International Journal of Engineering & Technology, 7 (2.33) (2018) 71-75

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A novel based software testing process for user-friendly projects

Kuntam Babu Rao *

Vice-Principal & Professor in Computer Science and Engineering Department at Vaageswari College of Engineering, Karimnagar
**Corresponding author E-mail: professor.kbrao@gmail.com*

Abstract

Technology is growing very fast with various changes in Computer Systems. They are chasing human life. They are used in various fields around us. There are some governments who fund for computer systems. Still research related them is carry on and will carry. Mostly there services are used in defense purpose. Based on computer technology there are so many improvement in them, even embedded system are developed. Any system can present some risk to its owner's, users and environment. Risk is based on low and high. Some systems present more risk than others. Those which are related to more risk are called safety-critical systems. Safety critical systems are categorized into dangerous systems whose failures could result in loss of life, loss of revenue, knowing them and unable to solve them is sabotage to property damage or damage to the environment. This paper focus better exercises for making good user-friendly software. Every system has two sides as if coin, one side is the physical system with hardware and other side is software. If anything is neglected, there is a chance of bug appears at run-time. This may cause loss to humans. Again a committee report will be preserved for future purpose.

Keywords: *Safety-Critical Systems; Software; User-Friendly; Hardware.*

1. Introduction

Computer Systems plays a key role in todays technology. Microprocessor is a main component in Computer System. Microprocessor is having similar history as Computer System. We have a single core processor to multi-core processors. They are technology is used in very complex systems. Sometimes complex systems may be treated as danger to human life and environment. Paper deals with some of the errors, reasons that dragged them to failures and how to overcome them at high risks.

Software plays a key in computer systems because hardware parts of computer systems are becoming cheaper and cheaper day-to-day. It is also necessary to focus on safety critical systems. Software can exhibit a wide range of different behaviors. With these behaviors and attitude engineers got brilliant ideas in implement software everywhere like jet engines, submarines, ships, controlling missiles to target locations, chemical plants, embedded systems, production and manufacturing area, accounts, medical scanning, medical surgery, treating cancer patients, defense bombs, banking system and airplanes.

Computer systems are accurate then humans, there is a phobia that all humans will be replaced by computer systems in coming days. Every two to four humans are replaced with a computer system. Where a human cannot achieve tasks are also chased by computer systems. These system should be designed and maintain properly without causing any damage to the human life. Software has grown with importance in computer systems, it controls the computer system and finally software is the computer system.

Every system has two sides of a coin, it is complex and there is a high risk which should be concern for safety. Safety is treated as being free from accidents or loss. When we go for building complex process they should be checked or tested for validity. But not all systems are checked for validity. Reason is that they are complex process and professionals or developers or programmers cannot penetrate to combat complexity in them because they are expensive and have high risk. All high risk systems should be concerned with safety. They are tested to particular stage and based on parameters they will decide the safety issues. Example if you take a missile that targets different locations on global, it is tested to particular range and after that is declare the target range.

Based on above versions it is declare that software is neither safe nor unsafe; however, it is part of safety-critical system.

2. Focus on safety

Every complex system is not completely safe. If we focus on safety that means jet plane cannot fly safely, a car cannot run safely and a boat cannot set free to sail in water. No system is safe to use. Then we do not have any history to speak about them. We need not feel proud in describing them, so no real system can be made completely safe. Example point of view if you see airplane they are unsafe but people are still interested in travelling and pilots are interested in drive them. We cannot stop anybody. But they should be properly engaged and there should be good communication between professionals and the device. Airplane is complex machinery with group of people and devices. It is made by humans, they had every credit in build them and in updating them in making perfect every day. People cannot stop there travelling even though then are unsafe but it is the duty of government and professional to create a safety system to save human life. People or public they feel that it is perfect safe and travel from one place to other place with confidence. When we speak about safety and save human life means the cost related to system will increase. This should be balanced making safety first and next the cost. Software

professional first they should plan properly for the complex system; they should balance every aspect to it, precede with past history errors and built a better system.

There is lot of million dollars are spend over the research of software products by USA government. There primary focus is on safety related issues. It can be a complex software development projects which should be handed very planning and should focus on safety. Therefore, we must make every effort to ensure that software-intensive systems are reliable and safe. The aviation industry has a good track record, but as complexity and criticality increase, care must be taken.

However, now is not the time to sit back and marvel at over past, the future is riskier for the following:

- SLoC is base for generating software. Source Line of Code plays a key role in software. It should be properly code and should focus on safety-critical systems. For example, the sources lines of code from the Boeing airplane 777 to the recently certified Boeing 787 have increased ten fold to twelve fold. Upcoming complex systems will have more code in future.
- High complexity: Day to day we come known about complex system. Software process itself is a complex process. Complexity of software increases based on its requirements. For example, Integrated Modular Avionics (IMA) are real-time computer networks airborne systems, this network consists of different critical levels in implementing them. IMA provides a framework with weight savings, easier installation, efficient maintenance, and reduced cost of change. Single hardware platform consists of different functions associated at one point and have to perform multiple functions. This process is a complex process and intended to have safety.
- High critical levels: A normal program contains few lines of code, where as complex program contain more code and more complex. When we speak about more means it is in critical area. For example, flights are controller interfaces are based on mechanical in past fifteen years ago; recent years onwards aircraft manufactures are transitioning to flu-by-wire software. Flight technology is a complex process and needs safety and it is a critical element. Complexity also rises based on increasing size of the flight. Hence they should be considered based on performance and stability wise.
- Technology transitions: Rapid changing elements in technology are electronics and software. New inventions are taking place around the world, out dated technology is wiped out with new technology. Technology is taking its new challenges everyday and most of the government spends lot of dollars on them. For example, an aircraft building takes five years and microprocessor is taking less than one year to see latest release. Easily old microprocessors are replaced by new technology microprocessors. When new microprocessor releases it contains new features which support new technology. Same code related to them is also changing. Sometimes it may require ada, c, c++ and java programmers to map software with hardware processors. It is a complex process.
- Increased outsourcing and off shoring: software profession is facing lot of crises related to employment. Because of this reason many software systems are facing problems. They are not tested properly because of existing professionals are have domain knowledge about the system and injecting new errors. Which are treated as serious problem and requires safety-critical system.
- Lack of unavailable training: Every day we are having news about new softwares. They are released to the market with promos and teasers. But there are lack of professional to train and guide young engineers. How we can get quality products? And there is no question for safety, validation and verification.
- Lack of proper testing of the system. Mostly when testing is related to large systems like aircraft, submarine, missiles, rockets, space shuttles and so on, they are not completed tested. Example, if a missile ranging upto Islamabad is not test exactly. They test for few meters and conform that it can travel upto Islamabad.

3. Existing system

Failures are the main reasons for not considering total system. It can be Flight, Submarine, any Embedded System, Nuclear Plants, Missiles, Satellites, Radars, Defense Rockets and Vehicles etc., Because every system in above list is related to software. Software plays a key role in all systems. Every system is of two parts, Physical system and Software System. Software is like a shield over physical system. Example flight system (Fig. 1 Physical System of Airplane) has flight controls, IMA, landing gear, flaps, ice protection, nose wheel steering, flight management, battery management, display, navigation, terrain awareness and warning, traffic collision and avoidance, real-time operating systems and many more.

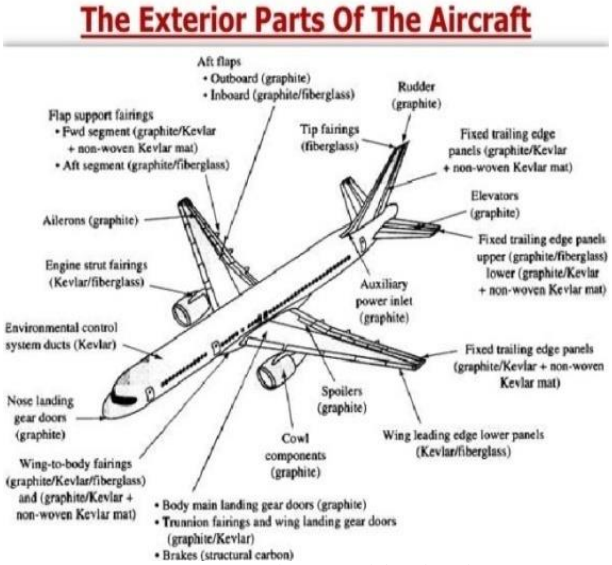

Fig. 1: For Exterior Parts of the Aircraft.

They all are developed physically with boards and chips (hardware) and software is added to it later. The variety of positions and systems has to be experienced and observe common issues, as well as effective solutions, for developing safety-critical software. Inside airplane, every information about external is shown in the form of signals and other data (Fig. 2 Cockpit). Information which carries from external to the pilot is based on software. Software is designed based on prototype test and past history of failures. Whenever a new problem related to software or physical arises the device should not loop between code and exceptions.

Fig. 2: For Cockpit Signals.

4. Literature study

1) Software Safety

We announce greatly about software and software related products but there they are concluded with properly. There is no proper finishing to them. There is no software project which is implemented 100 percent of 100 percent design. Every project will have risk and it is a complex process. After coining software engineering, the systems are made safe to use or out of danger. While designing and implementing software products one has to follow 360 degree view of the product. Professionals should be in a conduction to assess what is degree of safety and what level of complexity will applicable. They should think and review properly and consider any catastrophic situations. Primary think to be considered is safety, safety of humans and next step is safety of the devices. Most of the errors or problems are identified at run-time maintenance of the software project. At that time professional will get more idea about system and its problems. It is up to the professionals to correct the system in tuff times.

2) The Role of Software in Recent Catastrophic Accidents

In history of computer systems we have seen lot failures. All the mistakes are done by software professionals. Because lack of knowledge, not properly understanding the system, no domain knowledge about the system, and other things which lead to damage of the human life. Because of software mistakes we lost so many innocent people. Now a days software is used everywhere in areas of transport, space, medical, defense, aerospace, websites, electro-mechanical systems, electronics and day-to-day life. We interact every day with such systems and they should be safely operated by the humans. It is up to the professionals to make them safe and user friendly. Sometimes there may not be errors or faults in the software directly or indirectly and it may cost human life. When accidents appear, learning about the accident is important and rectifying the accident in upcoming software is important. We maintain a library to archive them in future use or consideration.

5. Proposed system

World has gone through various stages in computer systems since 1970's onwards. From day one to today if we observe computer systems they are from huge mass to mirco mass. Now they are seen as tiny devices. That is the power of computer systems. Almost every computer system will work based on the software that it uses. Software plays a key role in every device. Without software there will be no hardware. Software is of two types, one system related software and application related software. Both the software should be tested. The test may be related to safety. Everywhere people use so many devices around them and they should test for safety. Based on systems complexity level the coding and testing of the devices will take place. Televisions, telephones, gaming system, and automobiles all contain computers and computers themselves. Finally, the software we write today potentially touches millions of people, they have their own works related to the job and they use computer systems and computer related systems with trust and faith that they will never harm them.

1) Software Testing

Software process face a big hurdle that is software testing. Every software project should be tested for it is deployment. Software companies have templates for testing the devices. Some companies maintain a check list for testing. This exercise is prepared because of to deliver error free software to the client. Software testing is easy process to spell but it has it is own pain. Software testing is maintained in a form of document because again it is taken into consideration for future use and it is used for upgrading the device. Software testing process concentrate on user interfaces. They should be prepared based on human mind. Human factor specialist will take active role in preparing them. A software project success or failure is based on user interfaces. The software document should be user friendly. The design work is based on class and objects. Object is treated as one complete interface in the software. Object is an entity in real time system. This paper focus on how we can design new software project. The paper can help designers, developers and project managers better understand the value of comprehensive testing, and provide guidelines to help them achieve required testing goals. Software Testing Process is shown in Fig. 3.

6. Implementation

1) Self-Assessment

Software code implementation is one stage and the next stage is testing which very important. Testing process make sure that what is design is implemented in coding. After texting the code software professional will run the program for bugs. Based on bugs software professional will review the code. This process is said to self assessment of the code. By doing, we can reduce maximum errors from the code. Some professionals use test templates test cases for assessment.

2) The Psychology and Economics of Software Testing

Software process is related to technology and software testing is a technical task. Software testing involves some important considerations of economics and human psychology. Software testing is a big hurdle in developing software. It requires knowledge related to the system which is going to develop the software. Knowledge related to the programming, study of parameters, syntax related to the language or package. Not only these aspects but knowledge related to ideas and basic decisions making of things is required. Every source code line is tested and is documented. Complete software testing process is a complex process. It should be treated more important than other. It is a logic based level. If the testing process is done properly then the device can be kept safe. But still it is unsafe that means we cannot predict what will happen in next few hours. We may lose some instruction code lines while running a task. It may go for disaster. These devices should be monitored safely and observed seriously. It is related to metal conditions and economical status of software professionals. Each company treat their employees in different ways. There are on proper metrics for it.

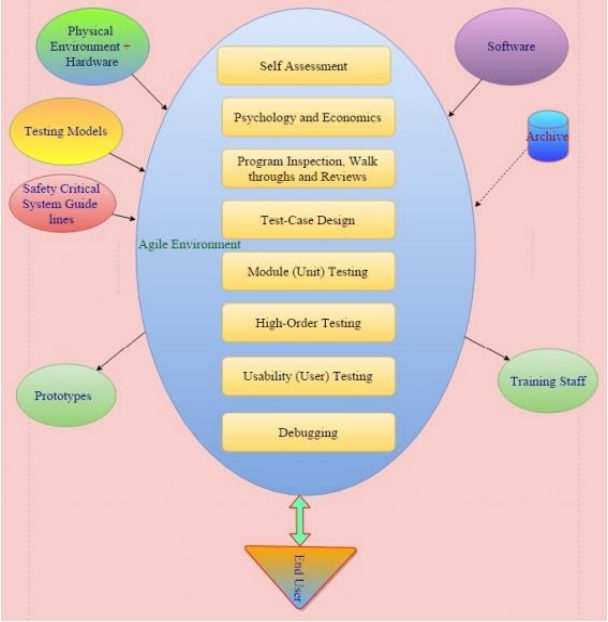

Fig. 3: Software Testing Process.

3) Program Inspection, Walk through and Reviews

Software testing is conducted by different department in most of the multi-national companies. The department staff will receive a testing document from coding wing. After receiving the document, testing department will start the inspection of the document. They go through the comments and commands related to syntax and prepare a review document. There will be through inspection on the document. Reviewer will prepare the review document related to the testing. Sometimes there can be walk through decision with the implementers to avoid unambiguous. Based on coordination and support the testing hurdle will be solved. Software testing document is preserved for future use. This exercise is carried to delivery error defect free software to client and end users.

In software engineering, a walkthrough or walk-through is a form of software peer review "in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems".

A software review is "A process or meeting during which a software product is examined by a project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval".

4) Test Case Design

Various software companies follow different test case designs. Testing is key tone in software development. Every device should be compulsory tested for errors or fault or bugs. This process is mandatory because software cannot be delivered without testing. Every interface, system should be checked thoroughly. This process is done to deliver error free software to the client and end users. Software Testing contains test case which has inputs, outputs and computing part of the code.

5) Module (Unit) Testing

Module testing (or unit testing) is a process of testing the individual code units of the source code. It may contain subprograms, subroutines, classes, or procedures in a program. Instead of testing entire code it is better to test individual units in the code and later reunion of the code can take place. We can test individual units of code or after integrating the entire code we can test.

6) High-Order Testing

Software testing is primary focus of this paper, testing units and integrate testing is not testing. We cannot guarantee that it will work safely and properly. Beyond this testing is called as high-order testing. In high-order testing software professionals will have a bird eye. This is because how systems will work in improper conditions and different cases. Even they will go for negative testing also. Sometimes it is mandatory. Understanding the device and properly examining the device will provide some view in correcting the code. Sometimes mistakes give lot of communication between the team members.

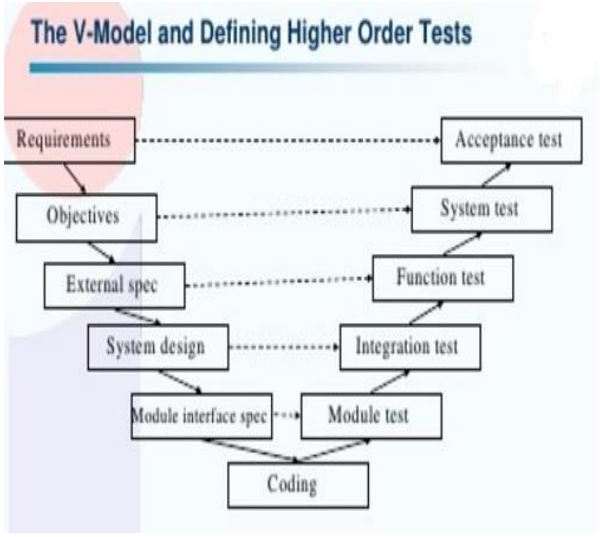

Fig. 4: For V-Model Testing.

7) Usability (User) Testing

According to non-functional requirements usability plays a key role in development of software. Usability is related to level of documentation provided to the user. Whether user requires any training session related to the user interfaces provided under the software. Human factor specialist will design the document and he thinks on behalf of the user to provide user friendly user interfaces to the end user. Project success or failure depends on end users. In software company arranges the testing for user interfaces in a block and gets the feedback and do necessary changes to the user interfaces. Software user interfaces will be designed based on end users. Software success or failure depends on user friend software.

8) Debugging

Debugging is very important role in every software development process. Debugging is the process of what you have to do after completing the code, it is related to testing. After completing the code we can have different decisions on the code and go for certain changes in them. After executing the code what you see is related to successful test case. Successful test case is that which shows what it is insisted to do in the code or program. Debugging has two sides of a coin, one step is related to what has to be done when an error occurs and other side is that after modification what the impact of it on the software is important. It is not that much easy is to fix errors in software. It is related to observation, review and planning of the caused error. Replacing of error is important in debugging.

Debugging process is not new; it is following when first program appeared on the screen. Most common way to deal with debugging process is brute force method. It is on screen because it requires little thought and least mentally bothered method; it is inefficient and unsuccessful method.

Brute force method is categorized into three categories:

- Debugging process carried on with a storage dump.
- Debugging process carried on according based on some available common suggestion throughout the program.
- Debugging process can be carried with automated debugging tools which written based on past history.

9) Physical Environment with Hardware

It can be Flight, Submarine, any Embedded System, Nuclear Plants, Missiles, Satellites, Radars, Defense Rockets and Vehicles etc., Because every system in above list is related to software. Software plays a key role in all systems. Every system is of two parts, Physical system and Software System. Software is like a shield over physical system. Example flight system (Fig. 1 Physical System of Airplane) has flight controls, IMA, landing gear, flaps, ice protection, nose wheel steering, flight management,

battery management, display, navigation, terrain awareness and warning, traffic collision and avoidance, real-time operating systems and many more.

They all are developed physically with boards and chips (hardware) and software is added to it later. The variety of positions and systems has to be experienced and observe common issues, as well as effective solutions, for developing safety-critical software. Inside airplane, every information about external is shown in the form of signals and other data.

10) Software

Information which carries from external to the pilot is based on software. Software is designed based on prototype test and past history of failures. Whenever a new problem related to software or physical arises the device should not loop between code and exceptions.

11) Archive

We face lot of failures in different days of life span. Archive process is that storing all information about the database for future use like developing to build new systems. We maintain archive data in cyclones or disaster management to compare and study the impact of the disaster. Data archives are related to software with indexed and have search capabilities of files. Details can be easily files the parts located and retrieved.

12) Safety Critical System Guide Lines

When software is coin in the year 1970s, the professional had given some guide lines for safety-critical systems to remove or eliminate errors which cause damage to human life rather than devices. USA like government focuses on human life rather than the devices. For every system development is concerned with safety critical systems guide lines in universe. Before implementing any system this guide lines should be followed mandatory.

13) Training Staff

Every Green-field or re-engineering projects, staff should be trained well related to the system, software and hardware. It is better to use senior and freshman combination in project tasks.

14) Prototypes

Better to use prototype systems rather than testing for a small range. Even if it is expensive, better to test prototype model. Prototype model is also called miniature.

15) End User

Project success or failure depends upon end user. Requirements are collected from end user and final project is delivered to end user.

16) Agile Environment

Agile development is related to fast track implementation of complex processes, there should be coordination and collaboration among people is an alternative way of developing project management. In agile process the team work and decision among team are important. The professionals role plays a key role in agile development.

7. Result

By following above instructions we can have better safety software for systems. Below graph shows the difference between them.

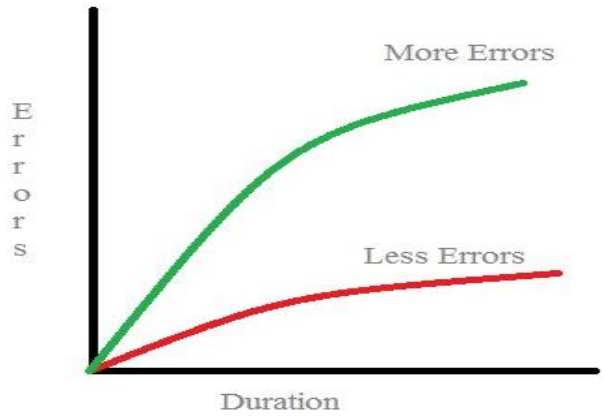

Fig. 5: Difference between Errors.

8. Conclusion

Software is also given importance in every system. World is dependent upon software. It should be handled carefully. It is not matter of system, it matter of human life. Every country should have responsibility in making study on safety critical systems. Recent aircraft accidents are one of the other sides of a coin. Software is on other side. Software is not the primary cause of those failures. Software should help the organizations and government. Let us make user friendly systems with coordination of the software.

References

[1] Michael Scheinholtz “Software Safety”.
[2] W. Eric Wong, Vidroha Debroy, Andrew Restrepo “The Role of Software in Recent Catastrophic Accidents”.
[3] John C Knight “Safety-Critical Systems: Challenges and Directions”.
[4] M Ben Swarup and P Seetha Ramaiah “A Software Safety Model for Safety Critical Applications”.
[5] TRS Prasad Babu, Dimmiti Srinivas Rao, Prathipati Ratna Kumar “Naegative Testing is Trivial for Better Software Products”.
[6] P Ratna Kumar, Raghu D, Udayasree D, Amanullah Mohd. “Chronological Test Generation for Software Project SLoC”.
[7] Ch Suresh Kumar, Dumpati Raghu, Prathipati Ratna Kumar, Dr. G P Saradhi Varma “Addiction of Computer is Sabotage to Human Life”.
[8] Divya Manusha Seethalam, Valiveti Karthik, Valluri Gowthami, Gudikandula Radha Krishna Murthy, Prathipati Ratna Kumar “Software Bug Lilliputians which cause Giant Damage to Systems”.
[9] Chodagam Suresh Kumar, Prathipati Ratna Kumar, Dr. G P Saradhi Varma “Good Requirements collection for Better Software Project”.
[10] Krishna Chaitanya Rudraraju, M Krishna “A Bug Ignorance causes Big Expensive on Smart Systems”.
[11] Kadupukotla Satish Kumar, Panchumarthy Seetha Ramaiah ”Perspectives on Safety Critical Computing Systems”.
[12] Ch Suresh Kumar, D Raghu, P Ratna Kumar “A Domestic Case Studies on Software Failures and Probability to Overcome”.
[13] N Prameela, Raghu Dhumpati “RDCT Software Experts for Healthy Software Products”.
[14] Kadupukotla Satish Kumar, Konda Sreenu, Bathula Prasanna Kumar, Chintagunta Ambedkar “Basic for Easy Caputring and Modelling of Requirements Elicitation, Analysis and Design in Software Development Process”.
[15] Chodagam Suresh Kumar, Mikkili Dileep Kumar, Donavalli Venkata Vidya Deepthi, Prathipati Ratna Kumar “Fast Track Technique for Software Testing and Quality Assurance Practice in Project Development Life Cycle”.