed these characteristics with an objective to achieve the goal of achieving universal automation system. Idealsoftware would not only automate the various industrial and business processes but would also improve the efficiency of the processes towards maximum value (infinity) to better suit the needs of the people. An idealsoftware shall have characteristics which can be predicted and classified. Based on various factors which decide the ideal software characteristics, a model consisting of the input conditions, output conditions, system requirements, and social & environmental constraints is derived by a qualitative data collection instrument namely, focus group method [3, 4]. The block diagram of such a system is shown in fig. 2.

Input: The various resources and qualities both tangible and intangible consumed while developing and using the software for the automation of various functions will come under input characteristics.

Processes: The various software system characteristics like operational characteristics, transition characteristics, revision Characteristics, performance characteristics and maintenance characteristics. For ideal system, these characteristics should lead to the corresponding parameters either zero or infinity depending on making the system ideal.

Output: The output characteristics like performance in achieving the goal are accuracy, correctness, reliability, sustainability, reusability, efficiency and readability.

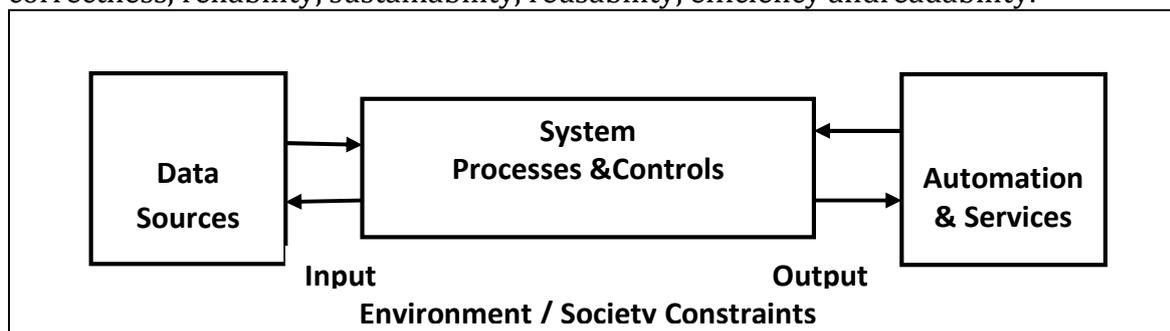


Figure 2: Characteristics of an Ideal Software System

3.2 Ideal Software Characteristics:

The Quality factors of software can be determined using their input characteristics, operational characteristics, transition characteristics, revision characteristics and output characteristics. These characteristics are obvious and essential features expected from any project during development and implementation.

3.2.1 Input Characteristics:

The input characteristics of ideal software are listed in table 1. The prominent eight input characteristics are (1) Zero input resources, (2) Infinite selectivity, (3)

Ubiquitous input acceptance, (4) Infinite input security, (5) Infinite reliability, (6) Infinite usability, (7) Infinite efficiency in data acceptance, and (8) Zero energy consumption at input.

Table 1: Description of various input characteristics of an ideal software

S.No	Input characteristics	Description
1	Minimum Resources	Zero input resources except data & Instructions
2	Minimum Input	Minimum amount of data & Instructions as input
3	Input Selectivity	Select input data which is appropriate for process and eliminate redundancy and inconsistency
4	Ubiquitous input	Ubiquitous input data (Any time, any amount, any amount of time)
5	Security	With the increase in security threats nowadays, this factor is gaining importance. The input shouldn't have ill effects on software / hardware.
6	Reliability	The input should not have any defects. Not only this, it shouldn't fail while execution. Zero defects & errors prone data
7	Usability	Infinite usability of inputs
8	Efficiency	100% efficiency of data for getting accurate results
9	Short processing time	Select input which takes minimum time for processing
10	No input energy	Zero input energy for Processing & Storing and thus maximize performance of system

3.2.2 Software Process/ System Characteristics:

System characteristics include operational characteristics, transitional characteristics and maintenance characteristics [12].

(A) Operational Characteristics: These are functionality based factors and related to 'exterior quality' of software and this tells us how well software works in operations. It can be measured on following characteristics that are listed in Table 2. The ten prominent operational characteristics are (1) Zero budgets, (2) Full correctness, (3) Easy usability, (4) Perfect integrity, (5) 100% reliability, (6) 100% efficiency, (7) Infinite tolerance to security threats, (8) 100% safety against hazards, (9) Infinite functionality, (10) Perfect Robustness.

Table 2: Description of various operational characteristics of an ideal software

S.No	Operational Characteristics	Description
1	Budget	Determines the cost and difficulty/ease to develop and maintain an software
2	Correctness	The software which we are making should meet all the specifications stated by the customer.
3	Usability	The amount of efforts or time required to learn how to use the software should be less. This makes the software user-friendly even for IT-illiterate people.
4	Integrity	Refers to the degree to which unauthorized access to the software or data can be prevented.
5	Reliability	The software product should not have any defects.

		Not only this, it shouldn't fail while execution.
6	Efficiency	This characteristic relates to the way software uses the available resources. The software should make effective use of the storage space and execute command as per desired timing requirements.
7	Security	With the increase in security threats nowadays, this factor is gaining importance. The software shouldn't have ill effects on data / hardware. Proper measures should be taken to keep data secure from external threats.
8	Safety	The software should not be hazardous to the environment/life.
9	Functionality	The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions (what the software does to fulfill needs).
10	Robustness	The degree to which the software can keep on functioning in spite of being provided with invalid data.

(B) Transitional Characteristics: Importance of any of these factors varies from application to application. In systems where human life is at stake, integrity and reliability factors must be given prime importance. In any business related application usability and maintainability are key factors to be considered. Always remember in Software Engineering, quality of software is everything, therefore try to deliver a product which has all these characteristics and qualities. This aspect is important when the software is moved from one platform to another and various transition Characteristics of software are listed in Table 3. The four most prominent transaction characteristics of ideal software are (1) Perfect interoperability, (2) 100% reusability, (3) perfect portability, and (4) 100% performance guarantee.

Table 3: Description of various transitional characteristics of an ideal software

S.No	Transitional Characteristics	Description
1	Interoperability	Interoperability is the ability of software to exchange information with other applications and make use of information transparently.
2	Reusability	If we are able to use the software code with some modifications for different purpose then we call software to be reusable.
3	Portability	Portability refers to the ability of an application to run on different platforms (operating systems) with or without minimal changes. Due to rapid development in the hardware and the software, nowadays platform change is a common phenomenon. Hence, if a program is developed for a particular platform, then the life span of the program is severely affected.
4	Performance	Software should perform reasonably well in regular and stressful scenarios.

(C) Maintenance Characteristics: This aspect briefs about how well software has the capabilities to maintain itself in the ever-changing environment and various maintenance characteristics of software are listed in table 4. The ten most prominent maintenance characteristics of ideal software are (1) Zero maintenance cost, (2) Perfect flexibility, (3) Perfect generality, (4) Infinite extensibility, (5) Infinite scalability, (6) Easy testability, (7) Highest modularity, (8) Best readability, (9) Easy documentation for anybody use, (10) Infinite tenant efficiency, and (11) Easy configurability.

Table 4: Description of various maintenance characteristics of an ideal software

S.No	Maintenance Characteristics	Description
1	Maintainability	Maintenance of the software should be easy for any kind of user.
2	Flexibility	Software should be flexible enough to handle most of the changes without having to rewrite the entire program. Most of the programs are developed for a certain period and they require modifications from time to time. For example, in case of payroll management, as the time progresses, some employees may leave the company while some others may join. Hence, the payroll application should be flexible enough to incorporate all the changes without having to reconstruct the entire application.
3	Generality	Apart from flexibility, the software should also be general. Generality means that if software is developed for a particular task, then it should also be used for all similar tasks of the same domain. For example, if software is developed for a particular organization, then it should suit all the other similar organizations.
4	Extensibility	It should be easy to increase the functions performed by it.
5	Scalability	It should be very easy to upgrade it for more work (or for more number of users).
6	Testability	Testing the software should be easy.
7	Modularity	Any software is said to make of units and modules which are independent of each other. These modules are then integrated to make the final software. If the software is divided into separate independent parts that can be modified, tested separately, it has high modularity.
8	Readability	The program should be written in such a way that it makes other programmers or users to follow the logic of the program without much effort. If a program is written structurally, it helps the programmers to understand their own program in a better way. Even if some computational efficiency needs to be sacrificed for better readability, it is advisable to use a more user-friendly approach, unless the processing of an application is of utmost importance.

9	Documentation	Documentation is one of the most important components of an application development. Even if a program is developed following the best programming practices, it will be rendered useless if the end user is not able to fully utilize the functionality of the application. A well-documented application is also useful for other programmers because even in the absence of the author, they can understand it.
10	Multi-Tenant Efficiency	All the sites share the same instance of the software, providing a significant savings in server resource use, maintenance, and costs. Updates are rolled out instantly to customers, all at once.
11	Configurability	Each site's administration interface has variable configuration options including: network content access, simplified blog interface, Social Site (social media) extensions, and custom tool modules. These can be turned on or off on a per site basis with minimal effort and time.

3.2.3. Output Characteristics:

The output characteristics of ideal software are listed in table 5. The prominent eight output characteristics are (1) 100% accuracy, (2) Perfectly correct output, (3) Perfectly reliable output, (4) Long term sustainability, (5) Infinitely reusability, (6) 100% output efficiency at very low input, (7) Readability to everyone, and (8) Perfect satisfied user experience.

Table 5: Description of various output characteristics of an ideal software

S.No	Output Characteristics	Description
1	Accuracy	Software should result in high accuracy output.
2	Correctness	Software result should meet all the requirements of end users.
3	Reliability	The software output should not have any defects which misleads in decision making.
4	Sustainability	Long period of sustainability.
5	Reusability	Same output can be able to take different inferences for decision making.
6	Efficiency	More output with the same input and Same output with less input
7	Readability	Result of software should be in simple readable form so that anyone can understand.
8	User Experience	Its user interface would immediately make sense to its users. It would be simplistic, elegant and something that is painless to use.

3.2.4 Environmental Characteristics:

The external environmental and market characteristics of ideal software are listed in table 6. The most prominent external characteristics are (1) Inelastic demand, (2) Infinite market for ideal software, (3) Infinite ability, (4) Cannot be copied by others/competitors, and (5) High quality service to every user.

Table 6: Description of various environmental characteristics of an ideal software

S.No	Environmental Characteristics	Description
1	Inelastic Demand	Inelastic demand refers to a software system that people need or desire almost at any price.
2	Infinite market	Unlimited global market due to its unique character of providing service to every automation systems
3	Infinite ability to process inelastic demand	Ideal software has infinite ability to process inelastic demand from entire world due to its infinite capacity.
4	Cannot be copied	Ideal software cannot be copied by competitors. It can be protected by copyright /patent so that the developer of ideal software can enjoy its proprietary for long time.
5	High quality service to every user	Ideal software is expected to provide high quality service to every user so that it provides 100% satisfaction to external users.

4. Ideal Software & Universal Automation:

Ideal software has infinite internal ability to automate any processes, applications, and systems. Ideal software system can automate industrial production systems, business service systems, form systems and even home applications. Such fully automated systems are possible using software systems having ideal characteristics developed with the help of artificial intelligence, optical neural networks & optical solutions, to modify the algorithms, instructions, and coding based on data processing requirements. Such automation system called universal automation systems can automate all processes in the society leaving the human beings free from work and challenges. Even though, developing complete ideal software is only the prediction, using technological advancement, one can develop software which can mimic the ideal characteristics in future days which leads further development of anticipated universal automation systems [13-16].

5. Possibility of Development of Ideal Software:

It is interesting to know how one can develop software which shows ideal characteristics. The possibility of development of such software using presently existing and future available technologies is a real challenge for software professionals and also the consequences of such software development on IT industry in the society.

Operating Systems:

- ✓ Operating systems will have to provide protocols, interfaces and resource managers that support easy consumption of services from the cloud as well as provisioning of cloud services.
- ✓ Operating systems need to efficiently exploit the compute power available from multi core and many-core processor architectures that are extended with various types of co-processors.
- ✓ Failure detection of hardware and platform software components and silent shutdown to support safe state of the system.
- ✓ Performance-optimized processor architectures to maximize computational throughput.
- ✓ Misbehavior detection of application software components and restart or shutdown of affected components to increase robustness of the system.

Application Software:

- ✓ A rich interface definition language to specify application software component interfaces unambiguously.
- ✓ A hardware-independent programming interface for acceleration hardware to port application software components easily.
- ✓ Software must run on variety of platforms and it should be platform independent to support portability.
- ✓ Parallel programming models and patterns through the support of cloud computing environments in order to efficiently utilize underlying hardware and deliver comparable performance.
- ✓ The software must exploit the power saving features of the hardware and allow for large sleep cycles as much as possible.

Self-Modifying Software:

- ✓ Artificial Intelligence program, capable of developing its own programs, using a genetic algorithm implementation with self-modifying and self-improving code that performs a specific task.
- ✓ Verifying the authenticity of a program's code during execution through Self-check summing.

6. ABCD Analysis of Ideal Software:

Any innovative concept or model or system can be analysed using either SWOC, or newly developed ABCD analysis method. ABCD listing and ABCD framework are two models of qualitative [17- 23] and quantitative ABCD analysis method [24-30] respectively. In this section, we have used ABCD analysis for qualitative listing of advantages, benefits, constraints and disadvantages of ideal software system.

Advantages:

- ✓ Ideal software has ideal characteristics which are the upper limits for developing software.
- ✓ Ideal software focus on optimum resource utilization at input.
- ✓ Ideal software processes any amount input at negligible time consumption.
- ✓ Ideal software can automate any processes in any system.
- ✓ Gives an idea in how to optimize the software performance.
- ✓ Ideal software cannot be copied by competitors.
- ✓ Ideal software provides ubiquitous and instantaneous service to everyone.

Benefits:

- ✓ Ideal software can process any amount of input and gives output immediately.
- ✓ 100% efficiency
- ✓ 100% customer satisfaction
- ✓ Benefit to designer to set their objectives to optimum level.
- ✓ Benefit to customer to choose comparatively better software.

Constraints:

- ✓ Design and development of ideal software is very challenging and cumbersome process.
- ✓ Design and manufacturing appropriate hardware for loading ideal software is also challenging task.
- ✓ Difficult or rather impossible to develop.

Disadvantages:

- ✓ Ideal software kills all other application software due to its universal automation ability.

- ✓ Ideal software if realized decreases jobs and employment opportunities in software industries.
- ✓ Ideal software if realized, supports lazy and lethargic people in the society.

7. Conclusion:

Ideal systems are those systems which have ideal characteristics. In any ideal system, their characteristics are divided into four categories as input characteristics, system characteristics, output characteristics, and environmental characteristics. In this paper, we have tried to develop ideal software concept by identified its characteristics. Further these characteristics are divided into four categories. The input characteristics identified are Zero input resources, Infinite selectivity, Ubiquitous input acceptance, Infinite input security, Infinite reliability, Infinite usability, Infinite efficiency in data acceptance, and Zero energy consumption at input. The operational system characteristics identified are Zero budget, Full correctness, Easy usability, Perfect integrity, 100% reliability, 100% efficiency, Infinite tolerance to security threats, 100% safety against hazards, Infinite functionality, Perfect Robustness. The transition system characteristics identified as Perfect interoperability, 100% reusability, perfect portability, and 100% performance guarantee. The maintenance system characteristics are found to be Zero maintenance cost, Perfect flexibility, Perfect generality, Infinite extensibility, Infinite scalability, Easy testability, Highest modularity, Best readability, Easy documentation for anybody use, Infinite tenant efficiency, and Easy configurability. The output characteristics identified are 100% accuracy, Perfectly correct output, Perfectly reliable output, Long term sustainability, Infinitely reusability, 100% output efficiency at very low input, Readability to everyone, and Perfect satisfied user experience. The identified environmental characteristics are Inelastic demand, Infinite market for ideal software, Infinite ability, Cannot be copied by others/competitors, and High quality service to every user. Even though ideal software is a hypothetical assumption and difficult to realize in practice, by knowing its characteristics, one can improve the characteristics of developed software towards ideal level. This will disclose an opportunity to software developers to continuously improve the quality of the software.

8. References:

1. Aithal P. S., and Shubhrajyotsna Aithal, Ideal Technology Concept & its Realization Opportunity using Nanotechnology, International Journal of Application or Innovation in Engineering & Management (IJAIEEM), Vol. 4, Issue 2, pp. 153 - 164, 2015.
2. Aithal P. S., Concept of Ideal Business & Its Realization Using E-Business Model, International Journal of Science and Research (IJSR), Vol. 4, Issue 3, pp. 1267 - 1274, March, 2015.
3. Aithal P.S., Mobile Business as an Optimum Model for Ideal Business. International Journal of Management, IT and Engineering (IJMIE), Vol. 5, Issue 7, pp. 146-159, (July 2015).
4. Aithal P. S. and Shubhrajyotsna Aithal, An Innovative Education Model to realize Ideal Education System, International Journal of Scientific Research and Management (IJSRM), Vol. 3, Issue 3, pp. 2464 - 2469, March, 2015.
5. Aithal P. S. and Shubhrajyotsna Aithal, (2014) Ideal education system and its realization through online education model using mobile devices, Proceedings of IISRO Multi Conference 2014, Bangkok, 7/01/2014, pp. 140 - 146, ISBN No. 978-81-927104-33-13.

6. Aithal P. S., & Shubhrajyotsna Aithal, (2016) Impact of On-line Education on Higher Education System, International Journal of Engineering Research and Modern Education (IJERME) Volume I, Issue I, 2016, pp. 225-235.
7. Aithal P. S., The concept of Ideal Strategy & its realization using White Ocean Mixed Strategy, International Journal of Management Sciences and Business Research (IJMSBR), Vol. 5, Issue 4, April 2016, pp. 171-179.
8. Sridhar Acharya P. and Aithal P. S., (2016) Concepts of Ideal Electric Energy System FOR production, distribution and utilization, International Journal of Management, IT and Engineering (IJMIE), Volume 6, Issue 1, pp. 367-379, (January 2016).
9. Aithal P. S., Concept of Ideal Banking and Realization of it using Ubiquitous Banking, Proceedings of National Conference on Changing Perspectives of Management, IT, and Social Sciences in Contemporary Environment, Manegma 2016, SIMS, Mangalore, India, Vol. 14, pp-13-24, ISBN 978-93-5265-6523.
10. Aithal P. S., Concept of Smart Library Models for Future Generations, Proceedings of National Conference on Changing Perspectives of Management, IT, and Social Sciences in Contemporary Environment, Manegma 2016, SIMS, Mangalore, India, Vol. 14, pp-50-62, ISBN 978-93-5265-6523.
11. Aithal P. S., Padmanabha Shenoy and Priyanka Neelam, Opportunities & Challenges in Starting Software Company in Developing Countries. International Journal of Management, IT and Engineering (IJMIE), Vol. 5, Issue 7, pp. 201-214, July 2015.
12. Penna Sparrow, Characteristics of a Good Software, Referred from <http://www.ianswer4u.com/2011/10/characteristics-of-good-software.html>.
13. Valeriy Vyatkin, Software Engineering in Industrial Automation: State-of-the-Art Review IEEE Transactions on Industrial Informatics, Vol. 9, Issue 3, pp. 1234 – 1249. 2013.
14. Vogel-Heuser, Birgit; Diedrich, Christian; Fay, Alexander; Jeschke, Sabine; Kowalewski, Stefan; et al., Challenges for Software Engineering in Automation, Journal of Software Engineering and Applications, Vol.7, Issue 5, pp. 440-451, May 2014.
15. Tariq Samad, Leveraging the Web: A Universal Framework for Building Automation, 2007 American Control Conference, New York, NY, pp. 4382 – 4387. 9-13 July 2007, ISSN 0743-1619.
16. Julien Chouinard, Software for Next Generation Automation and Control, 4th IEEE International Conference on Industrial Informatics, Singapore, 16-18 Aug. 2006, pp. 886 – 891.
17. Aithal P. S. and Suresh Kumar P.M., Opportunities and Challenges for Private Universities in India, International Journal of Management, IT and Engineering (IJMIE), Vol. 6, Issue 1, pp. 88-113, January 2016.
18. Sridhar Acharya P. And Aithal P. S., Concepts of Ideal Electric Energy System FOR production, distribution and utilization, International Journal of Management, IT and Engineering (IJMIE), Vol. 6, Issue 1, pp. 367-379, January 2016.
19. Padmanabha Shenoy, and Aithal P. S., A Study on History of Paper and possible Paper Free World, International Journal of Management, IT and Engineering (IJMIE), Vol. 6, Issue 1, pp. 337-355, January 2016.
20. Aithal, P.S., Comparative Study on MBA Programmes in Private & Public Universities - A case study of MBA programme plan of Srinivas University,

- International Journal of Management Sciences and Business Research (IJMSBR), Vol. 4, Issue 12, pp. 106-122. December 2015.
21. Aithal P. S., & Shubhrajyotsna Aithal, Impact of On-line Education on Higher Education System, International Journal of Engineering Research and Modern Education (IJERME), Vol. I, Issue I, pp. 225-235, March 2016.
 22. Aithal P. S., and Suresh Kumar P. M., Analysis of Choice Based Credit System in Higher Education, International Journal of Engineering Research and Modern Education (IJERME), Vol. I, Issue I, pp. 278-284, May 2016.
 23. Varun Shenoy and Aithal P. S., Changing Approaches in Campus Placements - A new futuristic Model, International Journal of Scientific Research and Modern Education (IJSRME), Vol. I, Issue I, pp. 766 – 776, June 2016.
 24. Aithal P. S., Shailashree V. T., Suresh Kumar P. M., A New ABCD Technique to Analyze Business Models & Concepts, International Journal of Management, IT and Engineering (IJMIE), Vol. 5, Issue 4, pp. 409 - 423, April 2015.
 25. Aithal P. S., Shailashree V. T., & Suresh Kumar P. M., Application of ABCD Analysis Model for Black Ocean Strategy, International Journal of Applied Research (IJAR), Vol. 1, Issue 10, pp. 331 - 337, Sept. 2015.
 26. Aithal P. S., Shailashree V. T., & Suresh Kumar P. M., ABCD analysis of Stage Model in Higher Education, International Journal of Management, IT and Engineering (IJMIE), Vol. 6, Issue 1, pp. 11-24, January 2016.
 27. Aithal P. S., Shailashree V.T., & Suresh Kumar P. M., Analysis of NAAC Accreditation System using ABCD framework, International Journal of Management, IT and Engineering (IJMIE), Vol. 6, Issue 1, pp. 30 - 44, January 2016.
 28. Aithal, P.S., Study on ABCD Analysis Technique for Business Models, Business strategies, Operating Concepts & Business Systems, International Journal in Management and Social Science, Vol. 4, Issue 1, pp. 98-115, 2016.
 29. Aithal P. S., Shailashree V. T., & Suresh Kumar P. M., Application of ABCD Analysis Framework on Private University System in India, International Journal of Management Sciences and Business Research (IJMSBR), Vol. 5, Issue 4, pp. 159-170, April 2016.
 30. Aithal P. S., Shailashree V. T., & Suresh Kumar P. M., The Study of New National Institutional Ranking System using ABCD Framework, International Journal of Current Research and Modern Education (IJCRME), Vol. I, Issue I, pp. 389 – 402, May 2016.