ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

A FRAMEWORK FOR RELIABILITY VALIDATION IN DEPENDENT AND CRITICAL RESPONSIVE SYSTEM DEVELOPMENT

Asheesh Kumar[1], Apurva Mohan Gupta[2], Naresh Rameshrao Pimplikar [3]

M.Tech Student, VIT UNIVERSITY, Vellore, TamilNadu, India[1,2,3]

Abstract: Belonging to the present time the size and complexity of the software systems are increasing rapidly or say growth is exponentially. Present day's scenario for software engineering development method is "first build the software and then Test software" making software's too expensive to develop and less eligible for qualifying. In this paper, discussion is concentrated on the challenges that are faces by software systems for making them highly absolute. Discussion continues with recognize respective source cause region or range of activities and put forward an appropriate framework for RV and better actions that supports to combine respective desirable technology for solving the problem. The respective useful technology that provide solution, such as make a rules of convention for requirements; a conceptual structure or architecture viewpoint centric. The framework makes available the underlying support for the system productive reliability benefits succeed in the dealing with the problems is found software reliability and complexity.

Keywords: Reliability Validation, System Design, Software Development, Framework Analysis for critical response.

I. INTRODUTION

In present days the various embedded and aircraft software is increasing in size and need to respond in the critical situation accordingly. These heavy in size software are combining with the hardware and for a secure and accomplish the aim. When discuss about the analog systems in the highly critical system like avionics, they contain lots of analog data computation and communication is performed using analog signals conversion. So in present time era of software development the digital systems are performs all the tasks, which are the implemented in Embedded systems are taken place over analog design [5]. The fact keeping in mind with increasing complexity the quality and reliability of such software happen to justify inside the range of time schedule and allotted budget. The classical development model (V chart) is given in the fig1. In classical development model, V chart if go towards the lower level the process of developing and code implementation are done, and move towards the upward level process of testing and maintenance are performed [2]. So the building the software first then testing makes it too expensive.

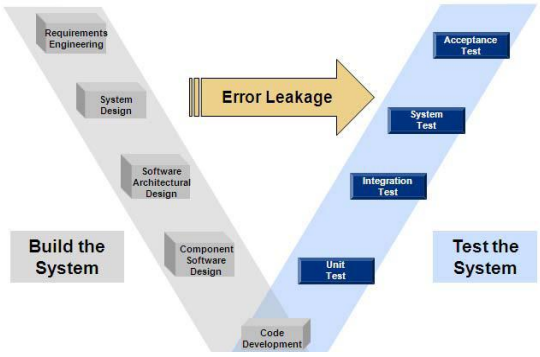

Fig. 1 Classical Software Development Model

II. ITERATURE REVIEW

The different keys term research is done by the various researchers in this field. The different types of the technologies are discussed by the Boehm 2006, Redman 2010 that is discussed: an engineering based on models operate by architectural model using semantics standards. Interaction between the module or self contained small systems and focus on the expressive requirements, produce better specification [1]. Assurance situation of the system gives the confidence that the developed software come into the presence of its purpose and its necessary conditions. The different kinds of the standard are also developed for the meeting the specific project or kind of system development. In this paper the key terms are discuss: Expose the difficulties in development of the highly decision importance system and massive size systems. Discuss about the appropriate framework for the RV and advancement that support or combine with the various technologies. As the hardware implementation in critical responsive system, improvement is always needed up gradation for reliability validation. The implementation of system development can be simulated, analyze and modeled. These kinds of improvements enhance the quality assurance in the developed product. SAE has developed various models and architecture [10] for the modeling of aerodynamics systems, designing the components and testing at embedded level. SAE developed architecture level language [10] for defining the design of critical responsive systems. The COMPASS Model [11] developed for the checking the correctness of the developing product, it also provide the better risk analysis, and performance evolution of the developing system. The COMPASS Project mainly analyzes the requirement analysis at various levels of development

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

phase, and analysis methods for the safety issues, risks analysis, resource consumptions, dependency between modules and performance [11]. The scalability issue in the design of dependent system is a crucial task that achieve by defining resource efficiently by using the resource manager [1]. Reliability engineering is the core component of the designing the critically responsive system because simulation and modeling are based on the previously available data. Reliability validation addresses the mechanical components of in terms of performance, scalability, and correctness [11].

III. DIFFICULTIES IN DEVELOPMENT OF DEPENDENT AND ITICAL RESPONSIVE SYSTEMS

In this paper part, hold a deep outlook at difficulties originates from the massive dependent size software and possibility of source cause. Do this by inspecting the scientific experimental data, on gained knowledge and high risk regions that used to express advantages from software of greater intended engineering and qualifying technology.

A. APPROCHING HIGH COMPLEXITY AND RISING SIZE OF SOFTWARE SYSTEMS

In current days, software development legal expenses and integration of the modules are the serious anxiety. Consider the avionics software that is growing in the massive size, that's why complexity is also increases

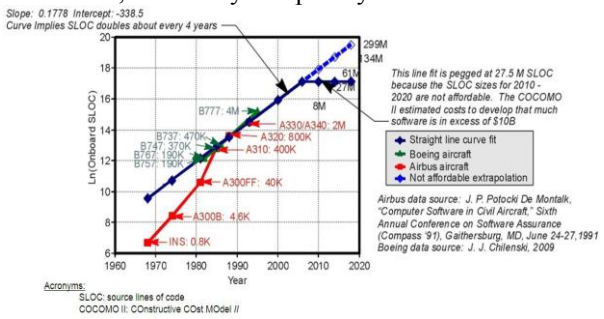

Fig 2 Growing SLOC in Software System Development rapidly.

Fig2 describe the size of the software in aircraft software is become twice in four years. Size of system is growing for appropriate reliance on the system for: understanding the judgmental power, proper communication and handling in modules and fault and risk managing capability for secure and reliable operability. [Fig. 2 source Boeing Synopsys]

B. ERRORS PRODUCED IN REQUIREMENTS GATHERING

Requirement analysis is basic phase of information collection about the software system development. Various types of the requirements error are generated in development as absent or missing, incompatible, over specified, ambiguity, incomplete, wrong or not according to the facts. The predicted way behavior is the segment of the requirement analysis, so it is easily understood that errors in this requirements frequently transmit in the design and code review phases. For that reason it is useful to prove the validity of the detailed description at each point of the requirements gathering.

C. UNSUITABLE PREASSUMPTIONS IN SYSTEM COMMUNICATION

When the system operations are put into the effect, different module of software interacts with each other. A system engineer accept that variables are convey in a specified unit, but that was not share with the software engineer at what time system requirements were converted

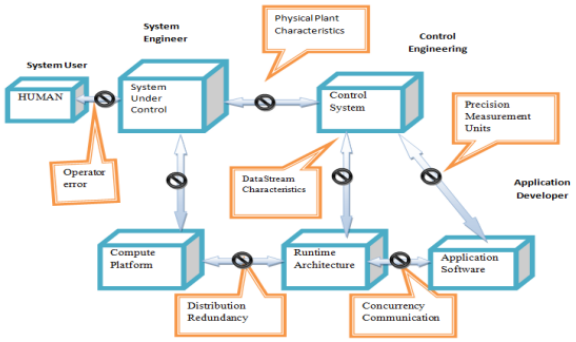

Fig 3 Communication Complexity and Unsuitable assumptions

into the software requirements. Fig 3 gives a clear picture of unsuitable assumptions made and complexity between interactions in software development

D. WORK PLACE COMPLEXITY

At what time a fault is find, a correct answer is select by analyzing it, but not uses to fix it in system in time, in this case the changes may have need to update at design and it may lead the unexpected errors at implementation phase.

In different term can say work place are join in accordance with operationalism and engineers may use up greater time to carry out them. That's point out some serious need for: designer and engineer behavior description.

E. ACHIEVING BETTER RELIABILITY AND BEING DEGRADED OVER TIME

Present plan in better reliability of system is producing clear visual on discovering and take off bugs inspecting review and testing. Testing is interest to apply the inputs to certify that all code segments are running as likely to happen and delivering the supposed results. During testing, large number of communication happens across the source code statements. So abstraction is the concept that used to manage the complexity; object oriented adoption and data abstraction, satisfactory interfaces between modules, apply number of restrictions such as static memory allocation.

Some software systems executes at real time information and data. Such systems are not easy to test on stored data so we need to update out software system. So this is justification that we need to change in design, add some new capability according to data processes in future and made the modification in currently running code. Fig 4 gives a understanding that need of updating over time is required due to better reliability.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

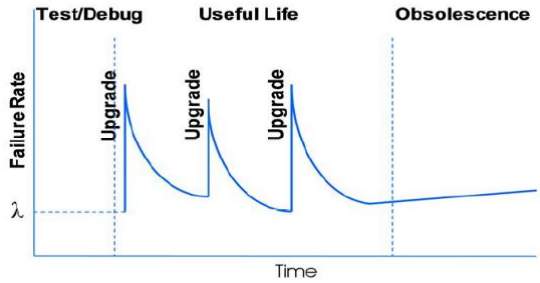

Fig 4 Rate of failure in multiple stage release

IV. DEVLOPING A IMPROVED FRAMEWORK FOR RELIABILITY VALIDATION

In this paper reliability framework is discuss for the improvement in the software system development. The reliability framework combines the software engineering technologies for development phase and quality assurance of the system. In Fig 5 the integration of these technologies discusses:

1. Schematic Requirement analysis at the system and software level at development phase.
2. Model based architecture design.
3. Critical system design properties and static analysis.Quality assurance of system software.

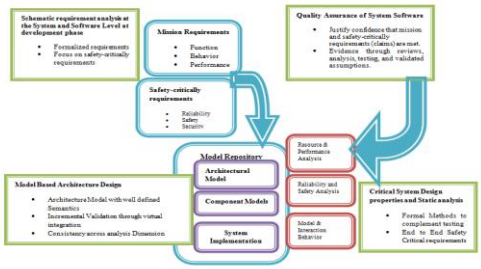

Fig. 5: Reliability Validation and Improvement Framework

A. Schematic Requirement analysis at the system and software level at development phase

The requirement analysis of the system build the in the confidence by insuring consistency about the specification, and also assure about the subsystem requirement. Requirement analysis contains process, business, and product requirements respectively. Requirement analysis further studies as functional and non-functional, design and solution analysis. Critical system design needs some specific requirement that is discuss below: [Miller 2001] [9] Purposed the system for the representation of the data in the tabular form and event/action for showing the requirement specification of critical system. The communication between external environment and the system specification is requiring because based on the input system response accordingly. Specification about the safety and critical system, reliability and safety, security are the main challenge for the system. For satisfying these conditions, make a specification about the faults and hazards in the system. These hazards converted into the safety and critical requirement on the system.

B. Model based architecture design

Modeling the system simulating it and then analyze the system are crucial phase of any engineering field. It is very important for designing of a system that model used in the system is using well defined semantics for the simulating and analyzing to give believable results. In this section architecture design is used the SAE AADL standard for developing the model [11], that is discuss in the Figure 6.

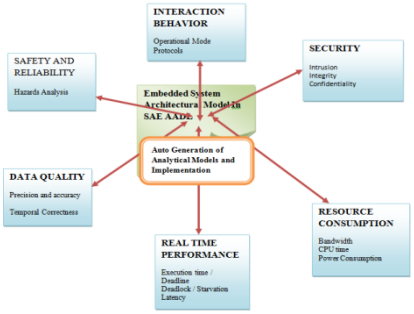

Figure 6: SAE AADL Model

Some advance analyses properties are considered for the architecture design, used for developing the reliability framework are; Scheduling of the resource data flow control security and safety, consumption of resource, error model behavior model are mainly concentrated for the architecture design model.

C. Critical system design properties and static analysis

Before the deployment of the software, it is necessary to mathematically proves system properties using the description of the system. It is a analysis technique that can apply throughout the any software development lifecycle. Static analysis is used for the requirement validation, design model checking for requirement analysis. Implementation phase verification in contrast to design model. Static analysis techniques for the dependent and discrete systems are discussed in this section. Linear temporal logic can be describes about the individual execution of a component within the system: Such as globally, true for every state of execution. Computation tree logic can be used for describing about the requirement with respect to all system behavior at single step. For end-to-end validation using static analysis the COMPASS tool [11] is used for dependent and very large scale software development. The COMPASS tool architecture is discussed below in Fig 7.

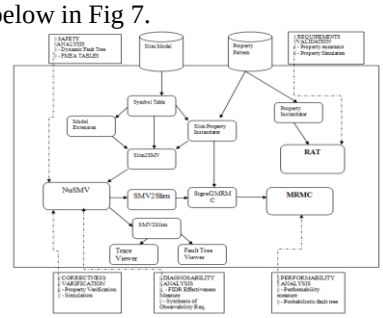

Fig 7 Compass Tools Architecture for critical responsive system

D. Quality assurance of system software

In classical development model the quality assurance only justify by testing and code review verification. But in the dependent and critically respond system some quality certification technique are define such as Integration of

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

safety module with the process description module in the requirement analysis. For the quality assurance of software development a goal structure plan is discussed in this paper. If the evidence support to the goal, which means it satisfies the claim. The goal structure model is discussed in fig 8

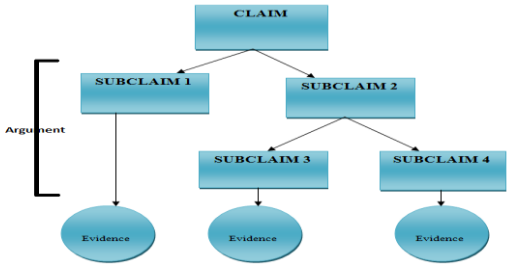

Fig 8 Quality assurance at different level of satisfaction

Quality assurance applies over the different phases of software development life-cycle. In each phase of development life-cycle the quality assurance is fully documented for developed system. It provides a confidence for quality of product and also helps in the other development phases. The reusability of design software with a quality assurance document is importantly greater as compare to another develop without it.

V.CONCLUSION

In the classical software development model "first build the software and then Test software" gives the error leakage at each phase and it increase the cost of system development and reduce the quality assurance of the product. In this paper the key model for developing of giant system is discussed. Schematic Requirement analysis at the system and software level at development phase study all about the requirement of the system development, checking the consistency of developing system. Model based architecture design provides physical interaction between components of system. The well defined semantic information annotated with analysis and design phase gives greater understating about the system. Critical system design properties and static analysis provides the detail design procedure, system requirement consistency throughout the software development life-cycle. Quality assurance of system software provides confidence at each phase of development that supported by evidence to safety and critical response of system.

REFERENCES

[1] Quan, hoang, Nguyen, N.Nguyen, ET. Al. "MAFSE A MODEL-BASED FRAMEWORK FOR SOFTWARE VERIFICATION", 2010 Fourth IEEE IC on SSI and RI page 150-156.

[2] Meedeniya, Grunske "an Efficient Method for Architecture-based Reliability Evaluation for Evolving Systems with Changing Parameters" 2010 IEEE 21st IS on SRE Page 229-238.

[3] Koziolek, Schlich, Bilich "a Large-Scale Industrial Case Study on Architecture-based Software Reliability Analysis" 2010 IEEE 21 IS on SRE page 279-289.

[4] Dina Salah, "a framework for the Integration of User Centered Design and Agile Software Development Processes" ACM May_2011 page 1132-1134

[5] Dina Salah, "a framework for the Integration of User Centered Design and Agile Software Development Processes" ACM May_2011 page 1132-1134.

[6] Redman, David, Ward, Donald, Chilenski, John, & Pollari, Greg. "Virtual Integration for Improved System Design", Proceedings of The First Analytic Virtual Integration of Cyber Physical Systems Workshop in conjunction with the Real-Time Systems Symposium (RTSS 2010). San Diego, CA, November 2010.

[7] Miller, S. & Tribble, A. "Extending the Four-Variable Model to Bridge the System-Software Gap."presented at the 20th Digital Avionics Systems Conference (DASC01). Daytona Beach, FL, October 2001.

[8] Society of Automotive Engineers International. "Architecture Analysis & Design Language (AADL)", SAE International Standards Document AS5506B, 2012.

[9] ["Correctness, Modeling and Performance of Aerospace Systems" (COMPASS).

BIOGRAPHIES

Asheesh Kumar is M. Tech (CSE) Student at VIT University Vellore TamilNadu India. He did his B.Tech (IT) from IIMT Engineering College, Meerut, UP. His area of Interest is Algorithms Analysis, Theory of Computation, Web Services and Cloud Computing.

Naresh Rameshrao Pimplikar is M. Tech (CSE) Student at VIT University Vellore TamilNadu India. He did his BE (IT) from Priyadarshini College of Engineering, Nagpur, Maharashtra. His area of Interest is Database Management System, Operating System, and Computer Graphics.

Apurva Mohan Gupta is M. Tech (CSE) Student at VIT University Vellore TamilNadu India. He did his BE (CSE) from Global Institute of Technology, Jaipur, Rajasthan. His area of Interest is Data Structure, Computer Architecture and Computer Graphics.