The Characteristics of Good Systems

Chandra Amaravadi,Zachary L. Lessard

ABSTRACT

Software engineering attempts to produce systems that are "good systems" in terms of reliability, ease of maintenance etc. ?We take a broader definition of a good system as any general system that produces benefits that exceed initial expectations or intended scope or initial investment. ?There appear to be common characteristics that tie together such systems. ?These are hypothesized to include functional "goodness", good infrastructure, reliability, connect-ability, versatility and benefits that overflow/overwhelm the system's scope or initial investment. ?A case study approach involving four examples of what are regarded as "good systems" and four examples of what are regarded as "bad systems" fully supports this hypothesis. ?But support for the converse hypothesis, a bad system not having these characteristics was only 68.7%. ?The implications of these findings are discussed.

Keywords: good systems, system characteristics, strategic systems, systems theory, software engineering philosophy, information systems philosophy, complex systems analysis.

Classification: D.2

Language: English

London Journals Press

LJP Copyright ID: 157266
ISBN 10: 1537631683
ISBN 13: 978-1537631684

London Journal of Research in Computer Science and Technology

1 7 U K

Volume 17 | Issue 1 | Compilation 1.0

The Characteristics of Good Systems

Chandra S. Amaravadi[α] & Zachary L. Lessard[σ]

I. ABSTRACT

Software engineering attempts to produce systems that are "good systems" in terms of reliability, ease of maintenance etc. We take a broader definition of a good system as any general system that produces benefits that exceed initial expectations or intended scope or initial investment. There appear to be common characteristics that tie together such systems. These are hypothesized to include functional "goodness", good infrastructure, reliability, connect-ability, versatility and benefits that overflow/overwhelm the system's scope or initial investment. A case study approach involving four examples of what are regarded as "good systems" and four examples of what are regarded as "bad systems" fully supports this hypothesis. But support for the converse hypothesis, a bad system not having these characteristics was only 68.7%. The implications of these findings are discussed.

Keywords: good systems, system characteristics, strategic systems, systems theory, software engineering philosophy, information systems philosophy, complex systems analysis.

Author α: School of Computer Sciences, College of Business and Technology, Western Illinois University.
σ: School of Law, Southern Illinois University.

II. INTRODUCTION

Systems theory is a very fundamental tenet of the Information systems field. The whole of software engineering revolves around the concept of systems. Bertalanffy (1950) developed systems theory based on observations from biology and physics. A biological organism consists of many sub-organisms that work together to process inputs and keep the organism alive. So he defined a *system* as anything composed of subsystems that work together towards the common goal of transforming inputs into outputs. A complex system can be understood by analyzing its sub-systems.

Systems theory led to systems engineering which is concerned with engineering complex systems. The goal of systems engineering is to manage complex systems so that they are reliable (Wikipedia '16a). It encompasses a number of ideas such as user requirements, systems architecture and reliability analysis that ultimately paved the way for software engineering.

In the 1980's Yourdon and others pioneered structured systems methodology to develop systems that fulfill their requirements and minimize maintenance (Page-Jones '82, Yourdon '80). They developed a number of tools and techniques for developing such systems based on the software development life cycle (SDLC). According to their methodology, during the *analysis* stage, systems were specified using Data-flow-diagramming, data dictionary and Structured English. This was carried out in a 'top-down' fashion starting with the system and 'decomposing' the subsystems. Thus DFDs could be drawn for several 'levels' of the system. In the *design* stage, DFDs were 'transformed' into a structure chart that exhibited certain characteristics. The structure chart, which is part of the high level design, consists of modules that called each other in a hierarchy. Design was 'top-down' in the sense that modules at the top co-ordinated the modules below them i.e "boss modules" call the modules below them. Lower level tasks such as input and output were carried out by modules at the lower levels of the structure chart.

Modules were treated as 'black boxes' considering only the 'inputs' and 'outputs' of each module. When carrying out the low level design, care was taken not to 'couple' modules tightly to other modules. *Coupling* is the degree of dependence one module has on another. A high degree of coupling results in problems in one module having ripple effects on other. Coupling is minimized by: removing unnecessary relationships, minimizing the number of necessary relationships, and decreasing the "tightness" of the relationships that are required. It allows for a module to be changed without affecting other modules, making systems easier to comprehend and maintain. (Page-Jones '82: 101-103). Minimizing coupling between modules was thus a design objective (ibid:101). Another key characteristic derived from structured programming is the concept of cohesion. *Cohesion* is the degree to which the activities performed by a module are related to other activities within the module. The more closely related the activities, the better the cohesion. Thus the notion of a good system from the software engineering point of view is to design complex systems as modules, using a top down design strategy, and as black boxes, to minimize coupling and maximize cohesion. These concepts are further solidified in the object oriented methodologies. In object oriented approaches, packaging of data and methods into objects ensures high cohesion. Restricting access to methods through declaring an object as public or private controls coupling. These practices reduce errors and result in more reliable and maintainable systems.

In a different vein, successful system implement -ations of the 1980's led Wiseman (1985) to postulate the concept of a strategic system. A *strategic system* is defined as a system that supports or shapes the competitive strategy of an organization (ibid). Following the generic strategies introduced by Porter, a firm can make several strategic thrusts including cost leadership, differentiation, innovation, growth and strategic alliances. Wiseman (ibid) provided several examples of strategic information systems including McKesson's distribution system for drugstores and supermarkets, Banc One's credit card processing network, American Airlines' Sabre system, Benetton's apparel system and Walmart's merchandise information system to illustrate these different strategies. They also shared a number of characteristics in common – they are mainly infrastructural, supported a critical process/ process group or supported the value chain of the company. Infrastructural systems provided communications (Banc One's credit card processing network) or linked the company with its customers or suppliers (McKesson's drug distribution system) or provided access to other informational resources. As an example of the latter, PaineWebber (now part of Swiss Bank) negotiated with State Street Bank to enable its customers to use the MasterTeller network. Thus it exploited the information resources of other customers. Benetton's system supported its value chain - order entry, manufacturing as well as dynamic order management (ibid). All of these systems are reported to have produced great benefits for the host organization. For McKesson, benefits included reduction of sales force from 700 to 15 with sales increasing from \$922m to \$4.8b (Clemons and Row '88). There were also indirect impacts on the industry such as reduction in number of participants from 180 to 90 and increase in market share for the top four participants (ibid). An added bonus was that McKesson was able to leverage its distribution expertise into other business areas such as veterinary supplies, beverages and general merchandise.

These cases lead one to suspect if there is something more to a good system beyond good software design i.e. characteristics such as good infrastructure, core process support etc. If good systems had generalizable characteristics, these must be true regardless of whether they are information systems or other general systems fulfilling the system definition. To test this concept, we developed a set of criteria for a 'good system' -- it is hypothesized that these criteria hold for good systems and are absent in bad systems. We selected four cases of what are regarded as "good systems" and four cases of what are regarded as "bad systems" to see if the hypotheses are valid. If validated, such criteria can be used to evaluate systems at an early stage to correct problematic systems and ensure that system investments yield desired results.

2.1 A description of the selected criteria

As discussed in the previous section, two streams of thought serve as inspiration for the present study. The first is the notion of a good system from the software engineering area and second the notion of a good information system as a strategic system. From the first we have characteristics such as modularity, black boxed-ness, low coupling, high cohesion which would result in systems that fulfill requirements, are readily maintainable and reliable. From the second, we have characteristics such as infrastructural ability, connectability and the notion of secondary and tertiary benefits. We discuss these and additional criteria in the following. We need to emphasize again that the discussion is not restricted to information systems but extends to any type of system fulfilling the definition of a system.

2.2 Fulfills its functionality

The philosophy of structured systems is to develop systems that fulfill their intended need. The entire SDLC process is geared towards this end (Page-Jones '80). Accuracy of the analysis stage is often considered key to developing systems that fulfill their functional requirements. So the first test of a good system is whether or not it fulfills its functionality in the present as well as the future, and for the purpose for which it is designed. Further since systems function under a variety of conditions, a good system should fulfill its functionality under all conditions.

2.3 Is Infrastructural

An infrastructure is defined as the collection of basic physical, organizational structures and facilities needed for the operation of a system (Oxford dictionary '15). For an airplane, the combination of airframe, fuselage, wings, rudder and engine forms the infrastructure. The significance of infrastructure arises from the nature of systems -- open systems are characterized by exchange of energies (Bertalanffy '50). In biological systems, fluids and chemicals are exchanged among components. Impediments to such flows will adversely affect the functioning of the system. Thus a good infrastructure is an indispensable requirement for a good system. A good system facilitates the flow of information/ goods within the system's scope. If the system's scope is an organization, then a good system should facilitate the flow of materials within an organization.

2.4 Is easily Connectable

It is hypothesized that a good system possesses a high degree and ease of connectability. This idea is related to coupling and modularity. If a system is modular, it enables it to be part of a larger system or enables it to be the host system for another system. Coupling is the other side of the coin. Low coupling and high cohesion lead to modularity and ease with which a system can be connected to another system. A simple example is a railway carriage (smaller system) being connected to a train (a larger system). If the coupling is high (electricity, vacuum, hydraulics) the process will be difficult. Low coupling on the hand, is essential to use ability. Connectivity to systems outside a system's environ -ment allows for goods or information to move freely into and out from the system just as carriages *connected* inside the train allow passengers to move back and forth. It enables a system to take advantage of assets in other systems and thus enhances their versatility. As discussed earlier, PaineWebber using the information assets of State Street Bank exemplifies this situation (Wiseman '85).

2.5 Is Adaptable/Versatile

A good system it is hypothesized, must be adaptable and versatile. It should be capable of being modified easily to serve a wide variety of functions. This has been illustrated by the case of Sabre system for American Airlines (Sabre Holdings 2015). Sabre was used not only for reservations but for crew scheduling and flight forecasting (ibid). To be considered adaptable and versatile, the system must readily lend itself to being utilized in a large number of potential scenarios and must readily shift to meet changes in its operating environment. This readiness for utilization appears to be a hallmark of a good system.

2.6 Is Reliable

Reliability is one of the main characteristics of a system as identified in the introduction. It is defined as the percentage of time the system is operational or can be measured in terms of % of failures. It will be operationalized differently depending on the type of system being dealt with. For airlines, reliability is defined as percentage of on-time arrivals (Watson et al. '06) whereas for aircraft it is given as the number of plane crashes per million departures (Boeing '14). Needless to say, reliability affects confidence in the system and therefore its usability, It is a required quality of a good system.

2.7 Produces Additional Benefits (or problems)

It is hypothesized that a good system provides benefits that are disproportionately high when compared to the initial investment (Amaravadi '05). The $40 million initial investment in SABRE has resulted in a spinoff worth $6.2b dollars (McCartney 1999). This may be evidence of the multiplicative benefits of good systems. Sabre revolutionized the airline industry by enabling computerized flight listings (ibid). Judging by this and other strategic systems, the impacts of a good system extend to different parts of the organization and in some cases to the industry and ultimately to society. Conversely a bad system should result in problems that exceed the scope of the system. For example, a badly designed building could cause problems in roofing, wiring, heating, ventilation and occupant movement.

III. METHODOLOGY

The discussion above leads to the following Hypotheses:

Hypothesis H1: A good system has all or most of the following characteristics: fulfills its functionality, has good infrastructure, ready connect-ability with other systems, good versatility/adaptability, good reliability and produces benefits far exceeding the initial investment (or scope of the system).

Hypothesis H2: A bad system has all or most of the following characteristics: does not fulfill its functionality, has poor infrastructure, poor connect-ability with other systems, poor adaptability, poor reliability and creates problems that exceed the scope of the system.

To test these hypotheses, the authors selected four cases of what are popularly regarded as good systems and four cases of what are popularly regarded as bad systems. An attempt was made to select systems that are from very diverse domains. The criteria for selection was: 1) does it satisfy the definition of a system?, 2) Is it sufficiently complex to be interesting? 3) Is it a dynamic system rather than a static system such as a road network? This requirement is imposed since system behavior is part of the hypotheses. 4) Is the system widely regarded as a good (bad) system? 5) Is sufficient literature available to verify the hypotheses? This requirement entailed selecting systems that are in the U.S. Then using available sources, authors evaluated the systems against the criteria. A score of '1' was given if a good system fulfilled a particular criteria, '0.5' if it did not completely fulfill it. If a characteristic did not apply, it is labelled 'NA' and given a score of '0'. These were added across each system and across all four good systems to give the confidence level. The reverse was done for a bad system. If a bad system did not fulfill a criteria, a score of '1' was assigned. Then these were also tallied. This process was necessarily subjective and is one of the limitations of this study. System descriptions are given to enable readers to verify author's perceptions.

IV. CASES OF GOOD SYSTEMS

Cases of good systems include an Ecommerce system (Amazon.com), a Supply chain management system (Walmart), a mobile device (Apple iPhone) and a search engine (Google). Although all of these are related to information technology this was not intentional.

4.1 Amazon.com Ecommerce site

The First system the authors have selected is the highly regarded e-commerce site of Amazon.com (Post '12). The company initially started selling books, but soon branched into music, movies, software, household goods, home improvement and video games among others. The site has a well developed infrastructure that allows for 24/7 product display and shopping. Customers can shop for any item from any division in any part of the web site. In 2000 Amazon.com overhauled its systems using DBMS from Oracle, logistics from Manugistics, Analytics from SAS and Excelon to support business to business integration (ibid). Amazon's merchant and marketplace systems allow merchants to sell their products on the Amazon.com site. This is enabled by Excelon which allows partners with limited IT to connect into its systems in real time. The site also supports the formation of communities and allows for contributions from affiliates. The website has additional benefits exceeding the initial scope of their system -- their e-commerce infrastructure is so effective that they have turned it into a product and the company now sells web services (Amazon web services) to companies such as Sears, Bebe, Marks and Spencer among others (Wikipedia 16b). Additionally, the price check application they developed allows a customer to check the price of goods in a store to see what the prices would be on amazon.com (Hane '12).

4.2 Walmart's Information System

The second system the authors selected is the much regarded supply-chain management system of Walmart (Lu '15). Walmart's homegrown systems support its strategy of cost minimization and rock bottom prices. Their systems have been behind its phenomenal growth (Gallaugher '12). The company founded in 1962 now has over two million employees and 11,000 stores and is the largest retailer in the world (Wikipedia 16c). Their philosophy of utilizing centralized and common systems/platforms ensures connect-ability and good infrastructure (Post '12). All inventory items in store are bar coded and purchases are scanned and recorded by item and date of sale at the cash registers. Other items purchased by the customer are also recorded this way. This makes it possible to know how the items are selling as well as how they are selling relative to other stores and other times of the year. The data is stored in a large 423 Terabyte data warehouse (Gallaugher '12). This basic system provides a good foundation for its other activities. The sales information is shared with Walmart's more than 5,000 suppliers through the Retail Link system and drives all inventory decisions (ibid, Holstein et al. '98). Walmart uses the data to stock up on fast selling items and reduce inventory on slow selling items. The sales information together with demo -graphic data allows Walmart to customize its stores to individual regions (Holstein et al. '98). Walmart's various systems function together to fulfill the company's strategy. Employees are provided with VOF (Voice Based Order Filling) to direct them to item locations and place inventory orders, thus saving time and reducing costs (Purpura '97). Its web site, although not as highly regarded, is linked with its homegrown system (Post '12). In addition Walmart is allowing its customers to check out using 'Walmart App' on their iPhones (Wohl '15).

4.3 Apple iPhone

The Apple iPhone is the most successful smart phone to date with more than a billion units sold as of 2016 (Statista 16a) or 23% of all smartphone sales (Statista '16b). The main features of the iPhone include its high resolution touch interface, one click access to applications, built-in wi-fi connectivity, ease of texting and messaging, camera, video conferencing, internet browsing and digital music capability (Want '10). These features are in addition to its voice communication capability. Thus the basic infrastructure consists of the communications, interface, networking and multimedia capability. Apple devices are known for their versatility. 'Apple Apps' make the iPhone very versatile. These apps include: navigation, bar code scanning, dictation, credit card processing, invoicing and remote control functionality for home appliances among millions of others (Appstore '15). The combination of powerful technologies and an open application model make business applications such as payment processing and workflow possible. The 'apps' have been used for everything from identifying fonts (Turner '09) to properly tilting a patient during a C-section (Ramamoorthy and Bailey '12). Medical students use the iPhone as a substitute to referring to books (Chatterly and Chojecki '10). The usefulness, versatility and benefits of the device defy description.

4.4 Google Search Engine

According to Wikipedia, the Google Search Engine is the internet's most popular search engine, handling more than 3.5 billion searches a day and accounting for 64.5% of the market share (Wikipedia '16d). The powerful infrastructure that Google has developed connects users to content they desire through the use of a giant index of ranked searchable links (Barroso et al. 2003). It has used this infrastructure effectively to diversify into a number of related products and services that combine to make it a technology juggernaut. The search engine can be easily embedded into other web sites such as 'cnn.com' for localized searches. Also the same technology can be used to search blogs, journal articles, images, and RSS feeds. Their "adsense" technology is tied to the searches such that a search for a product triggers advertisements it. Thus a search for "flowers" brings up advertisements of florists. This technology resulted in 2015 revenues of around $67.3 billion (Statista '16c). Also when certain keywords are entered unit, currency and time conversions, weather, stock quotes and airline schedules are triggered. The "Universal Search" feature combines search information from multiple pages and presents it as a summary (ibid). These features amply exhibit its versatility, connectability and its ability to generate additional benefits.

V. CASES OF "BAD" SYSTEMS

Cases of bad systems include an aircraft (DC-10), an operating system (DOS), a healthcare system (U.S. Healthcare) and database management systems (hierarchical systems).

5.1 McDonnell Douglas DC-10

The DC-10 was a wide bodied aircraft that was introduced by the McDonnell Douglas corporation in 1967 and certified airworthy in 1971 by the FAA (Kull '14). There was also a tanker version of the DC-10, the KC-10 which exhibits by its existence, some degree of adaptability. Although the DC-10 ultimately proved to be a reliable aircraft, it has been maligned due to a number of spectacular crashes (ibid). It is due to this reputation that we included the plane in this study. On 12th June, 1972, a cargo door blew out from the aircraft during a flight between Detroit and Buffalo. The resulting decompression caused the floor over the cargo compartment to cave in, damaging flight control cables. A similar problem caused the deaths of 346 passengers and crew in 1974. In another incident, the engine separated from the left wing and flipped up over the top of the aircraft resulting in the deaths of 271 passengers (ibid). In all there were 56 aviation occurrences including 32 hull-loss accidents with a total of 1,262 occupant fatalities (Wikipedia '16e). Clearly there were problems with the aircraft body which led to these fatalities, including the cargo door design, cargo roof design and engine mounting. Failure in one subsystem has also resulted in problems with other subsystems further reflecting a bad design. For example cargo area decompression led to floor collapse, which led to loss of aircraft control.

Despite occupant deaths, reliability of the aircraft (2.94 accidents per million departures) was comparable to Boeing 747 (2.85) and the Airbus A300 (2.29) (Boeing '14).

5.2 DOS and MS-DOS

DOS was introduced in 1981 as the operating system for the IBM PC (Wikipedia 16f). Although there were three different brands, PC-DOS, MS-DOS, DR-DOS the most popular variant was MS-DOS and we refer to this here between 1981-1985. It was the dominant operating system for PCs until replaced by Windows in the mid 1990's. DOS was designed as a single tasking, single user system based on the Intel 8086 microprocessor (Paterson '15). Its core modules include file/directory management, memory management, command interpreter and kernel programs (Verma '09, Paterson '83). Since the main purpose of operating systems is to control the hardware, they are necessarily coupled to a particular hardware. This makes adaptability a non-issue. DOS was hastily developed and it had a number of limitations that continued into later versions. The first version was disk based and had to be physically loaded into RAM from floppy disks. Later when hard disks were introduced, DOS could be configured to run from hard disk. The initial versions also did not have device drivers, and these had to be separately installed until MS-DOS version 5 was introduced in 1991 (Ferelli '92). The major limitation of DOS was the 640K limit on memory imposed by the underlying 8086 architecture (Gookin '91). Programmers could address only 640K memory in 64K RAM segments causing them to write program chunks that utilized only 64K segments. This issue was rectified with the introduction of the 80386 processor even though maintaining backwards compatibility required extra programming on the part of the programmers (McDugall 2013). This was a nightmare by all accounts (ibid). There was also a file size limitation of 65K owing to the 16 bit architecture which dictates the address size (Disc 2015). This was corrected in later versions, but due to the "Fat16" volume size, file sizes were limited to 2 GB (ibid). To round out the list of limitations, the command line interface was user unfriendly and required users to remember commands like "dir *.*, copy *.*" etc. The command line interface was replaced in the mid-nineties with the introduction of Windows.

5.3 The U.S. healthcare system

According to Wikipedia the health care system is the organization of people, institutions, and resources that deliver *health care* services to meet the *health* needs of target populations. The U.S. Health care system has received criticism from many quarters, which justifies its inclusion here (see for example Brodwin '14). On a number of measures (80 in all) lumped into the dimensions of quality, efficiency of care, equitable-ness and health indicators, the U.S. ranked last when compared to a list of 11 advanced countries that includes Australia, Canada and European countries (Davis et al. 2014). The per-capita healthcare spending costs are also highest when compared to these countries. Other than costs, there are problems of uninsureds, inefficiencies in co-ordination among different agencies, patient deaths due to hospital errors, inefficient information exchange, billing inefficiencies among others (ibid, Jost '06). These problems amount to a poorly functioning system. The basic infrastructure of hospitals, physicians, nurses, staff and facilities are present in the U.S. although to a lesser extent than some European countries (Anderson and Squires '10). Co-ordination of information about patient has generally been described as poor (Jost '06). According to one study, the U.S. ranks 6th out of 11 in this respect (Davis et al. 2014). According to another study between 220,000-440,000 patient deaths are preventable with effective healthcare (Npr '13). Adaptability of the system cannot be judged as it has not been adapted to different systems of medicine, but it has certainly proved versatile in the face of new technologies. There are many additional problems caused by the healthcare system. Patient mobility between physicians is restricted due to the in-network limitation imposed by many HMOs (Garson 2000). Patients also do not know their financial liability until after their treatment.

5.4 Hierarchical Database Management Systems

Hierarchical database management systems are exemplified by IMS (Information Management System), a database management system developed by IBM in 1966 (Wikipedia 16g). In hierarchical database management systems, records are defined using "segments" i.e. what would correspond with tables in relational databases are defined as segments in hierarchical systems (Panneerselvam 2004). Each segment has a number of fields that correspond to attributes in relational databases. Segments can have child segments such as a department having employees. The structure of the database is thus encoded in the data definition in the form of data definition language (DDL) statements. This hierarchical structure embedded in the DDL in the form of segments and "records" forms the infrastructure of hierarchical dbms. This is awkward at best since it is suited for 1:M relationships in the data (for example the relationship Department: Employees) rather than M:N relationships which tend to be more frequent (for example the relationship Companies: Suppliers) and for which the relational model is eminently suited (ibid). Retrieval also presented problems. To retrieve data it is necessary to "navigate" the structure. For example, a segment can be retrieved directly with 'GU' (Get Unique). 'GN' (Get Next) gets the next occurrence of the segment. 'GNP' gets the next occurrence of a child segment under a particular parent. After navigating the structure, data manipulation operations such as 'ISRT' and 'DELT' could be performed (ibid). Thus there are two sets of operators, one for navigation and one for data manipulation, whereas, relational systems required only one set of operators for data manipulation and retrieval (Gibbs '85). Another great disadvantage of the hierarchical scheme was for the need to know database structure for retrieval since it is necessary to navigate this structure which quickly becomes complex when there are more than a dozen segments with interrelationships. Hierarchical systems fell by the wayside as a result of these disadvantages (Prescott et al. 2010). However, because of the "tree" organization of data and indexing, retrieval was very fast. The DBMS can be connected to a transaction manager for use in transaction processing environments. Speeds of up to 100,000 TPS (transactions per second) have been recorded (Wikipedia '16g). Not surprisingly hierarchical systems have been used reliably in the banking industry (ibid).

VI. RESULTS AND DISCUSSION

Results of the study are shown in summary form in *Table 1* below and in detail in *Appendix 1*. It is seen that there is 100% support for Hypothesis 1; a good system has the characteristics of fulfilling its primary function well, of having a good infrastructure, being readily connectable, being versatile/adaptable, being reliable and producing a number of secondary benefits. Unfortunately, the case is not so clear-cut in the case of bad systems. As seen from table 2 there is only a 68.75% support for Hypothesis 2 -- not fulfilling its primary function, not having a good infrastructure, not being connectable, not being adaptable, reliable and producing problems instead of benefits. One reason for the asymmetry in results may be that only a few of the characteristics when unfulfilled are sufficient to move a system from the "good" category to the "bad" category i.e. only a few of the characteristics may be enough to prevent a system from being a good system. Bad systems can arise from problems in the subsystems which in turn leads to some or most of the criteria to be unfulfilled. This is seen in the cases of DOS and DC-10. In DOS the problem was with the microprocessor used for the PC whereas for the DC-10 it was the cargo door, engine mount and floor design. Component problems could result from sub-optimal design decisions that are forced by design constraints. These are evidenced in the case of DOS. Here memory limitations clearly led to several problems that affected application programs for years. Bad systems could also result if the system is so complex that the inter-relationships cannot be controlled. We see this in the case of the U.S.

healthcare system. Here there is a complex relationship between providers, physicians, pharmacies and insurance companies. The Apple iPhone presents the opposite end of the spectrum of systems. Here the individual subsystems have been endowed with powerful capabilities that applications (iApps) could exploit resulting in a very versatile device.

In this research there was no attempt to control for system complexity, although an attempt was made to ensure that the systems are roughly comparable to one another. It would seem that simple systems such as pumps or internal combustion engines evolve fairly rapidly in the milieu of modern business and technological forces. The same does not appear true for complex systems. These perhaps present a degree of "wickedness" that defies good organization/ design (Selfridge et al. '84). Secondly if complex systems have human elements as in healthcare or aircraft maintenance, important system characteristics such as functionality and reliability are affected. The situation is aggravated if different parts of the system are under control of different entities. Therefore it is safe to say that the more complex the system the more the likelihood of it not satisfying all the criteria and therefore being a "bad system." Obviously such systems are undesirable since like DOS and the DC-10, these cause problems to all. The other side of the coin is whether or not the satisfaction of these characteristics at the design stage would ensure that system would be a good system. Empirical verification is needed to support such a conclusion.

Table 1: Summary Evaluation of Good System Characteristics

Type of system	*∑ Score 'Yes' on system characteristics*	*∑ Score 'No' on system characteristics*	*Confidence*
Good system	24	0	H1: 100%
Bad system	6.5	16.5	H2: 68.7%

VII. CONCLUSIONS AND IMPLICATIONS

Based on the cases presented, there is strong support for the hypothesis that good systems have the characteristics as identified in this research, viz. fulfills its primary function, good infrastructure, is easily connectable, is adaptable/ versatile, reliable and finally produces benefits that exceed the system's scope. A primary limitation of the study is its qualitative nature. Evaluation of whether or not the criteria was fulfilled was subjective. There was also no attempt to control for the size or complexity of the system although all systems presented have their own levels of complexity. There is no assurance that the criteria are complete. Some criteria such as modularity, configurability, flexibility, evolvability, architecture, feedback and self organization have not been considered. These, it was felt would not be applicable to all types of systems. For example, an airplane is a self-contained system and modularity would not be relevant to whether or not it is a good system from the point of view of transporting passengers. Similarly the notion of an architecture is not appropriate to systems such as the healthcare system. There is obviously opportunity for further development.

The findings have to be further verified and if possible extended. If empirically proved these factors have important implications for designing systems, especially those satisfying the broader definition of systems such as dams and buildings. An extended list of characteristics could be utilized as a checklist for developing good systems. In addition, the research suggests that high- level technological or business constraints or complex interrelationships can result in a poor design that causes innumerable problems. This is also an idea worth exploring because the economic and other costs of a badly designed system are staggering.

REFERENCES

1. Amaravadi, C. S., (2004). “The Laws of Information Systems,” *Journal of Management Research*, 4(3), pp 129-137, December.
2. Anonymous (2010). “Walmart makes fashion statement with item-level RFID”. Material Handling Management July 26th (no volume, issue or page information).
3. Anderson, G. F., and Squires, D. A. (2010). Measuring the US health care system: a cross-national comparison. Issue Brief (Commonwealth Fund), 90, 1-10.
4. App Store (2015). Apple Inc., https://itunes.apple.com/us/genre/ios/id36?mt=8.
5. Barroso, L. A., Dean, J., and Hölzle, U. (2003). Web search for a planet: The google cluster architecture. *IEEE Micro*, 23(2), 22-28.
6. Boeing (2014). Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations|1959–2013 (August, 2014). Retrieved from http://www.boeing.com/news/techissues/pdf/statsum.pdf current July 2015.
7. Bertalanffy, L. V. (1950). An Outline of General Systems Theory," *The British Journal for the Philosophy of Science,* 1:2 August, pp. 134-165.
8. Brin, S., and Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18), 3825-3833.
9. Brodwin. E., 2014. 11 Charts that show exactly what’s wrong with the US Healthcare System. *Business Insider*, September 23. http://finance.yahoo.com/news/11-charts-show-exactly-whats-133000491.html.
10. Chatterley, T., & Chojecki, D. (2010). Personal digital assistant usage among undergraduate medical students: exploring trends, barriers, and the advent of smartphones. *Journal of the Medical Library Association* : JMLA, 98(2), 157-160.
11. Clemons, E. K., & Row, M. (1988). McKesson Drug Company: a case study of Economost—a strategic information system. *Journal of Management Information Systems*, 5(1), 36-50.
12. Davis, K., Stremikis, K., Squires, D., Schoen, C. (2014). Mirror, Mirror on the Wall, How the performance of the U.S. Healthcare system compares internationally, *Commonwealth* Fund, June. http://www.Commonwealthfund.org/~/media/files/publications/fund-report/2014/jun/1755_davis_mirror_mirror_2014.pdf
13. Disc, (2015), PC Large-File Limitations, Data Interchange Service Company, http://www.3480-3590-data-conversion.com/article-large-files.html, Current July 2015.
14. Ferelli, M. (1992). Device Drivers Seek Universality, *Computer Technology Review*, 12(3), March, p34.
15. Garson, A. (2000). The US Healthcare System 2010 Problems, Principles, and Potential Solutions. *Circulation*, 101(16), 2015-2016.
16. Gallaugher, J. (2012). “Data asset in action: Technology and the rise of Walmart.” Getting the most out of information systems, Creative Commons, pp 467-470.
17. German, K., (2007). Apple iPhone review, CNET reviews, June 30, http://www.cnet.com/products/apple-iphone/
18. Gibbs, S.J. (1985). Conceptual modelling and office information systems. In Tsichritzis, D. (ed.), Office Automation, Springer-Verlag. pp. 193-225.
19. Gookin, D. (1991). Memory managers: Taming DOS' RAM. InfoWorld, 13(49), 69-69, 72+
20. Hane, J. P. (2012), Amazon’s Ever-Expanding Empire, *Information Today* February 29(2),
21. Holstein, W., Sieder, J., Svetcov, D. (1998). Data-crunching Santa. *U.S. News & World Report*, 125 (24), p44.

22. Hull, K., (2014). *Airline Reporter*, A Historical Look at the DC-10 Before its Final Passenger Flight, February 19th. [http://www.airlinereporter.com/2014/02/historical-look-dc-10-final-passenger-flight/]
23. Jost, T. S. (2006). Our broken health care system and how to fix it: an essay on health law and policy. *Wake Forest Law Rev*iew, 41, 537.
24. Lu, C. (2015) "Incredibly successful supply chain management: how does Walmart do it?" TradeGecko. https://www.tradegecko.com/blog/incredibly-successful-supply-chain-management-walmart.
25. McCartney, S. (1999). AMR announces its plan to spin off Sabre Holdings. Wall Street Journal, Dec 14. http://www.wsj.com/articles/SB945130846522949523 accessed July 2015.
26. Myers, D., (1983). "Porting MS-DOS", *Systems International*, 11(9), September, pp. 106-107.
27. McDougall, S., (2013). Limitations of the IBM PC Architecture. The World, http://world.std.com/~swmcd/steven/rants/pc.html.
28. Npr.org (2013). How many die from medical mistakes in U. S. Hospitals? September 20 http://www.npr.org/blogs/health/2013/09/20/224507654/how-many-die-from-medical-mistakes-in-u-s-hospitals, Accessed July 2015.
29. Oliver, D. W., Kelliher, T. P., Keegan, J. G. (1997). *Engineering Complex Systems with Models and Objects*. McGraw-Hill.
30. Oxford dictionary (2015). http://www.oxforddictionaries.com/us/definition/american_english/infrastructure.
31. Page-Jones, M. (1980). *The Practical Guide to Structured Systems Design*. New York, NY: Yourdon Press.
32. Panneerselvam, R. (2004). Database Management Systems. PHI Learning Pvt. Ltd.
33. Paterson, T. (2015). "An inside look at MS-DOS", Byte Magazine, June 1983. http://www.patersontech.com/dos/byte%E2%80%93 inside-look.aspx
34. Post, G., (2012). Cases in MIS, https://www.jerrypost.com/MIS/MISCases2012.pdf.
35. Prescott, M. B., McFadden, F. R., & Hoffer, J. A.. *Modern Database Management*. Pearson Higher Education, 2010
36. Purpura, Linda. (1997). At Wal-mart, voice based order filling speaks up, *Supermarket news*, June 30, http://supermarketnews.com.
37. Ramamoorthy, K. G., and Bailey, K (2012). iPhone for measuring tilt during 150 caesarean section, *Anaesthesia*, 67(5), 551-552, May.
38. Rogowski, M. (2014). Without much fanfare Apple has sold its 500 millionth iPhone, Forbes, March 25, http://www.forbes.com/sites/markro-gowsky/2014/03/25/without-much-fanfare-apple-has-sold-its-500-million th-iphone/.
39. Selfridge, O., Rissland, E., and Arbib, M. (1984). (editors), Adaptive control of ill-defined systems, Plenum Press.
40. Sabre Holdings. (2015). Sabre History. *Sabre Holdings*. http://www.sabre.com/home/about/sabre_history Accessed November 2016.
41. Statista (2016a). Global Apple iPhone sales from 3rd quarter 2007 to 4th quarter 2016 (in million units). Current August 2016, http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
42. Statista (2016b). Number of smartphones sold to end users worldwide from 2007 to 2015. Current August 2016, http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
43. Statista (2016c). Google's ad revenue from 2001 to 2015 (in billion U.S. dollars) Current November 2016, https://www.statista.com/statistics/266249/advertising-revenue-of-google/
44. Turner, K. (2009). 7 Surprising uses for the iPhone's camera, Macworld, May 7. http://

www.macworld.com/article/1140435/iPhone_photo_tips.html.

45. Verma, S. (2009). Operating System, Krishna Prakashan Media, Meerut, India.
46. Want, R. (2010). IPhone: Smarter than the average phone. *IEEE Pervasive Computing*, 9(3), 6-9.
47. Watson, H. J., Wixom, B. H., Hoffer, J. A., Anderson-Lehman, R., & Reynolds, A. M. (2006). Real-Time Business Intelligence: Best Practices at Continental Airlines. *Information Systems Management*, 23(1), 7-18.
48. Wikipedia (2016a). Systems Engineering. Current November 2016 http://en. wikipedia.org /wiki/Systems_engineering.
49. Wikipedia (2016b). Amazon.Com Current November 2016. http://en.wikipedia.org/ wiki/Amazon.com.
50. Wikipedia (2016c). Current August 2016. Walmart, https://en.wikipedia.org/wiki/ Walmart.
51. Wikipedia (2016d). Google Search Engine, http://en.wikipedia.org/wiki/Google_search _engine. Current November 2016.
52. Wikipedia (2016e). McDonnell Douglas DC-10, Current November 2016. http://en. Wikipedia.org/wiki/McDonnell_Douglas_ DC-10
53. Wikipedia (2016f). DOS, https://en. wikipedia.org/wiki/DOS. Current November 2016.
54. Wikipedia (2016g). Information Management System, https://en.wikipedia.org/ wiki/IBM_Information_Management_ System. Current November 2016.
55. Wiseman, C. (1985). *Strategy and Computers*. Homewood, IL: Dow Jones-Irwin
56. Wohl, J. (2015) Walmart Adds iPhone Scan-and-Checkout Feature to 12 More Markets, March 20 Reuters. http://www.reuters.com/ article/2013/03/20/us-walmart-checkout-ex pansion-idUSBRE92J0P020130320.
57. Yourdon, E. (1988). *Modern Structured Analysis*. Prentice Hall.

Appendix 1: Evaluation of Systems along Criteria

Criteria → System	Fulfills primary function well?	Has good Infrastructure?	Connectability?	Adaptable/Versatile ?	Reliable?	Additional Benefits/ (Problems)?
Amazon.com e-commerce system (Post '12, Wikipedia '16b)	Yes, the system is the only outlet for Amazon, #1 online retailer.	Yes! In 2000 Amazon revamped its infrastructure to include DBMS, ERP, mining and analysis. Ordering and fulfillment added in 2004.	Yes! Their new IT system allows easy connections with supplier's systems as well as their partners.	Very versatile. Initially started selling books then branched into music, toys, electronics, apparel and more..	Reliable. Web site failures are rare.	Yes. They branched into cloud services in 2002.
Wal-mart information system (Post '12, Gallaugher '12, Wikipedia '16c).	Yes, system enabled Walmart to be #1 retailer.	Focuses on using a centralized system with common platforms. Items RFID'd and scanned. sales transactions stored in a warehouse.	Yes. Systems integrated with Voice Order Filling and iPhone Apps.	Yes, easily adapted to several store formats.	Yes, apparently very robust.	Yes. Ability to spot high volume products through use of warehousing and RFID technologies.
Apple iphone (German '07, Want '10)	Yes, millions of users have used iPhone – many for critical tasks.	Built around a home screen and a graphical menu of available apps. Also provides 3G support.	Can provide connections through local WiFi networks, EDGE networks, or GSM Networks.	'Apple Apps' make the iPhone very versatile. Including: navigation, bar code scanning, dictation, credit card processing, invoicing and remote control	Yes, also dependent on battery.	Yes. It is being used as a payment processing terminal by some mobile vendors. Millions of other apps.

				functionality for home appliances.		
Google Search Engine (Barroso et al '03; Wikipedia '16d)	Yes. Extremely well. More than 3.5 billion searches/ day.	Yes. Has a giant index of websites and cached web pages that coupled with a page ranking algorithm, is able to respond quickly to user queries.	Search engine can be easily embedded into other web sites for local searches.	Very Adaptable. Initially started for word search, but expanded into synonyms, stock quotes etc. The same technology has been used to search blogs, journal articles, images, and RSS feeds.	Yes, due to sophisticate d algorithms.	Yes. Triggers special features for some keywords, such as: unit, currency, and time conversions, weather, stock quotes, and discrete math functions.
DOS (McDugall '13, Paterson '15)	No. Does not have device drivers, needed for writing any application.	No. Single user single tasking operating system.	No. DOS has always been plagued with inconsistencies and incompatibilities between different software programs.	No. DOS was written for the Intel 8086 family.	No. System crashes were frequent.	(Yes), developing applications was a nightmare.
DC-10/MD-10 (anonymous '14, Wikipedia '16e, Boeing '14)	Yes, it transported cargo and passengers as well as other aircraft.	Yes and No. Basic infrastructure sound, but engine mount, cargo door and hydraulic systems designs have led to crashes.	NA. An aircraft is meant to be an independent system.	Versatile. Had 13 versions, served a wide range of uses from short distance to long distance, cargo transport and tanker applications	Initially poor but later reliable - 2.94 accidents per million departures compared to 2.85 for Boeing 747 and 2.29 for Airbus A300.	(Yes). In three separate incidents, hydraulic systems were ruptured by decompression and engine separation.
Healthcare system (Davis et al. '14, Jost '06, Npr '13).	No, poorly functioning when compared to other advanced countries.	Yes but basic infrastructure is slightly poor when compared to other countries.	Poor. It is very difficult to share patient information.	No. Adaptability is not evident since the system has not been adapted to alternative systems of medicine.	Unacceptable because humans are involved. Between 220,000-440,000 deaths annually.	(Yes)! System makes it difficult for patients to move from one physician to another. Patients do not know their obligation prior to treatment.
Hierarchical DBMS (Panneerselvam '04, Wikipedia '16g)	Yes and No. Good for storage but not retrieval.	No. Data is stored in a hierarchy with links between them – made for poor infrastructure	Yes, system could be connected to other systems for banking etc.	No. It was difficult to write queries for different views than what was designed.	Yes. Very reliable, used in the banking industry.	(Yes). Users had to be aware of database structure to write queries.